



TRUSTEE

TRUSTEE D4.1 Live document, TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Work package	WP4: TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework
Task	- Task 4.1 Privacy impact assessment - Task 4.2 Workflow Automations, Robotic process automation, Dashboard and User interface - Task 4.3 SSI Homomorphic capable Framework - Task 4.4 Trustee SDK and Open cloud Continuum API - Task 4.5 Data Lake building for cross border data transfer - Task 4.6 Security and trust manager
Authors	Cristina Regueiro – TECNALIA, Iñaki Seco – TECNALIA, Julen Bernabe – TECNALIA, Alberto Berreteaga - TECNALIA, Santiago De Diego - TECNALIA, Marisa Escalante – TECNALIA, Borja Urquizu – TECNALIA, Xenofon Vurakis– HMU, Kalliopi Meladaki – HMU, Dimitra Papatsaroucha – HMU, Evangelos Markakis – HMU and the rest of the HMU team, Hande Oezkayagan Praendl – UNIVIE, Clara Saillant – UNIVIE, Katarzyna Barud - UNIVIE, Panagiotis Radoglou Grammatikis – K3Y, Savvas Ouzounidis– K3Y, George Nakas– K3Y, Marios Siganos– K3Y, Evangelos Mourikis– K3Y, Zacharenia Lekka– K3Y, Antonios Sarigiannidis - K3Y, Sarwar Sayeed – ENU, Nikolaos Pitropakis – ENU, Maria Zotou - ACCELI, Stergios Zissakis - ACCELI, Stelios Vourakis- ACCELI, Pantelis Velanas- ACCELI, Maurizio Martignano - UCSC, Andrej Grgurić - ENT, Ivana Stupar – ENT, Nikolaos Papachatzakis - ADR, Stelios Klados – ADR, Giulia Rutto – RINA, Manuele Barbieri- RINA, Akis Kourtis – ZORTENET, George Xilouris – ZORTENET, Andreas Oikonomakis – ZORTENET, Athanasios Grammatopoulos - INQBIT, Ilias Politis - INQBIT, Michał Kepkowski - INQBIT, Konstantinos Kefalas - INQBIT, Maria Liatsou – INQBIT, Maria Sampazioti Vasiliki – INQBIT, Ilkka Niskanen – VTT, Saneela Malik – VTT, Sofiane Lagraa - FUJITSU, Sven Rasmusen – FUJITSU, Moussa Ouedraogo- FUJITSU, Pasquale Di Dio - HPE, Christos Anagnostopoulos – ATHENA, Aris Lalos – ATHENA, Christos Mavrokefalidis – ATHENA, Vasilis Pikoulis - ATHENA
Dissemination level	Public
Status	Final
Due date	30/09/2023
Document date	27/09/2023
Version number	V1.0



**Funded by the
European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

TABLE OF CONTENTS

TABLE OF CONTENTS	3
1. EXECUTIVE SUMMARY	xxiii
2. INTRODUCTION	24
3. PART I: DASHBOARD	28
3.1. INTRODUCTION	28
3.1.1. Need and Objectives	28
3.1.2. Innovation	28
3.2. BACKGROUND	29
3.2.1. User interface design	29
3.2.2. Robotic process automation	30
3.2.3. Virtual assistant	31
3.3. REQUIREMENTS	31
3.4. ARCHITECTURE	36
3.4.1. High level	36
3.4.1. Low level	37
3.5. IMPLEMENTATION	41
3.5.1. Functional Description	41
3.5.2. Operational Description	44
3.5.3. Technical Description	55
3.5.4. Technical Details	113
3.6. VALIDATION	114
3.6.1. Validation Methodology	114
3.6.2. Validating the workflow automation	115
3.7. NEXT STEPS	117
3.8. CONCLUSIONS	117
4. PART II: DATA PROTECTION IMPACT ASSESSMENT	118
4.1. INTRODUCTION	118

4.1.1.	Need and Objectives	118
4.1.2.	Innovations	119
4.2.	BACKGROUND	119
4.2.1.	Legal and Regulatory Frameworks	119
4.2.2.	DPIA Processing Activities	121
4.3.	REQUIREMENTS	123
4.4.	ARCHITECTURE	125
4.4.1.	High level.....	125
4.4.2.	Low level	126
4.5.	IMPLEMENTATION	127
4.5.1.	Functional Description.....	127
4.5.1.	Operational Description.....	129
4.5.2.	Technical Description	130
4.5.3.	Technical Details	157
4.6.	VALIDATION.....	157
4.7.	NEXT STEPS	157
4.8.	CONCLUSIONS.....	158
5.	PART III: SELF-SOVEREIGN IDENTITY HOMOMORPHIC CAPABLE FRAMEWORK	159
5.1.	INTRODUCTION	159
5.1.1.	Need and Objectives	159
5.1.2.	Innovation.....	159
5.2.	BACKGROUND	160
5.2.1.	Homomorphic Encryption	160
5.2.2.	Self-Sovereign Identity	163
5.3.	REQUIREMENTS.....	164
5.4.	ARCHITECTURE	166
5.4.1.	High level.....	166
5.4.2.	Low level	168
5.5.	IMPLEMENTATION	179

5.5.1.	Functional Description.....	180
5.5.2.	Operational Description.....	182
5.5.3.	Technical Description	184
5.5.4.	Technical Details	186
5.6.	VALIDATION.....	186
5.6.1.	Main functionalities for the issuer	187
5.6.2.	Main functionalities for the holder	188
5.6.3.	Main functionalities for the verifier	189
5.7.	NEXT STEPS	190
5.8.	CONCLUSIONS.....	191
6.	PART IV: DATA LAKE	192
6.1.	INTRODUCTION	192
6.1.1.	Need and Objectives	192
6.1.2.	Innovation.....	193
6.2.	BACKGROUND	194
6.2.1.	Semantic Interoperability.....	194
6.2.2.	Data Lakes	198
6.3.	REQUIREMENTS.....	199
6.4.	ARCHITECTURE	201
6.4.1.	High level.....	201
6.4.1.	Low level	202
6.5.	IMPLEMENTATION	206
6.5.1.	Functional Description.....	206
6.5.2.	Operational Description.....	206
6.5.3.	Technical Description	209
6.5.4.	Technical Details	239
6.6.	VALIDATION.....	239
6.7.	NEXT STEPS	243
6.8.	CONCLUSIONS.....	244

7.	PART V: SECURITY AND TRUST MANAGER	245
7.1.	INTRODUCTION	245
7.1.1.	Need and Objectives	245
7.1.2.	Innovation	245
7.2.	BACKGROUND	246
7.2.1.	Ontology	246
7.2.2.	Graph database	246
7.2.3.	GDPR principles	246
7.2.4.	Technology Enablers	247
7.3.	REQUIREMENTS	247
7.4.	ARCHITECTURE	249
7.4.1.	High level	249
7.4.2.	Low level	250
7.5.	IMPLEMENTATION	253
7.5.1.	Functional Description	253
7.5.2.	Operational Description	260
7.5.3.	Technical Description	265
7.5.4.	Technical Details	276
7.6.	VALIDATION	278
7.7.	NEXT STEPS	278
7.8.	CONCLUSIONS	279
8.	PART VI: ONE STOP SHOP	280
8.1.	INTRODUCTION	280
8.1.1.	Need and Objectives	280
8.1.2.	Innovation	280
8.2.	BACKGROUND	280
8.3.	REQUIREMENTS	281
8.4.	ARCHITECTURE	282
8.4.1.	High level	282

8.4.1.	Low level	283
8.5.	IMPLEMENTATION	287
8.5.1.	Developer Handbook	287
8.5.2.	SDK External Connectors	289
8.5.3.	SDK Internal connectors	308
8.6.	VALIDATION	308
8.7.	NEXT STEPS	309
8.8.	CONCLUSIONS	309
9.	CONCLUSIONS	311
10.	REFERENCES	312
11.	ANNEX 1: GLOSSARY OF TERMS	320
12.	ANNEX 2: LEGAL REQUIREMENTS	324
12.1.	DA	324
12.2.	DPIA	328
12.3.	SSI-HE	337
12.4.	DL	341
12.5.	STM	344
12.6.	OneSS	352
13.	ANNEX 3: SOCIO-ETHICAL REQUIREMENTS	364
13.1.	DA	364
13.2.	DPIA	365
13.3.	SSI-HE	367
13.4.	DL	368
13.5.	STM	368
13.6.	OneSS	369
14.	ANNEX 4: DPIA ENDPOINTS	370
15.	ANNEX 6: STM ENDPOINTS	393
16.	ANNEX 5: DEVOPS AND CI/CD METHODOLOGY	398
16.1.	Scope	398

16.2.	DevOpS Process.....	399
16.2.1.	Pipeline CI/CD – Diagrams.....	400
16.2.2.	Pipeline CI/CD – Tools	400
16.2.3.	Pipeline CI/CD – Stages.....	401
16.2.4.	Pipeline CI/CD – Environments.....	402
16.2.5.	Adoption in Phases.....	403
16.3.	Development Guidelines.....	406
16.3.1.	Multi-Repository Structure.....	407
16.4.	Cloud Native Design.....	407
16.4.1.	Microservices as Independent Processes	407
16.4.2.	One Process per Container	407
16.4.3.	Stateless Microservices	408
16.4.4.	Externalized Configuration	408
16.4.5.	Logging	408
16.4.6.	Readiness and Liveness Probes	408
16.4.7.	Resource Requests and Limits.....	408
16.5.	Branching Models	408
16.5.1.	Requirements for Release Frequency	409
16.6.	Release Management - Principles	414

Index of figures

Figure 1: TRUSTEE Detailed architecture. WP3 and WP4 ABBs.....	25
Figure 2: Relationship between D4.1 and other WPs	26
Figure 3: DA - High level architecture diagram	37
Figure 4: DA- Low level architecture diagram.....	37
Figure 5: Sequence diagram of TRUSTEE Data Consumer use case flow part 1	46
Figure 6: Sequence diagram of TRUSTEE Data Consumer use case flow part 2	47
Figure 7: Flow diagram of TRUSTEE Data Consumer use case flow	48
Figure 8: Sequence diagram of TRUSTEE Data Provider use case flow part 1	49
Figure 9: Sequence diagram of TRUSTEE Data Provider use case flow part 2	50
Figure 10: Flow diagram of TRUSTEE Data Provider use case flow	51
Figure 11: Workflow automation Sequence Diagram	52
Figure 12: Data provider Virtual Assistant conversation.....	53
Figure 13: RPA Robot Flow diagram	54
Figure 14: RPA Automation Manager Flow diagram.....	55
Figure 15: RPA Process file data model.....	59
Figure 16: VA to Automation Manager message model	60
Figure 17: TRUSTEE Header.....	67
Figure 18: TRUSTEE User Profile Element and the Related Dropdown Menu.....	69
Figure 19: TRUSTEE Breadcrumbs placed on the top left beside the navigation sidebar	71
Figure 20: TRUSTEE Footer.....	72
Figure 21: TRUSTEE Navigation Sidebar	74
Figure 22: TRUSTEE Main - Buttons	78
Figure 23: TRUSTEE Disabled Buttons.....	78
Figure 24: TRUSTEE Add - Button.	78
Figure 25: TRUSTEE Delete, Remove and Cancel buttons	79
Figure 26: Select and Input (Forms)	80
Figure 27: Disabled Form component	80
Figure 28: TRUSTEE indicative information message	83

Figure 29: TRUSTEE indicative warning message	83
Figure 30: TRUSTEE indicative error message	84
Figure 31: DA Log-in page.....	85
Figure 32: DA Sign-up page.....	86
Figure 33: DA Home Page	87
Figure 34: DA Use Data Page Entry Point	88
Figure 35: DA Use Data Page Search point.....	88
Figure 36: DA Provide Data Page Entry Point	89
Figure 37: DA KR Page.....	90
Figure 38: DA My Results Page without results	91
Figure 39: DA My Results Page populated with results	91
Figure 40: DA My Datasets Page empty	92
Figure 41: DA My Datasets Page populated with datasets.....	93
Figure 42: DA Use Data Initial Page	94
Figure 43: DA Use Data Page Filtering.....	94
Figure 44: DA Use Data Page Filtering Operation in Advanced Search mode	95
Figure 45: DA Use Data Page Filtering Operation in Compiled Search mode.....	95
Figure 46: DA Use Data Page Similar published Queries Popup	96
Figure 47: DA Page Loader - Loading Search Query Results	96
Figure 48: DA Use Data Page Select Datasets for fusion.....	97
Figure 49: DA Page Loader – Fetching available computations that can be performed on the fused dataset	97
Figure 50: DA Use Data Page Select Computation	98
Figure 51: DA Use Data Page STM Redirection.....	98
Figure 52: DA Use Data Page Agreements Success.....	99
Figure 53: DA Use Data Page Agreements Failure	99
Figure 54: DA Use Data Page Computation Start	100
Figure 55: DA My Results Process In Progress.....	100
Figure 56: DA My Results Page Selected Process without computation details	101
Figure 57: DA My Results Page Reuse Popup	101

Figure 58: DA My Results Page with computation details.....	102
Figure 59: DA My Results Page Save Process in the Knowledge Repository pt.1	102
Figure 60: DA My Results Page Save Process Results in the Knowledge Repository pt.2.....	103
Figure 61: DA My Results Page Agreements not signed – computation now allowed to be performed	103
Figure 62: DA Provide Data Page Submit Dataset Description	104
Figure 63: DA Provide Data Page Edit Dataset Description	105
Figure 64: DA My Datasets Page	105
Figure 65: Virtual Assistant hidden state. Initial view.....	106
Figure 66: Virtual Assistant expanded state - First message	107
Figure 67: Virtual Assistant – Navigation use case #1	108
Figure 68: Virtual Assistant – Navigation use case #2	109
Figure 69: Virtual Assistant – Navigation use case #3	110
Figure 70: Virtual Assistant – Navigation use case #4	111
Figure 71: Virtual Assistant – Navigation use case #5	112
Figure 72: Workflow automation test on Windows Calculator	116
Figure 73: PACTS iterative process for conducting out a DPIA	122
Figure 74: High Level Architecture of the DPIA ABB (D2.1 [1]- Figure 108)	126
Figure 75: Low Level Architecture of the DPIA ABB (D2.1 [1]- Figure 109)	126
Figure 76 DPIA Workflow	129
Figure 77: Sequence Diagram of the DPIA ABB including external ABBs (D2.1 [1]– Figure 110)	130
Figure 78: Database Schema of the DPIA ABB	137
Figure 79: DPIA Interface Welcome Screen	139
Figure 80: DPIAs Screen	140
Figure 81: DPIA Archives Screen	140
Figure 82: DPIA Templates Screen	141
Figure 83: DPIA Knowledge Bases Screen	141
Figure 84: New DPIA, DPIA Template and DPIA Knowledge Base Forms	142
Figure 85: DPIA Templates Screen	143
Figure 86: DPIA Knowledge Base Screen	143

Figure 87: DPIA Knowledge Base Elements Form	144
Figure 88: DPIA Main Screen	145
Figure 89: DPIA Risks Assessment.....	146
Figure 90: DPIA Review Request	147
Figure 91: DPIA Messages.....	147
Figure 92: DPIA Review Phase.....	148
Figure 93: DPIA Validation Phase	149
Figure 94: DPIA Evaluation Options.....	149
Figure 95: DPIA Risks Overview Visualisation.....	150
Figure 96: DPIA Risk Mapping Visualisation.....	151
Figure 97: DPIA Action Plan Visualisation	152
Figure 98: DPO and Data Subjects' Opinions Screen	153
Figure 99: DPIA Final Rejection or Approval.....	154
Figure 100: Threshold FHE operation	162
Figure 101: SSI main components.....	164
Figure 102: SSI-HE high level architecture.....	167
Figure 103: SSI-HE low level architecture.....	168
Figure 104: Threshold FHE in the SSI-HE component.....	171
Figure 105: SSI-HE metadata generation phase	173
Figure 106: SSI-HE metadata validation phase	175
Figure 107: SSI-HE result sharing phase.....	177
Figure 108: SSI-HE result sharing phase with existing results.....	178
Figure 109: SSI-HE first implementation functional components.....	180
Figure 110: SSI-HE first implementation operation flow.....	183
Figure 111: SSI-HE first implementation details.....	184
Figure 112: SSI-HE first implementation available functionalities	187
Figure 113: SSI-HE first implementation functional Issuer_Test 1.....	187
Figure 114: SSI-HE first implementation functional Issuer_Test 2.....	187
Figure 115: SSI-HE first implementation functional Issuer_Test 3.....	188

Figure 116: SSI-HE first implementation functional Holder_Test 1	188
Figure 117: SSI-HE first implementation functional Holder_Test 2	188
Figure 118: SSI-HE first implementation functional Holder_Test 3	188
Figure 119: SSI-HE first implementation functional Holder_Test 4	188
Figure 120: SSI-HE first implementation functional Holder_Test 5	189
Figure 121: SSI-HE first implementation functional Holder_Test 6	189
Figure 122: SSI-HE first implementation functional Holder_Test 7	189
Figure 123: SSI-HE first implementation functional Verifier_Test 1	189
Figure 124: SSI-HE first implementation functional Verifier_Test 2	189
Figure 125: SSI-HE first implementation functional Verifier_Test 3	190
Figure 126: SSI-HE first implementation functional Verifier_Test 4	190
Figure 127: SSI-HE first implementation functional Verifier_Test 5	190
Figure 128: SSI-HE first implementation functional Verifier_Test 6	190
Figure 129: SSI-HE first implementation functional Verifier_Test 7	190
Figure 130: FAIR Principles.....	197
Figure 131: DL ABB breakdown diagram	202
Figure 132: GraphDB architecture	203
Figure 133: Data Indexes low level architecture	204
Figure 134: Data catalogues low level architecture	205
Figure 135: Attributes catalogue low level architecture	205
Figure 136: AI models storage low level architecture	205
Figure 137: HE Public Keys storage low level architecture	206
Figure 138: Sequence diagram for operations related to DL	207
Figure 139 DL interactions for Data Consumer persona scenario	209
Figure 140 DL data flow for Data Provider scenario.....	209
Figure 141: Space data model for moonquakes dataset (OWLGrEd).....	211
Figure 142: Space data model for moonquakes dataset (WebVOWL)	212
Figure 143: Space weights dataset semantic representation	213
Figure 144: Solar index dataset semantic representation.....	214

Figure 145: Holistic Space data model semantic representation	214
Figure 146: Holistic Space data model relational representation.....	215
Figure 147: Education data model semantic representation	216
Figure 148: Students school dataset relational representation	218
Figure 149: Simplified visualization of semantic schema for Students dataset	219
Figure 150: Relational semantic visualization of Students dataset	219
Figure 151: Simplified semantic diagram of dummy Schools dataset.....	220
Figure 152: Relational semantic visualization of Schools dataset	220
Figure 153: Simplified semantic visualization of Subjects dataset.....	221
Figure 154: Relational visualization of Subjects dataset	221
Figure 155: Simplified semantic visualization of Grades dataset	222
Figure 156: Relational visualization of Grades dataset	222
Figure 157: Simplified semantic visualization of Classes dataset	222
Figure 158: Relational visualization of Classes dataset.....	223
Figure 159: Simplified semantic visualization of Assessments table	223
Figure 160: Relational visualization of Assessments dataset	224
Figure 161: Simplified semantic visualization of Student_grades dataset.....	225
Figure 162: Relational visualization of Student_grades dataset	225
Figure 163: Relational visualization of Automotive models	229
Figure 164: Simplified semantic visualization of Automotive ML models.....	229
Figure 165: Relational visualization of 2D Scene Analysis dataset	231
Figure 166: Simplified semantic visualization of 2D Scene Analysis dataset	231
Figure 167: Relational visualization of 3D Scene Analysis dataset	233
Figure 168: Simplified semantic visualization of 3D Scene Analysis dataset.....	234
Figure 169: GraphDB Semantic data sets of TRUSTEE pilots	235
Figure 170: DL GraphDB RDF store	236
Figure 171: GraphDB visual graph of TRUSTEE semantic classes.....	237
Figure 172: GraphDB visual graph of TRUSTEE semantic relationships / properties.....	238
Figure 173: Example of semantic dataset in GraphDB.....	240

Figure 174: SPARQL query of dataset concept in Health domain	240
Figure 175: SPARQL query of dataset concepts in Education domain (inter-disciplinary)	241
Figure 176: SPARQL query of dataset concepts (semantic interoperability)	241
Figure 177: DL Data Models (APIs) design	242
Figure 178: Storing HE Public Keys in PostgreSQL	243
Figure 179: HE Public Keys retrieval within DL	243
Figure 180: STM Manager's overall architecture	249
Figure 181: STM architecture	250
Figure 182: A detailed architecture of the STM.	252
Figure 183: Agreement management life-cycles	253
Figure 184: Agreement ontology class hierarchy	255
Figure 185: Agreement ontology's concepts and their relation to another.	256
Figure 186: Security and audit monitoring architecture	260
Figure 187: The sequence diagram for creating a dataset agreement as a Data Provider	262
Figure 188: The sequence diagram for signing a dataset agreement as a Consumer	263
Figure 189: Store Blockchain Network Credentials sequence diagram	264
Figure 190: Contract Events Subscription sequence diagram	264
Figure 191: Store Contract Events sequence diagram	265
Figure 192: Access Contract Events sequence diagram	265
Figure 193: The methodology used to develop the STM	266
Figure 194: The prototype Dashboard Page for the STM using Figma.	267
Figure 195: All 8 data objects and their relationships toward each other in the STM	268
Figure 196: An overview of the Smart Contract Agreement data model	269
Figure 197: An overview of the Hyperledger Explorer Dashboard	271
Figure 198: STM and IDS Connectors Interrelationship	273
Figure 199: Sequence diagram STM- TAI-SDF	274
Figure 200: An overview of the TAI-SDF-STM data model	274
Figure 201: An overview of the DPIA-STM data model	275
Figure 202 Testing Strategy	278

Figure 203: OneSS building block architecture (from D2.1 [1]).....	283
Figure 204: TRUSTEE Developer Handbook	284
Figure 205: IDS connector in integration with TRUSTEE.....	285
Figure 206: IDS connector architecture.....	286
Figure 207: GAIAX integration with TRUSTEE	287
Figure 208: TRUSTEE Developer Handbook contents (status in July 2023).....	288
Figure 209: An example of development-related content available within the handbook.....	289
Figure 210: IDS Connector integration cases – data provided from the IDS Connector to the TRUSTEE users ..	292
Figure 211: TRUSTEE components involved in the GAIAX Integration Case 1	294
Figure 212: Sequence diagram - GAIAX Integration Case 1	296
Figure 213: TRUSTEE components involved in the GAIAX Integration Case 2	298
Figure 214: Sequence diagram - GAIAX Integration Case 2	300
Figure 215: EOSC Services [87]	301
Figure 216: Copernicus Data Space Ecosystem roadmap	303
Figure 217: Copernicus Data Space Economy home page	304
Figure 218: EGI Check-in Page.....	306
Figure 219: EGI Check-in Token Generation.....	307
Figure 220: Use of generated EGI token in API requests.....	307
Figure 221: DevOps - Stages.....	398
Figure 222: DevOps - Tools	399
Figure 223: Pipeline CI/CD – Diagrams.....	400
Figure 224: Pipeline CI/CD – Tools.....	400
Figure 225: CI/CD pipeline with QA Environment.....	405
Figure 226: Hotfix environments - Example	406
Figure 227: Git Branching Model [97]	410
Figure 228: TBD with release branches	412
Figure 229: TBD with Trunk-Based Releases	413

Index of tables

Table 1: DA – User-centred requirements	32
Table 2: DA – Functional and non-functional technical requirements	34
Table 3: DA-AM OIDC Client Registration Data Model	56
Table 4: DA-AM User Authentication Data Model (Code part)	57
Table 5: DA-AM User Authentication Data Model (Token part)	57
Table 6: DA Usability Requirements	62
Table 7: TRUSTEE’s colour palette	64
Table 8: TRUSTEE’s colour palette for the TRUSTEE Platform	64
Table 9: TRUSTEE Body Background and Text Guidelines	65
Table 10: TRUSTEE Headings - Guidelines	66
Table 11: TRUSTEE Header Guidelines	67
Table 12: TRUSTEE User Profile Guidelines	68
Table 13: TRUSTEE Icons for Dropdown Menu	68
Table 14: TRUSTEE Header Suggested Icons	69
Table 15: TRUSTEE Breadcrumbs Guidelines	70
Table 16: TRUSTEE Footer Guidelines	71
Table 17: TRUSTEE Footer Icons and EU Flag	72
Table 18: TRUSTEE Footer Labels and EU Text	72
Table 19: TRUSTEE Navigation Sidebar Guidelines	73
Table 20: TRUSTEE Navigation Sidebar Icons	74
Table 21: TRUSTEE Suggestions on Menu Tabs	76
Table 22: TRUSTEE Main Button Guidelines	76
Table 23: Green Button Guidelines - Positive Messaging	78
Table 24: Red Button Guidelines - Cancelation	78
Table 25: TRUSTEE Select and Input (Forms)	79
Table 26: All DA Colours	80
Table 27: Main UI technical details	113
Table 28: RPA Technical details	113

Table 29: Virtual Assistant technical details	114
Table 30: DPIA – User-centred requirements	123
Table 31: DPIA – Functional and non-functional technical requirements.....	125
Table 32: DPIA Interface’s Components and Routes	132
Table 33: Database Tables of the DPIA ABB	133
Table 34: Structure of Sections, Items and Questions	138
Table 35: Sections, sub-sections and included questions	154
Table 36: DPIA Technical Details.....	157
Table 37: SSI-HE - User-centred requirements	164
Table 38: SSI-HE - Functional and non-functional technical requirements	166
Table 40: SSI-HE component first implementation functionalities	180
Table 41: SSI-HE component first implementation API details	185
Table 42: SSI-HE component first implementation technical details	186
Table 43: DL - User-centred requirements	200
Table 44: DL – Functional and non-functional technical requirements.....	201
Table 45: Space dataset: Moonquakes (original).....	210
Table 46: Weights of neural networks dataset	212
Table 47: Solar index dataset.....	213
Table 48: Student school dataset	216
Table 49: Students dataset	218
Table 50: Schools dataset	219
Table 51: Subjects dataset	220
Table 52: Grades dataset.....	221
Table 53: Classes dataset.....	222
Table 54: Assessment table	223
Table 55: Student_grades dataset	224
Table 56: eCRF_Rapid_Covid19_Module2_2A_Vitals dataset.....	225
Table 57: Encounter dataset.....	226
Table 58: Countries dataset	227

Table 59: Automotive model labels.....	228
Table 60: Automotive 2D Scene Analysis dataset labels	230
Table 61: Automotive 3D Scene Analysis dataset labels	231
Table 62: Automotive Cooperative localization & KalmanNet dataset labels	234
Table 63: DL APIs communications	238
Table 64: DL component first implementation technical details	239
Table 65: STM - User-centred requirements	248
Table 66: STM - Functional and non-functional requirements.....	248
Table 68: A description of the standard HTTP methods.	275
Table 69: Technical details of STM frontend	276
Table 70: Technical details of STM backend	276
Table 71: Technical details of STM Server	277
Table 72: Technical details of STM connector.....	277
Table 73: OneSS – User-centred requirements.....	281
Table 74: OneSS – Functional and non-functional technical requirements.....	282
Table 75: DA – Legal requirements	324
Table 76: DPIA – Legal requirements.....	328
Table 77: SSI-HE - Legal requirements	337
Table 78: DL - Legal requirements.....	341
Table 79: STM - Legal requirements.....	345
Table 80: OneSS – Legal requirements	352
Table 81: DA - Socio-ethical requirements	364
Table 82: DPIA – Socio-ethical Requirements.....	365
Table 83: SSI-HE - Socio-ethical requirements.....	367
Table 84: DL - Socio-ethical requirements.....	368
Table 85: STM - Social-ethical requirements.....	368
Table 86: OneSS – Socio-ethical requirements	369
Table 87: Endpoints of the RESTful API of the DPIA ABB.....	370
Table 90: The first table of the STM's REST API endpoints.....	394

Table 91: The second table of the STM's REST API endpoints.	395
Table 88: DevOps – Stages and tools	399
Table 89: Division of responsibilities between Developers, QA, and DevOps Engineers	406

LIST OF ABBREVIATIONS AND ACRONYMS

Acronym	Meaning
ABB	Architecture Building Block
AI	Artificial Intelligence
AIMaaS	AI Models as a Service
AM	Authentication Manager
AMQP	Advanced Message Queuing Protocol
API	Application Program Interface
ATR	Accountable Transaction Recorder
CA	Certificate Authority
CC	Crypto Context
CLI	Command-Line Interface
CNIL	Commission Nationale de l'Informatique et des Libertés (French Data Protection Authority)
DA	TRUSTEE Dashboard
DAPS	Dynamic Attributes Provisioning Service
DFS	Depth First Search
DID	Decentralized Identifiers
DL	Data Lake
DOI	Digital Object Identifier
DPIA	Data Protection Impact Assessments
DPO	Data Protection Officer
DSIL	Data Spaces Innovation Lab
DSILC	Data Spaces Innovation Lab Connector
DSS	Digital Signature Services
EAA	European Accessibility Act
EGI	European Grid Infrastructure
eIDAS	Electronic Identification and Trust Services
EOSC	European Open Science Cloud
EU	European Union
FAIR	Findable, Accessible, Interoperable, and Reusable
FAQ	Frequently Asked Questions
FHE	Fully Homomorphic Encryption
FLT	Fermat's Little Theorem
GDPR	General Data Protection Regulation
GraphDB	Graph Database
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HE	Homomorphic Encryption
HEDF	Homomorphic Enabled Data Fusion
HLF	Hyperledger Fabric
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IDS	International Data Spaces
IDSA	International Data Spaces Association
IPFS	InterPlanetary File System
IoT	Internet of Things
JSON	JavaScript Object Notation

KPI	Key Performance Indicators
KR	Knowledge Repository
ML	Machine Learning
MT	Monitoring Tool
NIST	National Institute of Standards and Technology
NOSQL	Not Only Structured Query Language
OIDC	OpenID Connect
OneSS	One-Stop-Shop
OPC-UA	Open Platform Communication Unified Architecture
ORC	Optimized Row Columnar
OWL	Ontology Web Language
PACTS	Privacy and security Assessment for Cross-discipline Trust and Self-Sovereignty
ParIS	Participant Information Service
PIA	Protection Impact Assessments
pk	Public key
QA	Quality Assurance
RAM	Reference Architecture Model
RDF	Resource Description Framework
REST	Representational State Transfer
RPA	Robotic Process Automation
SDK	Software Development Kit
sk	Secret key
SPARQL	Protocol and RDF Query Language
SSI	Self-Sovereign Identity
SSI-HE	SSI Homomorphic Capable Framework
STAC	Spatio-Temporal Asset Catalog
STM	Security and Trust Manager
SW	Software
TAI-SDF	Trustworthiness Artificial Intelligence Support Design Framework
TBD	Trunk-based development
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VA	Virtual Assistant
VC	Verifiable Credential
VDR	Verifiable Data Registry
VO	Virtual Organization
WP	Work Package
ZKP	Zero Knowledge Proof

1. EXECUTIVE SUMMARY

This document, D4.1 – “Live document, TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework” is the first deliverable of Work Package (WP) 4 – “TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework”, in which six different TRUSTEE Architectural Building Blocks (ABBs) are discussed, as they were defined in D2.1 – “Live doc conceptualisation, use cases and system architecture V1”:

- TRUSTEE Dashboard (DA).
- TRUSTEE Data Protection Impact Assessment (DPIA).
- TRUSTEE Self-Sovereign Identity Homomorphic capable framework (SSI-HE)
- TRUSTEE Data Lake (DL).
- TRUSTEE Security and Trust Manager (STM).
- TRUSTEE One Stop Shop (OneSS).

The current deliverable defines the ABBs current design and implementation. As this deliverable is a living document, it is intended to serve as a real-time representation of the ongoing status of WP4. Consequently, the subsequent versions of the deliverable will provide updates and enhancements regarding the ABB descriptions and will continue to discuss the ongoing implementation activities and their outputs across the lifecycle of the project.

2. INTRODUCTION

This deliverable is a live document and presents the work carried out inside WP4, under the several Tasks it includes, and its goal is to serve as a real-time representation of the ongoing status of WP4.

For a better understanding of the document, Figure 1 presents the detailed architecture of TRUSTEE as described in D2.1 [1], in which ABBs remarked in red correspond with the ones described in this document, namely: DA, DPIA, SSI-HE, DL, STM and OneSS; while the ones in blue are detailed in the D3.1 [2]: Homomorphic Enabled Data Fusion (HEDF), Artificial Intelligence Models as a Service (AIMaaS), Trustworthiness Artificial Intelligence Support Design Framework (TAI-SDF), Accountable Transactions Recorder (ATR), Knowledge Repository (KR) and Authentication Manager (AM).

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

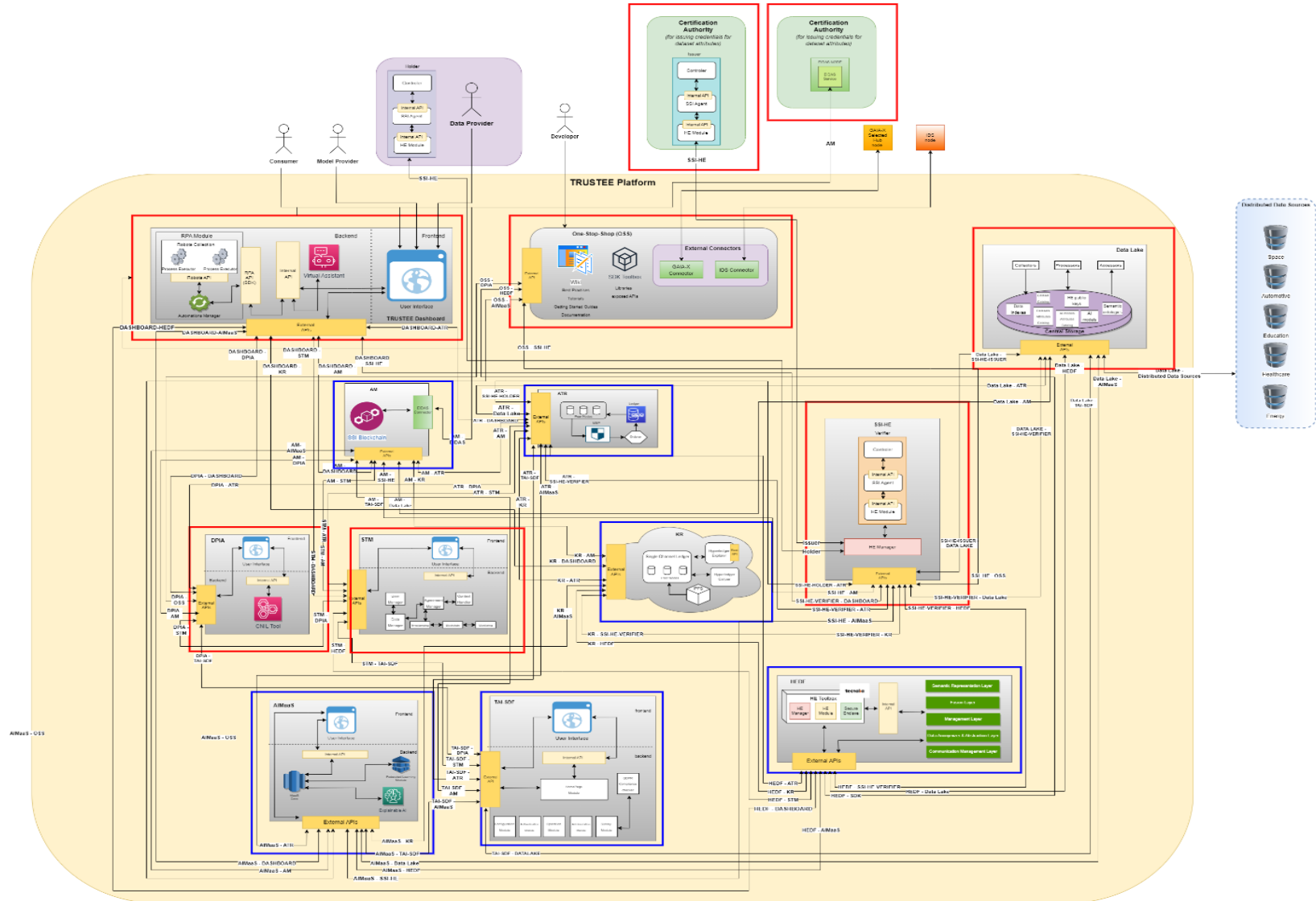


Figure 1: TRUSTEE Detailed architecture. WP3 and WP4 ABBs

Figure 2 represents the relationship between this deliverable (D4.1) and the work done in other WPs in TRUSTEE. D4.1 receives input from WP2 in terms of background, user-centred, technical, legal, and socio-ethical requirements, and architecture while it provides output in terms of elaborating more on technical requirements and implementation progress regarding the ABBs that are being developed under WP4. Furthermore, D4.1 provides output to WP5 through the generation of software (SW) modules that participate in the TRUSTEE Pan-European Pilot Campaign and receives input in terms of feedback for further updates. Finally, WP3 and WP4 are highly interrelated as each of them presents and develops different components that are part of TRUSTEE which interact with each other.

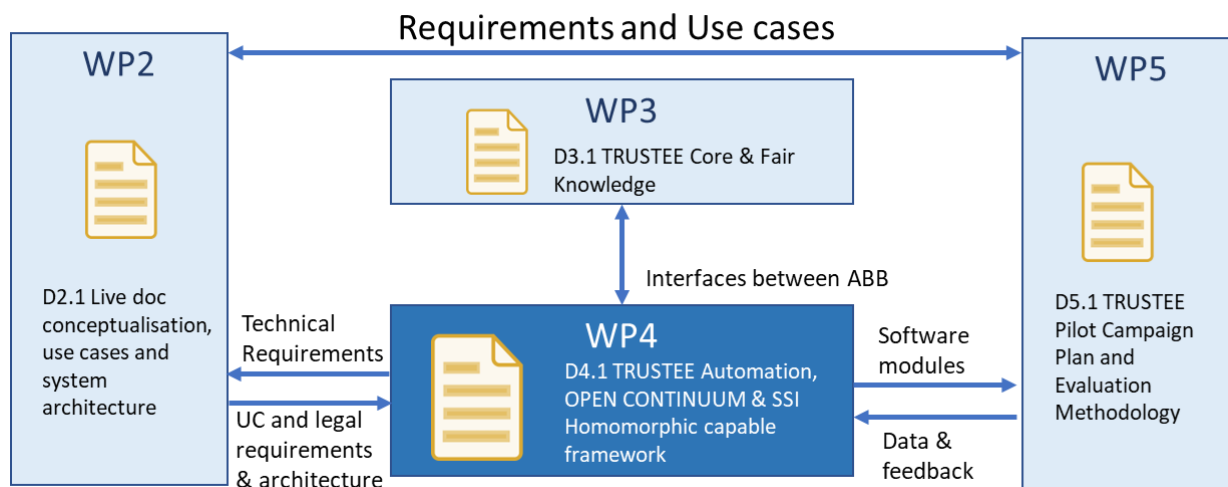


Figure 2: Relationship between D4.1 and other WPs

This section presents the mapping of each WP4 Task to each part of the deliverable (high level chapters), for enabling the reader to navigate through the document.

- Part I refers to Task 4.2 Workflow Automations, Robotic process automation, Dashboard and User interface. This part is mainly dedicated to the DA ABB, which includes three sub-components, namely the Robotic Process Automation (RPA) module, the Virtual Assistant (VA) module, and the primary User Interface (UI) of the TRUSTEE Platform. The DA ABB is considered the point of entry to the TRUSTEE Platform.
- Part II refers to Task 4.1 Privacy impact assessment. This part presents the DPIA ABB.
- Part III refers to Task 4.3 SSI Homomorphic capable Framework, in which the SSI-HE ABB is described.
- Part IV refers to Task 4.5 Data Lake building for cross border data transfer. This part describes the DL ABB.
- Part V refers to Task 4.6 Security and trust manager, in which the STM ABB is presented.
- Part VI refers to Task 4.4 TRUSTEE SDK and Open cloud Continuum API. This part introduces the OneSS ABB concept and the investigation towards different potential external communications between the TRUSTEE Platform and external clouds.

For each chapter (for each ABB) a similar structure has been followed wherever possible in accordance to the nature of the respective ABB and its functionalities and architecture:

- Introduction. This is the introductory section of the ABB, in which the needs and main objectives of the ABB are presented. Additionally, the main innovation is identified.
- Background. This section provides some basic ideas about the technologies considered by the ABB. The idea is to provide the necessary details to better understand the ABB architecture, operation, and implementation. The detailed explanation of all the considered technologies is gathered in D2.1 [1].

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- **Requirements.** This section analyses the current compliance level of each ABB against TRUSTEE user-centred, technical, and legal and socio-ethical requirements defined in D2.1 [1].
- **Architecture.** This section presents the design (high level and low level) of the ABB. The architecture, as well as the operational flows are detailed.
- **Implementation.** This section presents the current implementation level of the ABB at the time of writing in terms of functionalities, operation, and technical details.
- **Validation.** This section introduces any initial validation tests that have been carried out considering the maturity of the respective ABB.
- **Next Steps.** This section presents the next steps to be followed for the development of each ABB, which will be further discussed in upcoming versions of this deliverable.
- **Conclusions.** This section presents a summary of the objective and current status of the ABB.

Furthermore, the deliverable includes the following annexes:

- **Annex 1: Glossary of Terms,** a list of relevant terms with their definitions.
- **Annex 2: Legal Requirements,** gathering the preliminary compliance status with legal requirements gathered in D2.1 [1] for each considered ABB. This information has been included as annex due to its long extension.
- **Annex 3: Socio-ethical Requirements,** gathering the preliminary compliance status with socio-ethical requirements gathered in D2.1 [1] for each considered ABB. This information has been included as annex due to its long extension.
- **Annex 4: DPIA Endpoints,** gathering the endpoints for the RESTful API of DPIA ABB. This information has been included as annex due to its long extension.
- **Annex 5: STM Endpoints,** gathering the endpoint design for different operations and HTTP methods for the STM ABB. This information has been included as annex due to its long extension.
- **Annex 6: DevOps and CI/CD Methodology,** includes a proposal for development, validation, and deployment methodology for mentioned TRUSTEE ABBs.

3. PART I: DASHBOARD

3.1. INTRODUCTION

In this chapter, a comprehensive overview of the TRUSTEE DA ABB, developed under T4.2, will be presented, highlighting its core components, requirements, and implementation details.

3.1.1. NEED AND OBJECTIVES

As it is common for software applications to have a way for the user to interact with, there is need for an UI for the TRUSTEE Platform as it provides a means for the user to interact with TRUSTEE and its components. A dashboard user interface serves as the primary gateway for users to access the TRUSTEE's features and functionality. It enables users to authenticate themselves, input data, navigate through the platform, and view the results or outputs through its communication with the various ABBs of the TRUSTEE Platform. In order to ensure consistency across the visual representation of the TRUSTEE Platform, a set of clear User Experience (UX) guidelines were set and will be followed for the design and implementation of the UI of the DA ABB as well as for the design and implementation of the UI of any other ABB of the TRUSTEE Platform that needs to develop their own UI, in which the end-user will be redirected through the primary UI of the DA ABB.

It is worth noting that, TRUSTEE platform offers several functionalities to the user, which might be overwhelming for novice users with a low level of expertise in technology. To mitigate that, a VA will be utilized to assist the users in navigating through the DA as well as in executing workflow automations in order to relief the end-users from certain difficult to execute or repetitive actions.

Additionally, the DA communicates with various ABBs of the TRUSTEE Platform to enable end-users to access the TRUSTEE's features and offered functionalities. Therefore, an Application Programming Interface (API) for the DA is needed to be developed to enable such intercommunication.

Considering the fulfilment of the afore mentioned needs, the objectives that need to be achieved under T4.2 are:

- Define the UI/UX design guidelines and recommendations for the TRUSTEE Platform.
- Develop the DA and its primary UI by following the design guidelines and principles defined.
- Develop the VA that will assist the user in interacting with the TRUSTEE Platform.
- Develop the TRUSTEE RPA framework for offering end-users the ability to automate routine procedures with the help of the TRUSTEE VA.
- Build automation techniques for enabling faster interaction with the TRUSTEE Platform.

Therefore, the objectives of the DA ABB are as follows:

- Enable the user to navigate through the TRUSTEE Platform and access the platform's various features and offered functionalities in a consistent and user-friendly way.
- Assist end-users in interacting with the TRUSTEE Platform through the VA and TRUSTEE RPA framework modules.

3.1.2. INNOVATION

While VAs and RPA technologies, separately, are quite mature and well applied in the industry, it is uncommon to see an RPA powered VA conversational agent, being able to perform workflow automations and assist the user with repetitive tasks. Furthermore, the application of such technologies' combination in federation of data spaces is quite limited. For the above-mentioned reasons, TRUSTEE aims to be an integrated platform that utilizes a VA for user guidance and assistance while incorporating automation capabilities.

3.2. BACKGROUND

In this section, a high-level explanation of the technologies used for the development of the DA ABB is presented with the aim to lay a better understanding of the inner components of the DA.

3.2.1. USER INTERFACE DESIGN

The DA includes the primary UI for interacting with the TRUSTEE Platform, accompanied by the VA and RPA modules. Designed as a cloud-based web application, the DA enables users to leverage the full capabilities of the TRUSTEE Platform.

Key functionalities facilitated/visualized by the UI of the DA include:

- Sign-up and Login procedures, through communication with the AM ABB.
- Search and light computational operations over encrypted data, through communication with the SSI-HE and HEDF ABBs.
- Results Visualization regarding data querying and operations/computations by communication with the HEDF ABB.
- Dataset registration as part of the Distributed Data Sources of TRUSTEE by redirecting the user to the Graphical User Interface (GUI) of the STM ABB and by communicating with the DL ABB for providing description details about the dataset at hand.
- An overview of the user's registered datasets description, if they have registered any, by communicating with the DL ABB.
- Machine Learning (ML) / Artificial Intelligence (AI) Algorithm selection and use, by redirection of the user to the GUI of the AIMaaS ABB.
- Monitoring of activity, by redirection of the user to the GUI of the Monitoring Tool (MT) that will be part of the ATR ABB.
- Agreements Management, by redirection of the user to the GUI of the STM ABB.
- Privacy Impact Assessments monitoring and management, by redirection of the user to the GUI of the DPIA ABB.
- Exploring knowledge generated through the use of the TRUSTEE Platform and offered for public access by other users of the Platform through communication with the KR ABB.
- Navigation Assistance and Workflow Automation through the use of the RPA and VA modules of the DA ABB.

The main UI of the TRUSTEE Platform offered by the DA is accessible by the following TRUSTEE Personas: Data Provider, Model Provider, and Consumer (including both data and model consumers). Through the main UI of the DA the end-users belonging to these Personas groups can explore the functionalities of the TRUSTEE Platform as follows:

- Data provider is able to navigate and register their data as part of the Distributed Data Sources of TRUSTEE, sign and monitor the agreements for the provided datasets by their redirection to the STM ABB, and monitor the use of their datasets within TRUSTEE as well as their own history of activity by redirection to the UI of the ATR ABB.
- Model Provider is able to share and/or provide their AI/ML models with the TRUSTEE Platform by their redirection to the GUI of the AIMaaS and utilizing the functionalities offered by this block, sign agreements and monitor the agreements for the provided/shared models by their redirection to the STM ABB, and monitor the use of their models within TRUSTEE as well as their own history of activity through the GUI of the ATR ABB.
- Consumer, i.e., Data Consumer and Model Consumer, respectively, is able to consume data and/or AI/ML models, perform search queries over homomorphically encrypted data as well as lightweight computations, sign agreements for the requested queries, computations, and use of AI/ML models, perform data privacy impact assessments for the requested computations through their redirection to the GUI of the DPIA ABB, and view their own history of activity through their redirection to the GUI of the ATR ABB.

Overall, the UI of the DA serves as a comprehensive and user-friendly interface for users to interact with the TRUSTEE Platform. It enables secure access, efficient data operations, ML/AI utilization, result visualization, and workflow automation. More details regarding the structure of the main UI of the DA and the various information presented in each of its pages are presented further below.

3.2.2. ROBOTIC PROCESS AUTOMATION

RPA is an automation technology that makes use of software robots or "bots" to automate repetitive tasks within business processes. RPA software robots usually mimic human actions, using the computers mouse and keyboard or other input devices to interact with various digital systems, applications, and data sources.

3.2.2.1. BENEFITS

RPA offers several benefits to organizations, including increased efficiency, improved accuracy, enhanced productivity, reduced operational costs, and faster processing times. By automating routine and manual tasks, RPA allows employees to focus on more value-added activities.

3.2.2.2. SOFTWARE ROBOTS

RPA software robots can interact with applications and systems through user interfaces, screen scraping, APIs, or other integration methods. They are able to follow predefined rules and workflows, handle exceptions, and make decisions based on predefined logic. There are two distinct types of automations that the robots usually operate on, attended automation and unattended automation. When RPA software robots operate in an attended mode, they typically work alongside human users, where the automation cannot proceed without the intervention of a human (e.g., providing a passphrase). On the other hand, when the robots are performing unattended automation, there is no need for human interaction, because they typically run in the background, executing a predefined static automation process.

3.2.2.3. CHALLENGES AND CONSIDERATIONS

While RPA offers numerous benefits, it also presents challenges and considerations. These include selecting suitable processes for automation, managing change and user engagement, ensuring data security and privacy, and addressing scalability, flexibility, and bot management.

3.2.3. VIRTUAL ASSISTANT

A VA is a type of conversational agent and can be a simple chat-bot that follows a predefined flowchart or decision tree structure to interact with users. The decision tree consists of a series of questions and answers that guide the conversation based on user inputs or specific conditions. The chat-bot's decision tree typically starts with an initial question or prompt to engage the user. Based on the user's response, the chat-bot follows a predetermined path in the decision tree to provide appropriate responses or ask further questions. This process continues until the chat-bot reaches a conclusion or completes the conversation.

3.2.3.1. USE

These types of chat-bot agents are commonly used in cases where there is a defined set of questions or issues that users commonly encounter. They can be deployed in customer support, Frequently Asked Questions (FAQs), troubleshooting guides, information retrieval systems, and other domains where there is a need for structured interactions.

3.2.3.2. LIMITATIONS

Some limitations of conversational agents of this type are that they are typically not capable of handling complex or open-ended conversations, as they rely on predefined questions and answers. If the user deviates from the expected flow or asks a question that is not in the decision tree, the agent may struggle to provide a relevant response. Addressing this limitation, TRUSTEE's VA will not support free-text form of user input, rather the user's interaction with the agent will be in form of selecting one of the multiple valid questions and responses that the agent presents.

3.3. REQUIREMENTS

In this chapter, the compliance of the DA ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding how the aforementioned requirements are addressed during the development and by the implementation of the DA and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements are presented. As the implementation of the DA progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

User-centred requirements were identified through the UX Research that was conducted under T2.3. These requirements were reported in D2.1 [1], while they were also utilized in order to extract technical, functional and non-functional, requirements, that were also outlined in D2.1. Table 1 below presents the interplay between user-centred requirements and technical requirements, that is the functional and non-functional requirements that address each user-centred requirement. The user-centred requirement ID is used to reference the respective requirement reported in D2.1 [1] while the requirement's description is also provided alongside whether the DA is currently fully or partially compliant with the user-centred requirement, i.e., indicating whether technical requirements have been identified in order to address the respective user-centred requirement. The ID of the technical requirement, as reported in D2.1 [1],

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

that is addressing each user-centred requirement is also provided in the last column of the table. Additionally, it is possible than more than one technical requirement addresses a single user-centred requirement.

Table 1: DA – User-centred requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
USR-014	TRUSTEE should offer APIs or portals for some of its functionalities to be used	Yes	Req-DA-FUNC-4
USR-047	TRUSTEE should be free, easy-to-use, and offer user-friendly statistical tools.	Yes	Req-DA-FUNC-1 Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-NFUNC-2
USR-048	TRUSTEE should support the ability to automatically add data and include extra metadata fields to these data if needed	Yes	Req-DA-FUNC-8
USR-049	TRUSTEE should perform data querying through user-friendly, simple, and error-free programming languages	Yes	Req-DA-FUNC-1 Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-NFUNC-2 Req-DA-NFUNC-3
USR-050	TRUSTEE should utilize easy-to-use filters in data querying	Yes	Req-DA-FUNC-1 Req-DA-NFUNC-2 Req-DA-NFUNC-3
USR-051	TRUSTEE should allow to perform complex queries without the knowledge of a programming language	Yes	Req-DA-FUNC-1 Req-DA-FUNC-9
USR-052	TRUSTEE should extract knowledge without complex analysis methods and tools	Yes	Req-DA-FUNC-4 Req-DA-FUNC-9
USR-054	TRUSTEE should have easy access and permissions	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4
USR-055	TRUSTEE should provide automation techniques	Yes	Req-DA-FUNC-2 Req-DA-FUNC-6 Req-DA-FUNC-8 Req-DA-FUNC-9 Req-DA-FUNC-10
USR-056	TRUSTEE should offer the ability for automating procedures regarding data consumption and other data processes	Yes	Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-10
USR-057	TRUSTEE should be an efficient and highly secure platform	Yes	Req-DA-NFUNC-1 Req-DA-NFUNC-2
USR-058	TRUSTEE should enable users to access data-catalogues, tutorials, and analysis outputs easily	Yes	Req-DA-FUNC-6 Req-DA-FUNC-9 Req-DA-NFUNC-3

USR-059	TRUSTEE should be user-friendly	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-5 Req-DA-FUNC-6 Req-DA-FUNC-7 Req-DA-NFUNC-2 Req-DA-NFUNC-3
USR-060	TRUSTEE should show the types of data that are available for access	Yes	Req-DA-FUNC-4 Req-DA-FUNC-11
USR-061	TRUSTEE should have access to support regarding platform functioning	Yes	Req-DA-FUNC-4 Req-DA-FUNC-5 Req-DA-FUNC-6 Req-DA-FUNC-8 Req-DA-FUNC-9 Req-DA-FUNC-10
USR-062	TRUSTEE should have a user-friendly validation and presentation of data analysis results	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-NFUNC-2
USR-063	TRUSTEE should have example applications and tutorials for what is offering to be used by end-users	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-12
USR-082	TRUSTEE could provide information on the way a website/portal collects data from various sources	Yes	Req-DA-FUNC-6
USR-113	TRUSTEE could overcome the issues relating to automatic setup of data flows	Yes	Req-DA-FUNC-7 Req-DA-FUNC-10
USR-118	TRUSTEE could enable to reach and communicate with more specialists in a specific field	Yes	Req-DA-FUNC-6
USR-050	TRUSTEE should utilize easy-to-use filters in data querying	Yes	Req-DA-FUNC-1 Req-DA-NFUNC-2 Req-DA-NFUNC-3
USR-051	TRUSTEE should allow to perform complex queries without the knowledge of a programming language	Yes	Req-DA-FUNC-1 Req-DA-FUNC-9
USR-052	TRUSTEE should extract knowledge without complex analysis methods and tools	Yes	Req-DA-FUNC-4 Req-DA-FUNC-9
USR-054	TRUSTEE should have easy access and permissions	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4
USR-055	TRUSTEE should provide automation techniques	Yes	Req-DA-FUNC-2 Req-DA-FUNC-6

			Req-DA-FUNC-8 Req-DA-FUNC-9 Req-DA-FUNC-10
USR-056	TRUSTEE should offer the ability for automating procedures regarding data consumption and other data processes	Yes	Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-10
USR-057	TRUSTEE should be an efficient and highly secure platform	Yes	Req-DA-NFUNC-1 Req-DA-NFUNC-2
USR-058	TRUSTEE should enable users to access data-catalogues, tutorials, and analysis outputs easily	Yes	Req-DA-FUNC-6 Req-DA-FUNC-9 Req-DA-NFUNC-3
USR-059	TRUSTEE should be user-friendly	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-5 Req-DA-FUNC-6 Req-DA-FUNC-7 Req-DA-NFUNC-2 Req-DA-NFUNC-3
USR-060	TRUSTEE should show the types of data that are available for access	Yes	Req-DA-FUNC-4 Req-DA-FUNC-11
USR-061	TRUSTEE should have access to support regarding platform functioning	Yes	Req-DA-FUNC-4 Req-DA-FUNC-5 Req-DA-FUNC-6 Req-DA-FUNC-8 Req-DA-FUNC-9 Req-DA-FUNC-10
USR-062	TRUSTEE should have a user-friendly validation and presentation of data analysis results	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-NFUNC-2
USR-063	TRUSTEE should have example applications and tutorials for what is offering to be used by end-users	Yes	Req-DA-FUNC-2 Req-DA-FUNC-3 Req-DA-FUNC-4 Req-DA-FUNC-12
USR-082	TRUSTEE could provide information on the way a website/portal collects data from various sources	Yes	Req-DA-FUNC-6

Table 2 presents the preliminary results of the investigation regarding the compliance of the DA with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether DA complies with the requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 2: DA – Functional and non-functional technical requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
Req-DA-FUNC-1	The TRUSTEE Platform COULD offer a filter functionality for AI/ML models/algorithms and for data domain selection	Yes	DA will develop a dataset search page where a filtering process will be visualized while the functionality is offered by the SSI-HE and HEDF ABBs.
Req-DA-FUNC-2	The TRUSTEE Platform SHOULD offer visualization of the results stemming from the users' requests (e.g., search queries, etc.)	Yes	DA will develop a results page where the user's queries results will be presented.
Req-DA-FUNC-3	The TRUSTEE Platform COULD provide the user with the ability to request the history of their activity.	Yes	DA will provide a page for the user to preview their activity history.
Req-DA-FUNC-4	The TRUSTEE Dashboard MUST visualize the APIs of the building blocks that need to expose information to the Dashboard (e.g., ATR, STM, HE Data Fusion for data computation results, etc.)	Yes	DA will provide pages per ABB that need to visualise their API, such as DPIA, STM, AIMaaS, ATR (Monitoring Tool).
Req-DA-FUNC-5	The TRUSTEE Virtual Assistant SHOULD interact with the user through a Question-Answer model.	Yes	VA will develop a decision tree with the questions and answers encoded into it.
Req-DA-FUNC-6	The TRUSTEE Virtual Assistant MUST be able to assist the user navigate into the TRUSTEE dashboard	Yes	VA will offer navigation options to the user through the decision tree.
Req-DA-FUNC-7	The TRUSTEE RPA Robots SHOULD be able to get input from the user.	Yes	RPA Robot will halt execution to get any necessary information from the user.
Req-DA-FUNC-8	The TRUSTEE Virtual Assistant SHOULD be able to assist the user to register their data as part of the Distributed Data Sources of TRUSTEE.	Yes	A Data registration workflow automation procedure will be created.
Req-DA-FUNC-9	The TRUSTEE Virtual Assistant SHOULD be able to help users construct complex queries in a visual or text based way.	Yes	TRUSTEE VA will provide options to the user to construct complex queries.
Req-DA-FUNC-10	The TRUSTEE Virtual Assistant SHOULD provide automated procedure regarding data consumption.	Yes	A Data consumption workflow automation procedure will be created.
Req-DA-FUNC-11	The TRUSTEE Dashboard SHOULD show the types of data that are available, through data-catalogues.	Yes	DA will display the types of data that are available in the "Dataset search" page, where the Consumer may search for datasets that fit their search criteria.
Req-DA-FUNC-12	The TRUSTEE Dashboard SHOULD provide the user a way to access the tutorials and example applications.	Yes	A "Guide" page will be developed in the main UI of the DA where tutorials and example applications will be presented.

Req-DA- NFUNC- 2	The TRUSTEE Dashboard SHOULD display information clearly, consistently, and efficiently, in a user-friendly way	Yes	DA will be designed according to the UI/UX design guidelines and recommendations.
Req-DA- NFUNC- 3	The TRUSTEE Dashboard SHOULD make sure that all essential information is immediately accessible.	Yes	A “Home page” will be developed in the main UI of the DA where essential information will be immediately accessible
Req-DA- NFUNC- 4	The users of the TRUSTEE Platform MUST be able to securely register and authenticate themselves through the dashboard	Yes	A “Login” and a “Sign-up” page will be developed in the main UI of the DA for visualizing the process of secure authentication and authorization when a user requests to enter the TRUSTEE Platform. The functionalities of login and sign-up are offered by the AM ABB.

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

3.4. ARCHITECTURE

3.4.1. HIGH LEVEL

The DA is split in two major modules, each of these has its internal components as it can be seen in Figure 3. At the top left corner, the Dashboard package is depicted, which contains the UI of the DA as well as the VA and the External API. This package is meant to be deployed in a cloud infrastructure and the user will have access to it through a web browser.

At the bottom right corner, the second package depicted is the RPA component, which houses all the necessary components that enable the VA to perform automated workflows. These components comprise the RPA Robot, the Automation Manager, and the RPA API, which enables the intercommunication of the RPA with the VA. The aforementioned package is developed to be installed and ran locally on the user’s hardware.

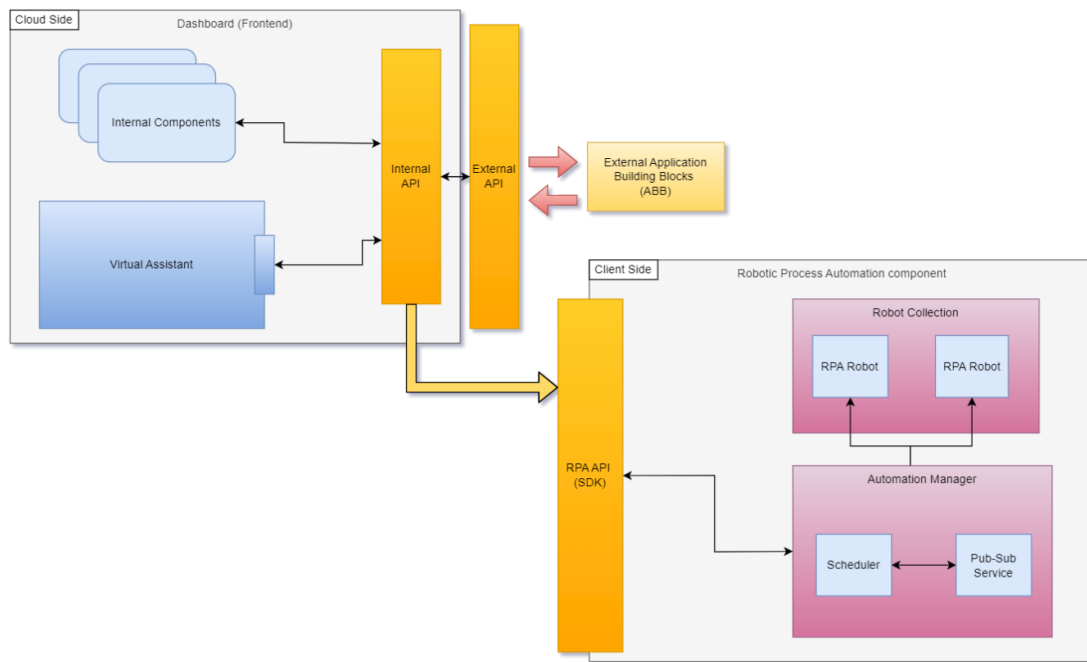


Figure 3: DA - High level architecture diagram

3.4.1. LOW LEVEL

In this section, the architecture of the DA ABB will be presented in a greater detail. As can be seen in Figure 4, this section is dedicated to explain in detail the inner modules and the functional principles of the various sub-components of the DA ABB, namely the main UI, the RPA module, the VA module, and the external API.

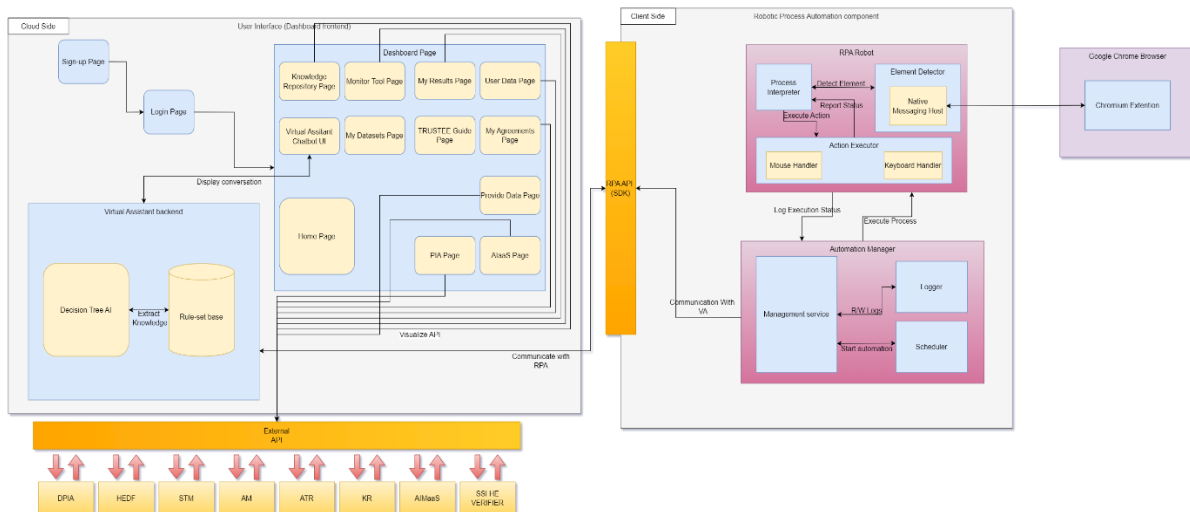


Figure 4: DA - Low level architecture diagram

3.4.1.1. DASHBOARD UI (FRONTEND) COMPONENT

As described earlier in this section, the frontend of the DA of TRUSTEE includes the main UI, the VA, and the external API that facilitates intercommunication of the DA with the various ABBs of the TRUSTEE Platform. In the following sub-sections, more architectural details of these modules are provided.

MAIN UI VIEWS

As it can be seen in Figure 4, the main UI of the DA comprises several pages that serve distinct purposes in facilitating user interaction with the TRUSTEE platform:

- Login page
- Sign-up page
- Home page
- Use Data page
- Provide Data page
- AIaaS page
- My Results page
- My Datasets page
- Privacy Impact Assessment page
- My Agreements page
- Monitor Tool page
- Knowledge Repository page
- TRUSTEE Guide page

The Login and Sign-up pages provide essential functionality for user registration and authentication. These pages, communicating with the AM ABB, ensure secure access to the TRUSTEE Platform. The Login page verifies registered users' credentials, granting them access to their personalized accounts, while the Sign-up page guides new users through the registration process, collecting necessary information and creating new accounts within the platform.

The Home page serves as a central hub within the DA, featuring graphics that depict the platform's functionalities. It provides users with an intuitive interface to navigate and explore the various features and services offered by the platform.

The Use Data page, through communication with the SSI-HE, KR, STM, and DL ABBs, enables users to search and explore available datasets within the TRUSTEE Platform based on specific criteria or requirements by providing a list of available filters that can be used to facilitate search. Moreover, the DA allows users to interact with the encrypted datasets that satisfy their search criteria while ensuring privacy and security, by visualizing the available computations that can be performed over the datasets stemming from the search process and by informing the user that any necessary agreements should be signed prior to these data computations within the TRUSTEE Platform. Additionally, through the Use Data page and when the user submits their own unique search query, they are provided with the option to explore similar and available search queries and their results in the KR.

Through the Provide Data page, the user can initiate the process of registering a dataset to be part of the Distributed Data Sources of TRUSTEE. The DA guides the user through this process by communicating with the STM and DL ABBs, providing a user-friendly way for the user to navigate through the process.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The DA provides also the user with the ability to be redirected to the UI of the AIMaaS in order to either explore and/or use AI Models that are offered through the TRUSTEE Platform or provide their AI Models to TRUSTEE.

The DA also visualizes the results stemming from performing computations over encrypted data by communicating with the HEDF ABB. These results are presented in the My Results page of the DA in an efficient, comprehensive, and user-friendly way. It also visualizes a complete overview of the dataset(s) that the user has registered as part of the Distributed Data Sources of TRUSTEE in the My Datasets page, offering the user complete control over his datasets.

The Privacy Impact Assessment page redirects users to the GUI of the DPIA ABB, which offers information and guidance regarding privacy implications and considerations associated with using the TRUSTEE Platform. It assists users in conducting privacy impact assessments and ensures compliance with relevant privacy regulations.

Furthermore, the My Agreements page, redirecting users to the UI of the STM ABB, facilitates the management of various agreements and contracts associated with the usage of the TRUSTEE Platform. Users can review, accept, or modify agreements through this page, ensuring transparency and compliance with legal obligations.

The Monitor Tool page redirects the user to the GUI of the ATR ABB enabling the users to view a history of their activity in the TRUSTEE Platform and, in case they are a Data Provider, to monitor the actions that have been performed over the dataset(s) they have registered as part of the Distributed Data Sources of TRUSTEE offering transparency and accountability.

Furthermore, the Knowledge Repository page, via communication with the KR ABB, acts as a knowledge resource hub of TRUSTEE. It provides users with access to valuable knowledge, information, and search and computational results acquired from the platform's use. This feature enhances collaboration and knowledge sharing among users, promoting efficient utilization of the TRUSTEE Platform.

The TRUSTEE Guide page that is part of the UI of the DA offers guidance to the user and additional information regarding the use of the TRUSTEE Platform enhancing the user-friendliness of the platform.

Finally, Virtual assistant's view is a chat-bot interface component that appears on the bottom right corner of every page of DA.

These pages collectively form the UI of the DA, providing users with a comprehensive and user-friendly interface to interact with the platform's functionalities. Through seamless communication with the corresponding ABBs of the TRUSTEE Platform, the DA ensures secure authentication, efficient dataset search, use, and provision, access to knowledge resources, privacy impact assessment and agreement management, contributing to an enhanced user experience and effective utilization of the TRUSTEE Platform.

VIRTUAL ASSISTANT

The TRUSTEE Virtual assistant is conceptualized as a simple but functional conversational agent that operates by using a predefined decision tree containing multiple possible conversations that can take place between the user and the agent. A simple chat-bot like interface will be used as the UI of the VA, where the questions and the possible answers will be presented to the user so the conversation can take place.

Part of the VA component, as seen in Figure 4, is the Decision Tree AI, which is the core module of the assistant, and the Knowledge base that contains the actual decision tree encoded in such a way so it can be used by the Decision Tree AI module. The Decision Tree AI module handles the conversations with the user and performs workflow automations

with the help of the RPA component. In addition, the Knowledge Base contains the workflow automation processes that will be possible to do using the agent.

EXTERNAL API

The external API of the DA serves as a communication bridge between the DA and the other ABBs of the TRUSTEE, enabling the frontend of the DA to send requests and receive responses from the APIs of the other ABBs that either communicates with or redirects the user to.

The categorization of ABBs based on the type of interaction with the frontend is as follows:

- i. Redirection:
 - DPIA
 - STM
 - AIMaaS
 - ATR
- ii. Request/Response Communication:
 - AM
 - KR
 - SSI-HE
 - HEDF
 - DL

3.4.1.2. ROBOTIC PROCESS AUTOMATION COMPONENT

The TRUSTEE RPA component is a set of components that enables the VA to perform human-like actions on the user's computer automatically, making the use of the TRUSTEE platform easier and faster, especially when the user is not that familiar with the technology. For this to happen, the RPA Robot, the Automation Manager, the Chromium extension, and the RPA API are developed as described in Figure 4.

RPA ROBOT

As the name implies, the RPA Robot is responsible for understanding and executing workflow automation procedures. This is plausible through the help of the internal components of the robot.

The Process interpreter is the core component of the robot and its main functionality is to compile the automation process into an executable format and setup the rest of the components to start the automation. The interpreter reads the process and executes the actions described in the process one by one. While executing, if an action is related to a GUI element, the interpreter provides the recorded data from the process to the Element Detector module, which is responsible to detect where this particular element exists on the user's desktop and return to the interpreter the position of that element. Lastly, for the robot to be able to simulate the user's input, the Action Executor module depending on the action that need to be executed (e.g. a Mouse Click), simulates the input using the appropriate handler.

AUTOMATION MANAGER

The Automation manager components is responsible for the robot management and communication with the rest of the ABBs of the TRUSTEE Platform. More specifically, the Logger module handles the status messages that the robot

generates throughout its execution and logs them to a file for debugging purposes. The Scheduler module is responsible to properly Start, Stop, and Pause the robot execution as well as ensure the security of the component by blocking any unauthorized access to the robot. Lastly, the Management service module is always running in the background and its purpose is to handle the communication between the Scheduler/Logger and the rest of the ABB components. The communication technology used is Publish-Subscribe messaging (Pub-Sub) using the Advanced Message Queuing Protocol (AMQP) [3] protocol and the RabbitMQ broker as an intermediate service. The RabbitMQ is considered the RPA API because it is used to convey information to any subscribed component.

3.5. IMPLEMENTATION

In this section, a concise and comprehensive account of how the DA was executed is presented. It consists of four key subsections. The functional description provides a high level description of the main functionalities of DA. The operational description highlights the software's functionality, how the DA works from inside and the user experience in real-world scenarios. The technical description offers insights into the implementation details, the input and output data models, the GUI of the ABB and the API that enables the communication with other ABBs. Lastly, the technical details delve deeper into the software's development and deployment requirements and the technologies needed to do so.

3.5.1. FUNCTIONAL DESCRIPTION

The implementation of the main UI of the DA will utilize VueJS framework enabling the creation of reusable and modular components (pages). To handle the routing within the DA, the Vue Router package¹ is considered to be employed. The implementation will leverage the Vue Router's navigation guards feature, which allows developers to define custom logic before entering or leaving a specific route. This feature enables the implementation of authentication and authorization checks, data validation, and other necessary operations to ensure the security and integrity of the DA. Additionally, to store the user's access token and other relevant information, Pinia², a state management library for Vue.js, will be used.

3.5.1.1. RPA

In this section the functional description of the RPA component is presented.

ELEMENT DETECTION

The RPA Robot uses Microsoft UIAutomation³ accessibility framework to read the user's desktop and get information about the open windows, the state of these windows, and the graphical elements (Buttons, Text Views, Labels etc.) that each application window displays. When the robot needs to find an element that is described in the action, the robot gets all the associated information about this element (Automation ID, Name, Label, Class etc.) and recursively searches each application window in the desktop. Then, when the robot finds a control that matches perfectly with the

¹ <https://v3.router.vuejs.org/installation.html>

² <https://pinia.vuejs.org/>

³ <https://www.microfocus.com/documentation/silk-test/200/en/silktestworkbench-help-en/GUID-C11F8AD3-4E53-48CF-96C2-ABD817A9D8FC.html>

requested data, it calculates its client area in a rectangular form and the clickable point in that area. Using these coordinates, the robot continues on executing the action (Mouse click, Type keystrokes etc.).

ACTION EXECUTION

For the actions to get executed, the robot uses the kernel API, which in our case is Windows API⁴, to simulate user input on a lower level. Calling the appropriate functions of the Windows API, the robot is able to move the mouse to a certain position, simulate a button press, change the keyboard focus and also simulate key presses and releases.

3.5.1.2. VIRTUAL ASSISTANT

The VA is a chat-like interface created with VueJS⁵ and a decision tree that defines its logic. The user interacts with the VA using buttons in the chat that clearly define the provided options to the user. When the user clicks on an option, the VA runs the decision tree using the Depth First Search (DFS) algorithm, determines the answer to that option and the action that might need to perform. In TRUSTEE, the three main actions that the VA will be able to perform are as follows: assist the user to navigate into the TRUSTEE platform; perform workflow automations; and aid the user by providing answers to some FAQs.

NAVIGATION

The TRUSTEE VA uses the VueJS router⁶ to perform navigation redirections inside the main UI of the DA and vanilla JavaScript to redirect to other components of the TRUSTEE Platform.

WORKFLOW AUTOMATION

The VA uses the RPA component to perform certain workflow automations. Upon user's request, the VA will craft a specialized access token and will publish a message to a specific AMQP queue that contains this access token and the encoded process file that describes the automation. After that, the VA will wait for validation messages from the Automation Manager. Once the 'Finished' message arrives, the message is displayed to the user and the automation stops.

SUPPORT

The VA will have encoded in its decision tree some FAQs alongside with their answers. The VA will display all the available questions to the user. If the user clicks on a question, the VA will display its answer.

3.5.1.3. TECHNOLOGY ENABLERS

UI DESIGN

⁴ <https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list>

⁵ <https://vuejs.org/>

⁶ <https://router.vuejs.org/>

Figma⁷ is a leading design and prototyping tool, offering a range of features and advantages that make it a preferred choice among designers and design teams for UI design. While there are several applications similar to Figma, such as Sketch⁸, Adobe XD⁹, and InVision¹⁰, Figma stands out for several reasons¹¹. Firstly, Figma's cloud-based platform enables collaboration allowing multiple team members to work simultaneously on a design project. This collaborative environment enhances productivity by reducing communication difficulties. Secondly, Figma's web-based nature eliminates the need for installation, making it easily accessible across different operating systems and devices. Additionally, Figma offers a comprehensive set of design tools and plugins, providing the ability to create visually interactive wireframes with additional functionality. Overall, Figma's powerful features, collaborative capabilities, cross-platform accessibility, and user-friendly interface make it the best choice for designing wireframes, prototypes, and user interfaces compared to other design tools.

FRONTEND DEVELOPMENT

There are many frameworks to choose for frontend development. The most popular of them are React¹², Angular¹³, and Vue.js¹⁴. For the development of the DA, the Vue.js framework has been selected as it is a powerful framework encompassed by a wide range of features^{15,16}. With its component-based architecture, Vue.js allows for creation of reusable and modular UI components, enhancing code reusability and maintainability. Its reactive data binding capabilities ensure real-time updates, resulting in a smooth and dynamic user experience. Vue.js benefits from extensive documentation and an active community, supporting efficient development and offering solutions to common development obstacles.

APPLICATION PROGRAMMING INTERFACE (API) DEVELOPMENT

7 <https://www.figma.com/>

8 <https://www.sketch.com/>

9 <https://adobexdplatform.com/>

10 <https://www.invisionapp.com/>

11 <https://www.angularminds.com/blog/article/comparison-between-sketch-vs-figma-vs-adobe.html>

12 <https://react.dev/>

13 <https://angular.io/>

14 <https://vuejs.org/>

15 <https://www.browserstack.com/guide/angular-vs-react-vs-vue>

16 <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>

Considering the implementation of the external API for the DA, there are several frameworks available, including FastAPI¹⁷, Spring Boot¹⁸, Express.js¹⁹, Django²⁰, and Flask²¹. Among those, FastAPI is a commonly preferred framework due to its ability of leveraging asynchronous programming and a robust event loop that ensures efficient handling of concurrent requests. Furthermore, FastAPI enforces type safety and automatic data validation through Python's type hinting system that enhances the reliability and integrity of the API. The user-friendly syntax, automatic documentation generation, and built-in security features of the FastAPI's framework make the development fast and secure. Overall, FastAPI's performance, reliability, ease of use, and built-in features make it the best fit for implementing the DA's external API for communicating with the various ABBs of the TRUSTEE Platform.

3.5.2. OPERATIONAL DESCRIPTION

The Data Provider, Model Provider, and Consumer (both data and model consumer) TRUSTEE Personas have access to the Login, Sign-up, and Home pages of the primary UI, as well as to the VA of the DA and utilize workflow automations offered by the RPA module of the DA through the VA. Data Provider, Data Consumer, Model Provider, and Model Consumer have different goals that they aim to achieve through the use of the TRUSTEE Platform, for which they use different pages of the primary UI of the DA and are redirected to any subsequent UI/GUI of the other ABBs when needed.

The use case flow for a TRUSTEE Data Consumer and Data Provider aiming to consume data registered as part of the Distributed Data Sources of TRUSTEE or register their datasets as part of the Distributed Data Sources of TRUSTEE, respectively, are presented through three visual representations for providing further clarity regarding user navigation in the TRUSTEE Platform through the main UI of the DA: a sequence diagram, a flow diagram, and a sequence of mock-ups included in the UI Mock-ups subsection inside the Technical Description chapter of this document that is presented further below. These diagrams provide a clear overview of the step-by-step process within the DA, showcasing the sequence of actions and interaction involved in utilizing the TRUSTEE platform as a Data Consumer or a Data Provider. The sequence diagram visually depicts the chronological order of events and the communication between different ABBs, while the flow diagram illustrates the logical flow of activities and decisions throughout the process at hand. In upcoming versions of this deliverable, namely D4.2 and D4.3, the sequence and flow diagrams as well as the sequence of mock-ups illustrating the processes followed are subject to further updates and enhancements.

The sequence and flow diagrams below include updates regarding the blocks that need to interface for the process to be performed and completed compared to the initial Basic Use Case Flows that are depicted in D2.1 [1], thus portraying the current status and progress of the project and they are subject to further enhancements in upcoming versions of this deliverable as the project continues to evolve. These updates will also be used to update the TRUSTEE detailed architecture and the Basic Use Case Flows in future versions of D2,1, namely D2.2.

17 <https://fastapi.tiangolo.com/>

18 <https://spring.io/projects/spring-boot>

19 <https://expressjs.com/>

20 <https://www.django-rest-framework.org/>

21 <https://flask.palletsprojects.com/en/2.3.x/>

3.5.2.1. TRUSTEE DATA CONSUMER USE CASE FLOW

The TRUSTEE Data Consumer Use Case Flow is described in Figure 5, Figure 6 and Figure 7. As Data Consumers enter the TRUSTEE platform, they are prompted to authenticate their identity through the Login or Sign-up page. Upon successful authentication, the user is redirected to the Home Page, where all the functionalities of the platform can be accessed and/or are visualized. Subsequently, the Data Consumer proceeds to the Use Data page, where the available search attributes that can be used as search filters are displayed. After users enter their criteria the available datasets that match the specified criteria are displayed and the user can select one or more datasets to fuse.

Once the desired dataset(s) have been selected, users can explore the assortment of available computations and select the specific computation they wish to perform. Prior to initiating the computation, Data Consumers are redirected to the My Agreements page accessed through the STM's UI, where they are required to review and sign the requisite agreement to proceed with the computation process. Data Consumers should also perform a data privacy impact assessment and General Data Protection Regulation (GDPR) compliance check for the operations they aim to perform over data, which are offered through the GUI of the DPIA ABB and the GUI of the TAI-SDF ABB, respectively. The latter processes are not showcased in the following diagrams as they are considered to be initiated by the STM ABB and not the DA. After the successful signing of the agreement, the computation process starts. When the computation is complete, the data consumer can access the results through the Results page. Finally, the data consumer has the option to keep these results within the KR, where they become available and accessible to all TRUSTEE users.

TRUSTEE Data Consumer Use Case Sequence Diagram part 1

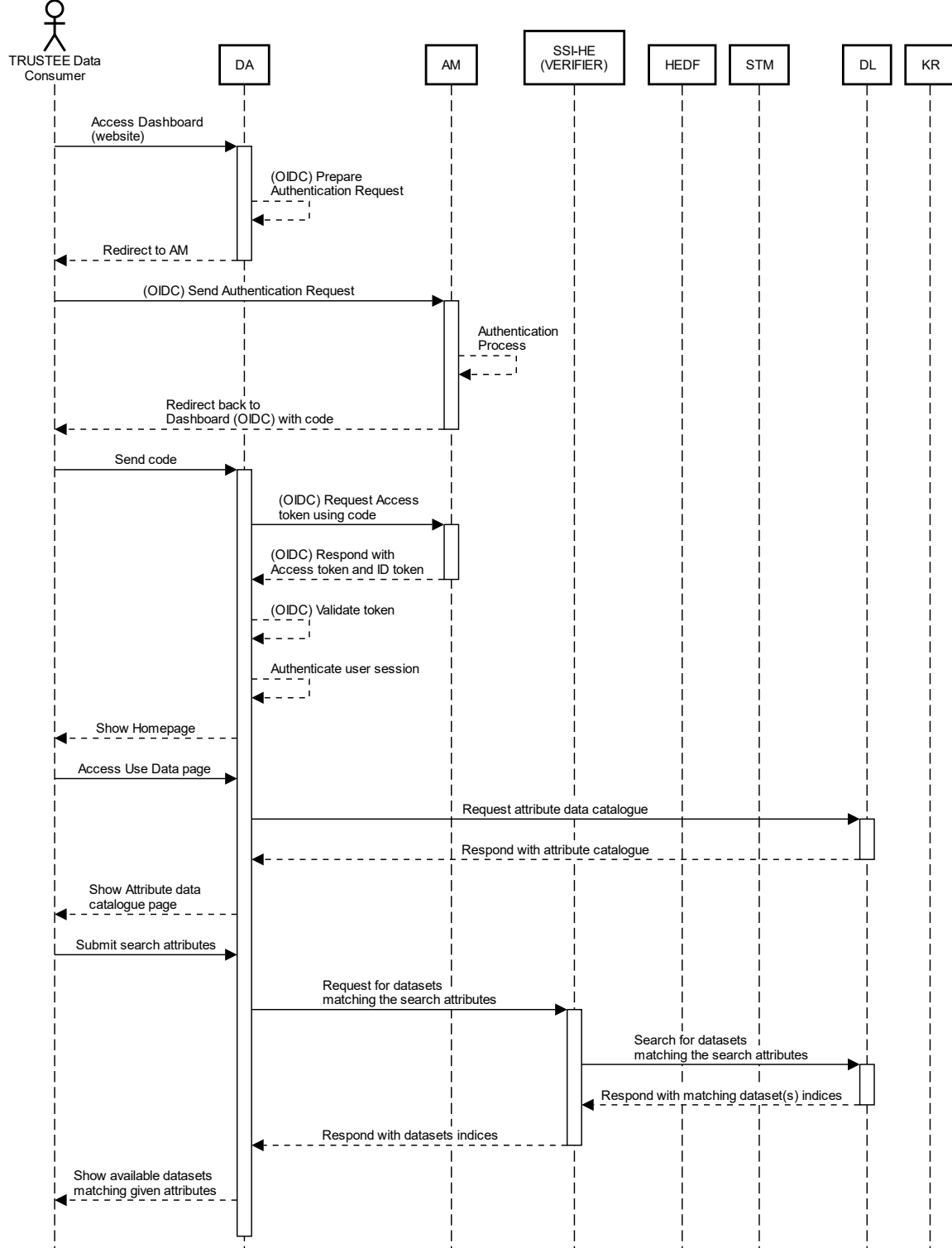


Figure 5: Sequence diagram of TRUSTEE Data Consumer use case flow part 1

TRUSTEE Data Consumer Use Case Sequence Diagram part 2

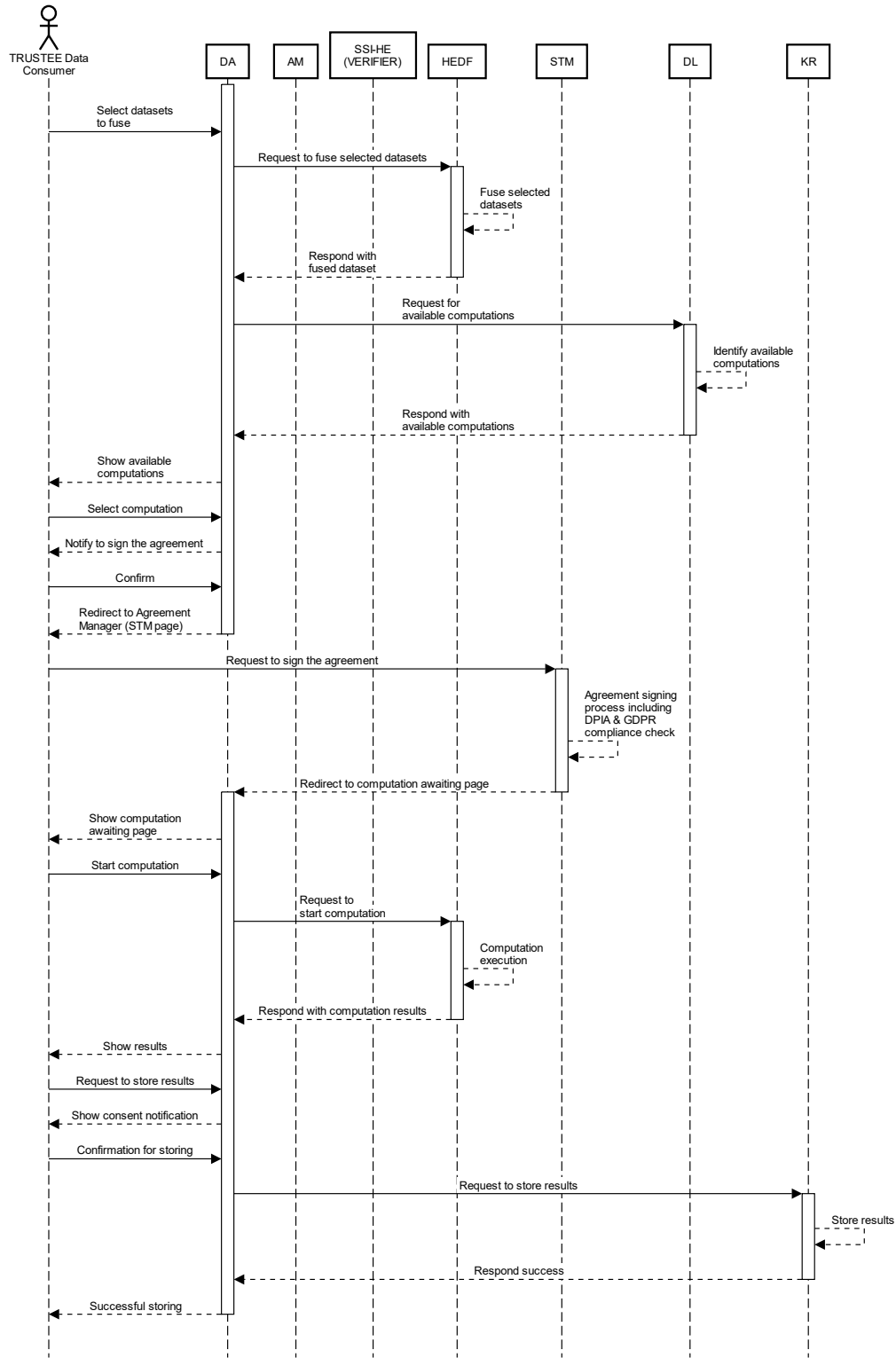


Figure 6: Sequence diagram of TRUSTEE Data Consumer use case flow part 2

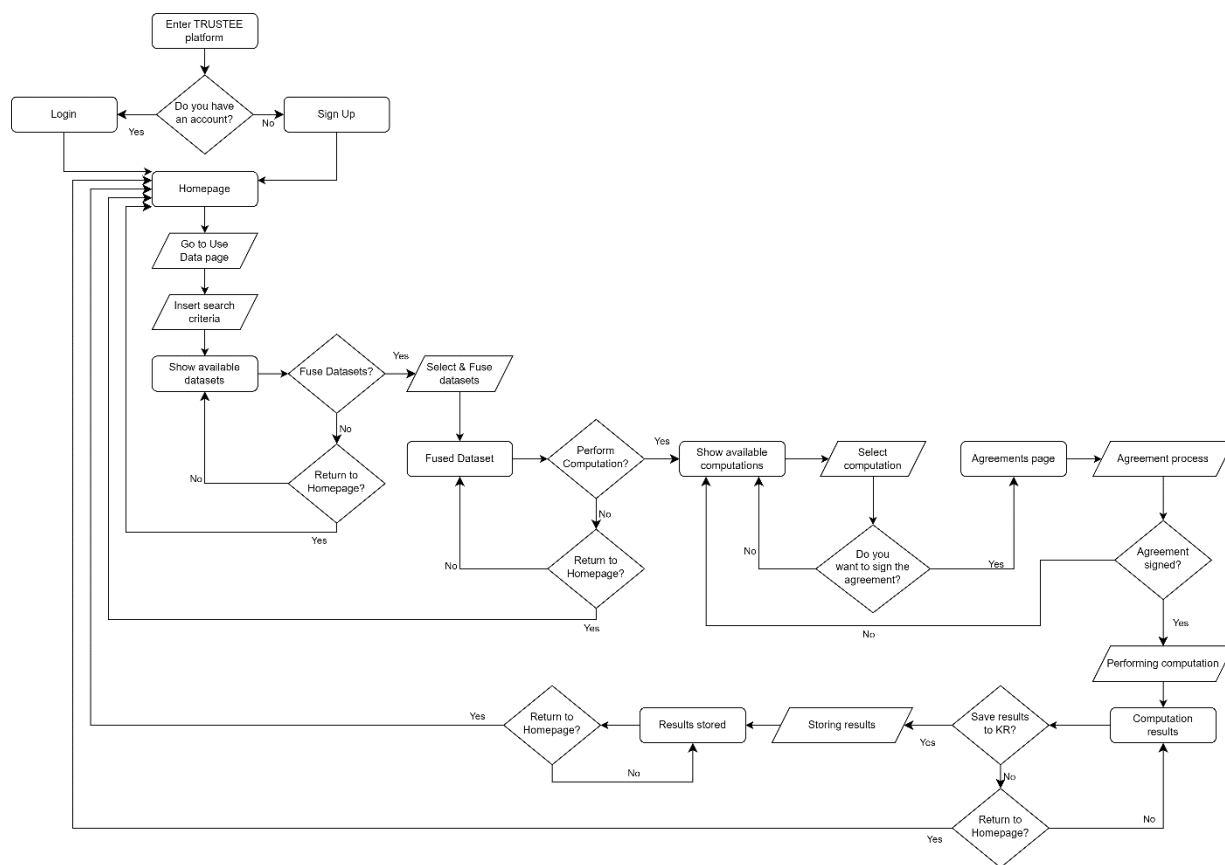


Figure 7: Flow diagram of TRUSTEE Data Consumer use case flow

3.5.2.2. TRUSTEE DATA PROVIDER USE CASE FLOW

The TRUSTEE Data Provider Use Case Flow is described in Figure 8, Figure 9 and Figure 10. As Data Providers enter the TRUSTEE platform, they are prompted to authenticate their identity through the Login or Sign-up page. Upon successful authentication, the user is redirected to the Home Page, where all the functionalities of the platform can be accessed and/or are visualized. Subsequently, the Data Provider proceeds to the Provide Data page where they can initiate the process of registering their dataset as part of the Distributed Data Sources of TRUSTEE. By selecting to provide a dataset, the user is directly redirected to the My Agreements page which is the UI of the STM ABB that handles the process of signing the necessary agreements, performing a DPIA and a GDPR compliance check. Upon successful completion of all these three processes the user is sent back to the DA to complete the final step of the process, which refers to providing additional information about their dataset, such as a dataset name, an endpoint for dataset access, a preferred access timeframe, the dataset's indices as well as the metadata attribute types that characterize the dataset. All latter information are then sent by the DA to the DL ABB.

TRUSTEE Data Consumer Use Case Sequence Diagram part 1

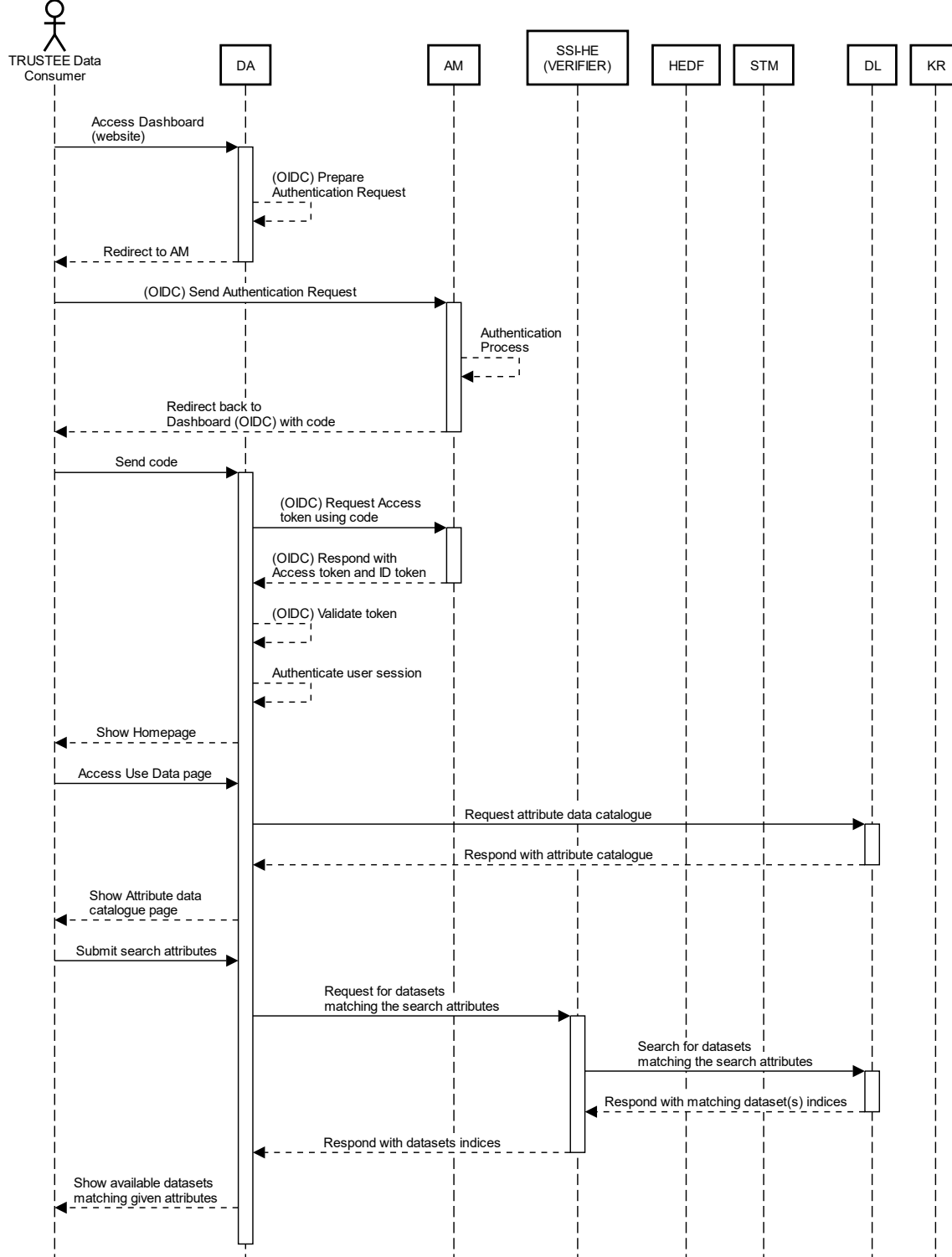


Figure 8: Sequence diagram of TRUSTEE Data Provider use case flow part 1

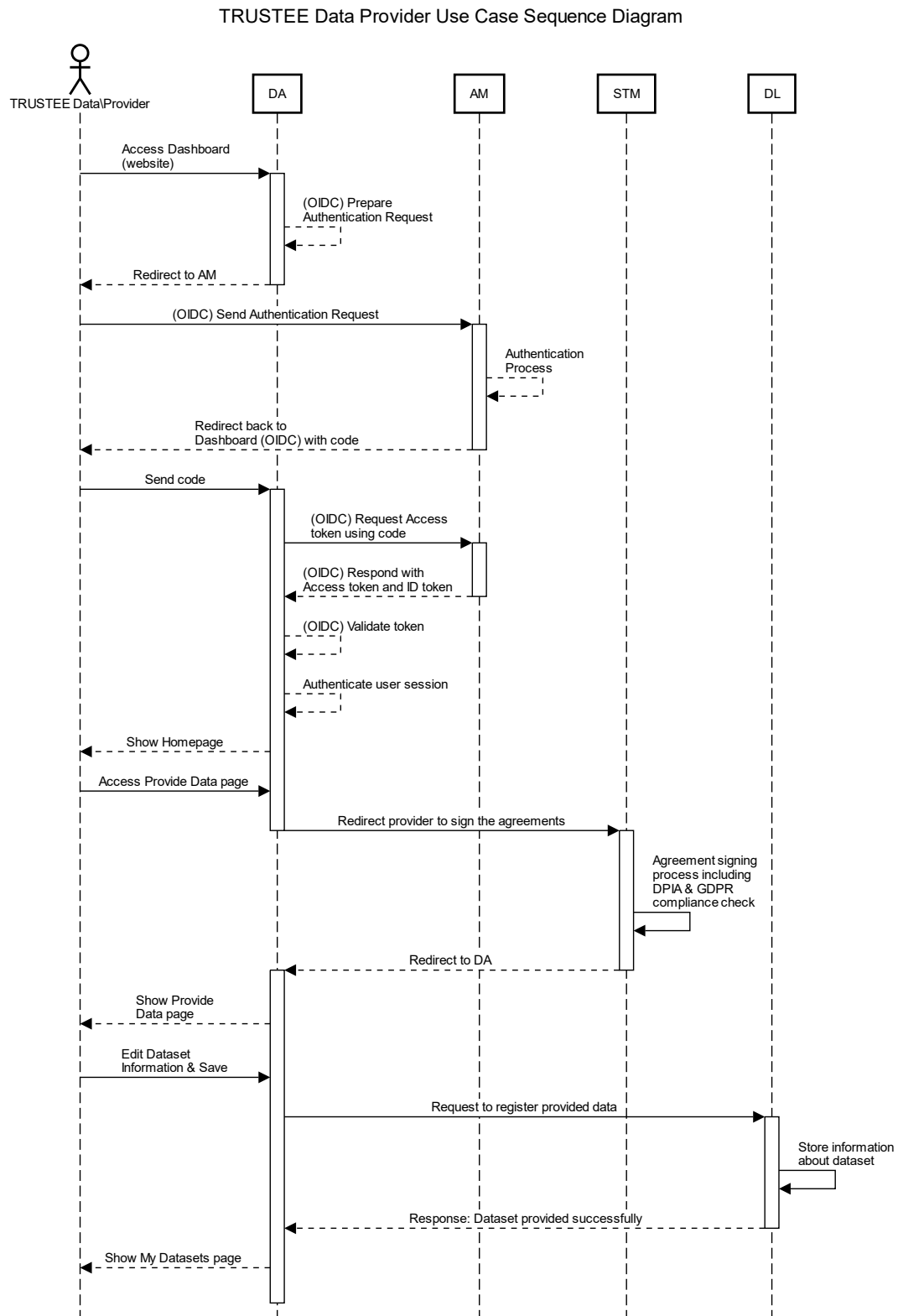


Figure 9: Sequence diagram of TRUSTEE Data Provider use case flow part 2

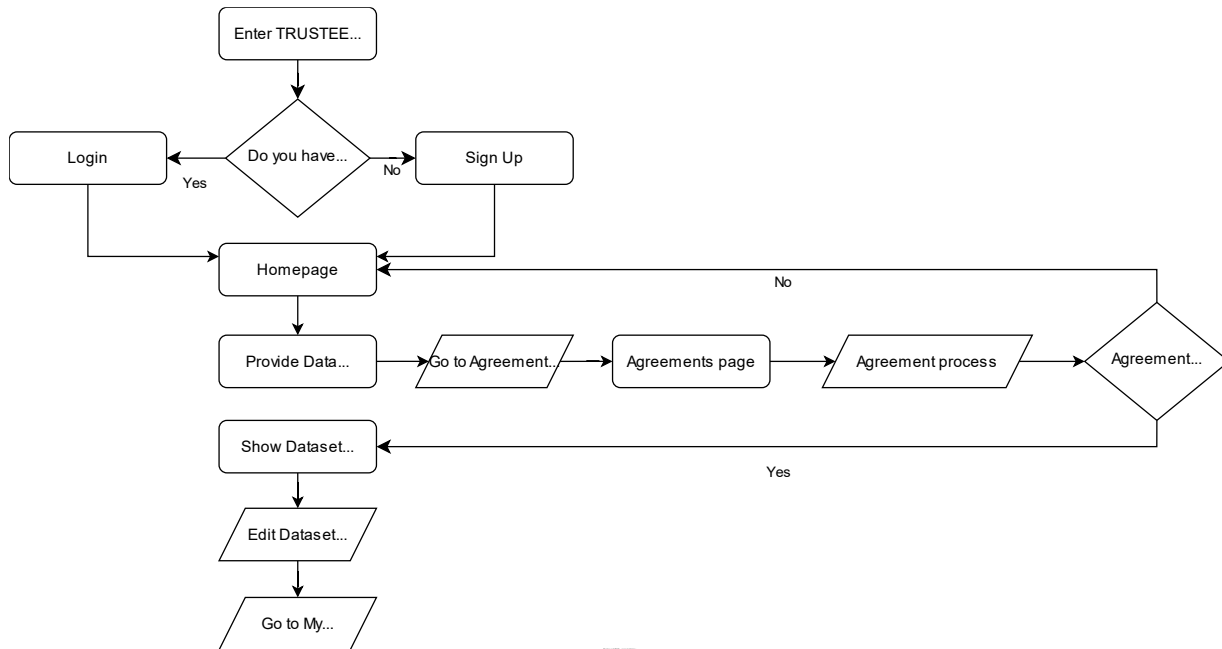


Figure 10: Flow diagram of TRUSTEE Data Provider use case flow

3.5.2.3. TRUSTEE MODEL PROVIDER & CONSUMER USE CASE FLOW

As Model Providers and Model Consumers enter the TRUSTEE platform, they are prompted to authenticate their identity through the Login or Sign-up page. Upon successful authentication, the user is redirected to the Home page, where all the functionalities of the platform can be accessed and/or are visualized. Subsequently, Model Providers and Model Consumers proceed to the AIaaS page through a redirection to the GUI of the AIMaaS ABB where the rest of the process is performed. Since this process is completely handled and visualized by the AIMaaS ABB, its description is provided in the deliverables of WP3, under which the AIMaaS ABB is implemented.

3.5.2.4. VA INTERNAL OPERATION

Using a simple workflow automation use case, Figure 11 shows the operational sequence between the VA and the rest of the DA internal components. Firstly, the user asks the VA to start a specific automation, such as several automated mouse clicks. The VA asks the External interface to the AM of the DA if the user is authenticated and if so, replies to the user that the automation is starting, crafts a specialised access token, and incorporates it inside the automations process file. After that, it sends the message to the Automation Manager who validates it and forwards it to the RPA Robot for execution. At the same time, the Automation Manager reports back to the VA the status of the robot execution.

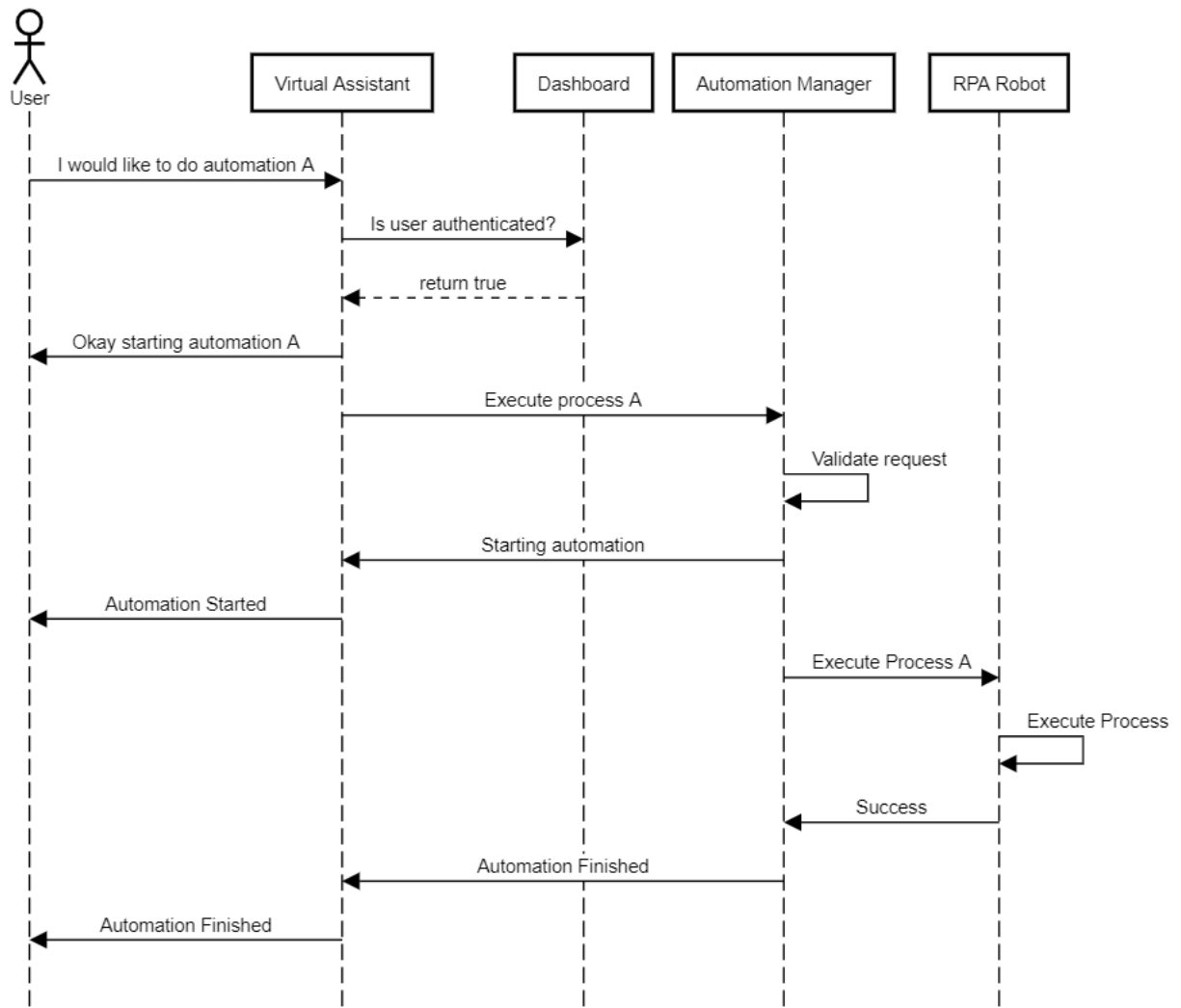


Figure 11: Workflow automation Sequence Diagram

A similar use case describing the interaction of a Data Provider with the VA, requesting assistance to add data to TRUSTEE is shown in Figure 12. First, the VA posts a welcome message. The Data Provider then asks the VA to assist them in registering their data as part of the Distributed Data Sources of TRUSTEE. The VA checks if the user is authenticated by asking the External interface to the AM of the DA and then redirects the user using the appropriate message to the STM user interface, so the Data Provider can move on with the data addition.

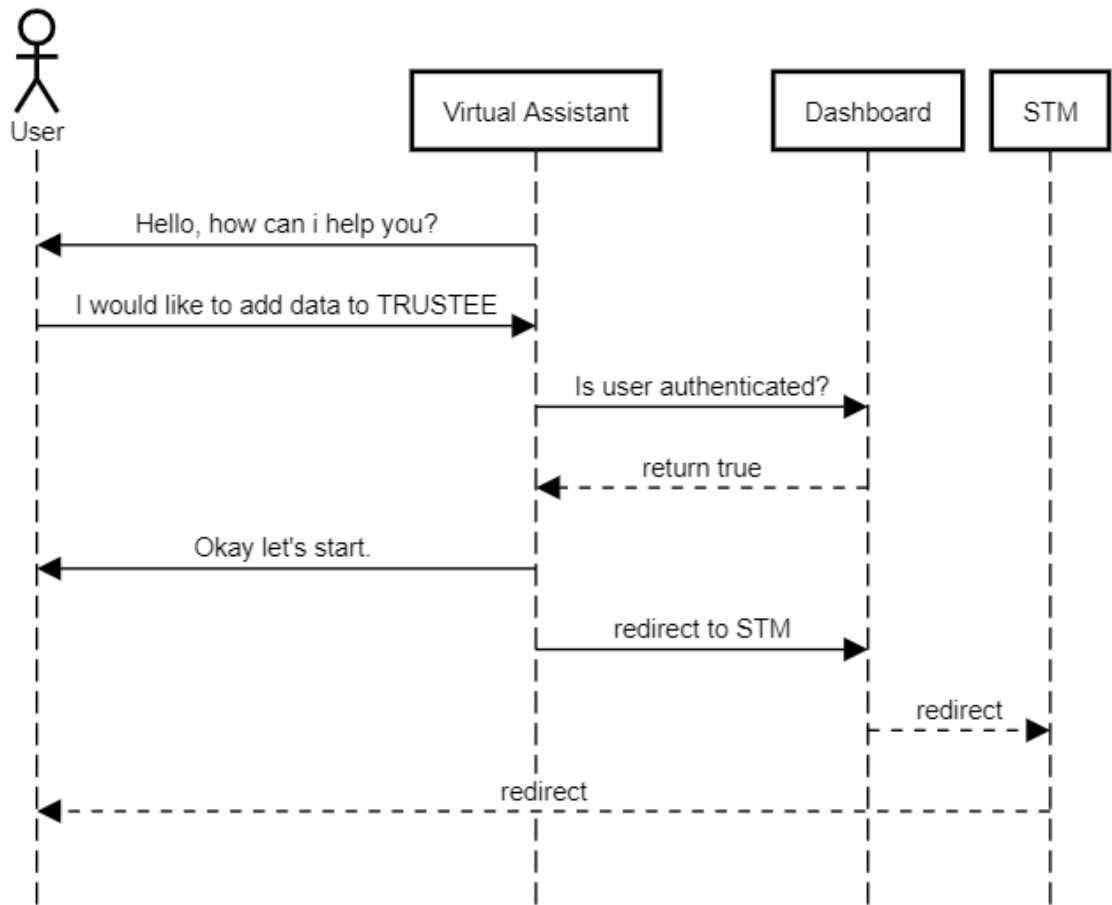


Figure 12: Data provider Virtual Assistant conversation

3.5.2.5. RPA ROBOT OPERATION

As presented in Figure 13, the robot starts by getting a process file and sequentially executes every action that is described in that file. To further elaborate on that, after the robot has interpreted the process object from the file, while there are more actions to execute, gets the next from the stack and tries to execute it. Firstly, the robot searches in the desktop for the associated UI element and, if it finds it, sets the keyboard focus on it and executes the action. If the robot fails to find the UI element, it waits 5 seconds and retries. In the meanwhile, if the user explicitly moves the mouse to the top left corner of the screen, the robot's fail-safe triggers, and the execution gets immediately aborted. This fail-safe is necessary to stop the robot execution in case of an unexpected failure.

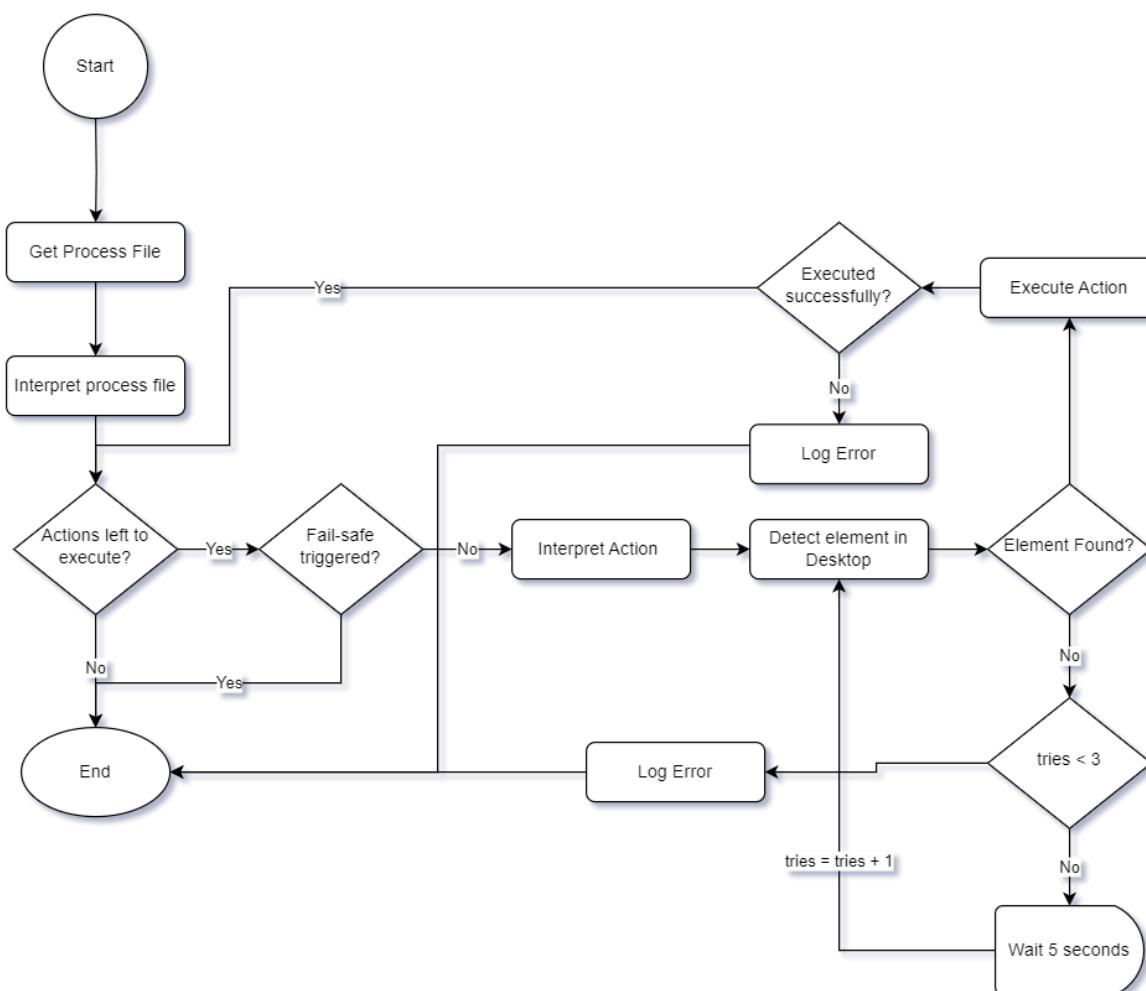


Figure 13: RPA Robot Flow diagram

The Automation Manager is the middleware between the VA and the RPA Robot. In principle, Automation Manager is a service that is always online and waits for a message in an AMQP queue. If such a message arrives, the Automation Manager interprets the message and tries to validate the access token that was provided by the VA. If the token is validated, then the Automation Manager extracts the process file from the message, saves it to memory and starts the robot, indicating it where the process file is. After the successful initialization of the robot, the Automation Manager waits until the robot sends a successful terminating log message, which indicates the end of the process, and the process restarts. This process is shown in Figure 14.

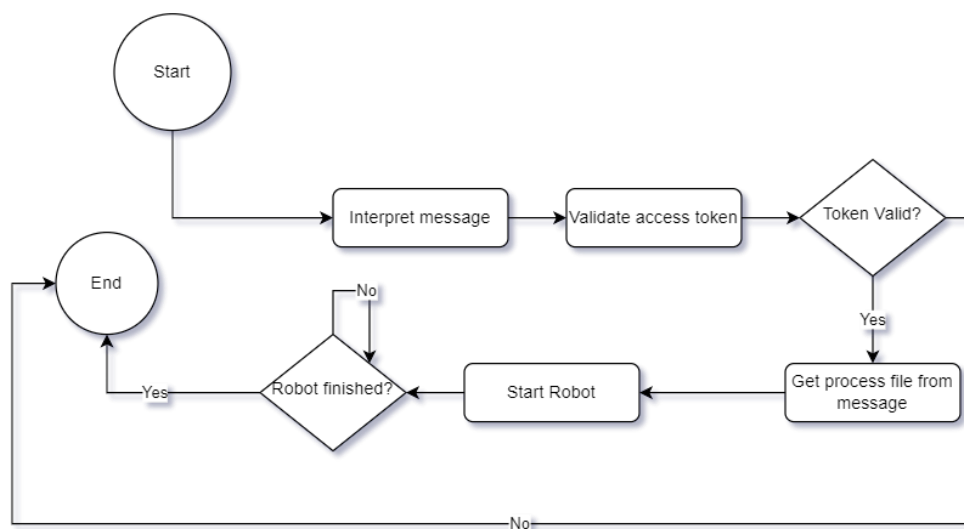


Figure 14: RPA Automation Manager Flow diagram

3.5.3. TECHNICAL DESCRIPTION

In this section, a comprehensive overview of the project's implementation details, covering Input/Output/Data Models, Graphical Interface (including mock-ups and frontend components), and the API will be presented. It highlights the technology stack, programming languages, and development methodologies utilized in the project's construction, along with data flow, user interactions, and data validation mechanisms. Additionally, the section emphasizes the user-friendly GUI design and showcases how the project communicates with external systems through its API.

3.5.3.1. INPUT/ OUTPUT DATA MODELS

This section presents the results of a preliminary investigation regarding the input, output, and the data models used by the DA sub-components. Further details, updates, and enhancements will be presented in upcoming versions of this deliverable, namely D4.2 and D4.3.

As the main entry point to the TRUSTEE ecosystem, the DA uses various data models to communicate with the other ABBs and visualize their content. The investigation of the data models that will be used has been initiated and a preliminary definition of the data model to be used for the communication between the DA and the AM is presented below. Additional data models are to be defined for the communication of the DA with the rest of the TRUSTEE ABBs and will be further described in upcoming versions of this deliverable, namely D4.2 and D4.3.

AUTHENTICATION MANAGER

The communication between DA and AM will be based on the OpenID Connect (OIDC)²² Authorization code flow (mostly based on Uniform Resource Locator (URL) encoded parameters) and is divided into two phases. The first

²² https://openid.net/specs/openid-connect-basic-1_0.html

phase is the registration of the DA as an OIDC Client and consists of two parts, the registration request and the registration response (Table 3).

Table 3: DA-AM OIDC Client Registration Data Model

Request (DA → AM)	Response (DA ← AM)
<pre>{ "id": "oidc_client_registration_request", "application_type": "web", "redirect_uris": ["https://client.example.org/callback", "https://client.example.org/callback2"], "client_name": "TRUSTEE Dashboard" }</pre>	<pre>{ "id": "oidc_client_registration_response", "client_id": "s6BhdRkqt3", "client_secret": "ZJYCqe3GGRvdudKyZ S0XhGv_Z45DuKhCUk0gBR1vZk", "client_secret_expires_at": 1577858400, "registration_access_token": "this.is.access.to ken.value.ff83", "registration_client_uri": "https://authentication- manager.trustee- dev.eu/connect/register ?client_id=s6BhdRkqt3", "token_endpoint_auth_method": "client_secre t_basic", "application_type": "web", "redirect_uris": ["https://client.example.org/callback", "https://client.example.org/callback2"], "client_name": "TRUSTEE Dashboard" }</pre>

The second phase of the DA – AM communication as presented in Table 4 and Table 5 is the actual authentication code flow that authenticates the user and consists of the following four parts:

- The code request
- The code response
- The token request
- The token response

Table 4: DA-AM User Authentication Data Model (Code part)

Code (DA→AM)	Request	Code Response (DA←AM)
{ "id": "code_request", "client id": "1122334455667788", "scope": "openid+profile", "responce_type": "code", "nonce": "abcdef", "redirect_uri": "" }		{ "id": "code_response", "code": "1234567890abcdef1234567890abcdef" }

Table 5: DA-AM User Authentication Data Model (Token part)

Token Request (DA→AM)	Token Response (DA←AM)
{ "id": "token_request", "grant_type": "authorization_code", "code": "1234567890abcdef1234567890abcdef" }	{ "id": "token_response", "access_token": "abcdefabcdefabcdefa bcdefabcdef", "token_type": "Bearer", "id_token": "eyJ0...NiJ9.eyJ1c ...I6ljI.iffX0.De Wt4Qu...ZXso", "expires_in": "3600", "refresh_token": "tGzv3JOkF0XG5Qx2TIKWIA" }

RPA

The RPA component uses two (2) main data models: the message that the VA sends to the Automation Manager; and the Process File. Both data models use the JavaScript Object Notation (JSON) format structure.

Process File

Figure 15 below depicts the process file that contains an array of actions to be executed, an array of connections between actions that describes the flow of execution, a description, a unique UUID, the method that was used to create the process, and the platforms that this process can be executed on. The action data model contains:

- A class name that determines the type of the action
- A description of the action

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- The detection method to use
- The name of the action
- The selector which contains all the necessary information that associates with an element in the screen
- Some action specific data (button, click offset, etc.)

```

        {
          "actions": [{
            "button": 1,
            "className": "RPA::Actions::ClickAction",
            "clickOffset": {
              "x": -1,
              "y": 7
            },
            "description": "Clicks the specified mouse button on
                           the specified control.",
            "id": "aa385cdf-289b-443f-a8c6-435b9ced7c42",
            "method": "IUIA",
            "name": "Click Action",
            "selector": {
              "AutomationID": "num7Button",
              "ClassName": "Button",
              "ClickablePoint": {
                "x": 913,
                "y": 398
              },
              "Culture": 1033,
              "FrameworkID": "XAML",
              "IsClickable": true,
              "IsContentElement": true,
              "IsControlElement": true,
              "IsDialog": false,
              "IsEnabled": true,
              "IsPassword": false,
              "IsPeripheral": false,
              "IsRequiredForForm": false,
              "LocalizedControlType": "button",
              "Name": "Seven",
              "Orientation": 0,
              "ParentAID": "",
              "ParentClass": "ApplicationFrameWindow",
              "ParentName": "Calculator"
            }
          }],
          "connections": [{
            "from": "aa385cdf-289b-443f-a8c6-435b9ced7c42",
            "to": "(null)"
          }],
          "description": "This is a dummy process file.",
          "id": "f1795297-b4af-4f1c-ac35-7c79db0afd39",
          "method": "COMBINED",
          "platforms": [
            "Windows",
            "Chromium"
          ]
        }

```

Figure 15: RPA Process file data model

VIRTUAL ASSISTANT

As presented in Figure 16, the message that the VA sends to the Automation Manager of the RPA module to start an automation contains:

- The issuer who is the component that sent the message
- The access token used for authentication
- The process ID to execute
- The Date of the request
- The scheduled execution which defines when the robot should start the automation

```
{
  "issuer": "TRUSTEE::VA",
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlUaGUgYmVzdCIsIm5hbWUiOiJKb2huIERlZXIiLCJpYXQiOiJlMTYyMzkwMjIsIm1lc3NhZ2UiOiJJZiB5b3Ugc2VlIHRobyYXQsIGNvbmdyYXRzLCB5b3UgYXJlIGEgaGFja2VyIDpPin0.AFsgE2ulWtvK4jb-4iGbVw_NEsFWozWTHnmXgUFLxm0",
  "process": "f1795297-b4af-4f1c-ac35-7c79db0afd39",
  "date": 1687943351,
  "schedule": "immediate",
}
```

Figure 16: VA to Automation Manager message model

3.5.3.2. GRAPHICAL INTERFACE

In order to design and develop the main UI of the DA of the TRUSTEE Platform a research was performed regarding design principles and UI/UX guidelines for interface design. These guidelines are presented below and resulted in developing the initial version of the TRUSTEE UI/UX Guidelines that serve as the groundwork for the development of the main UI and assist in maintaining uniformity and delivering a consistent feel to other ABBs through the TRUSTEE Platform. The UI/UX Guidelines for the design and graphical representation of the TRUSTEE Platform are subject to revisions, updates, and/or enhancements as the project progresses and will also take into account any end-user feedback that may be gathered in the future. These modifications will be made available in future versions of this deliverable, namely D4.2 and D4.3.

In addition to the TRUSTEE UI/UX Guidelines, this section also presents the initial High Fidelity Wireframes of the DA, which will be described in more detail further below.

UI/UX DESIGN GUIDELINES AND RECOMMENDATIONS

TRUSTEE is a platform where data use, re-use, and secondary use will be enabled and protected, also providing ways of data computations while in parallel data are secured. The functionalities of the platform require a pleasant and perceivable environment that will create a pleasant user experience, without causing user's frustration on finding the different sections and capabilities of the TRUSTEE Platform, while also being visually appealing and soothing for the eye.

The proposed guidelines aim to develop a unified Platform outlook and an aesthetically pleasing and soothing environment for the user that might spend quite some time discovering it and using it. The present guidelines intend to set the visual environment that will provide that experience.

European Accessibility Act

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

European Accessibility Act (EAA) sets the requirements for accessible, usable, and user-centred products and services. EAA doesn't provide detailed guidelines, leaving room for innovation, and the respective products and services have to provide information as far as it concerns the accessibility options they provide. Nevertheless, EAA provides a rolling plan for Information and Communication Technologies (ICT) [4] products and services, trying to form the standards and guidelines to establish accessibility for all citizens using ICT. To build these rolling plans and guidelines, EAA takes into account national and international standards like US Access Board, W3C WAI and other big organisations.

Requirements for product services

Annex I- Section 1.2 of Directive (EU) 2019/882 of the European Parliament and of the Council of 17 April 2019 on the accessibility requirements for products and services (Text with EEA relevance) [5] states what defines an accessible and easy to use Platform. In that sense, use cases of a platform should provide a colour alternative and there should not include visuals that may trigger photosensitive seizures. Moreover, alternative features or elements are important for visual elements ensuring the accessibility to visual information for visually impaired users, while there has to be consistent functionality and time for the user to interact with the Platform.

In terms of accessibility of a platform/service, its accessibility functions require appropriate settings of the visual environment luminance and contrast as well as the potential for the user to use assistive programmes and devices. Similarly, in case there are audio signals included, there should be provided the ability for the user to change audio settings (volume and speed) to ensure audio clarity, while in case of manual control features and functionality, sequential control and alternatives of motor control, without the need of simultaneous movements, should be provided.

Recommendations for functionality and accessibility in ICT products and services

In Annex II Section 1.2 of [5] the EAA legislation refers to possible ways of preventing some accessibility and functionality issues. Visual effects that can be cognitively triggering like flickering images or text should be avoided, while elements like buttons should be easily distinguishable as concerning their functionality by appropriate colour use and text labelling. Additionally, in terms of functionality, in case of an error occurrence the user has to be informed of the incident and the reasons it happened with the help of descriptive text and/or image use. In the same path, it is important for an ICT product to provide multiple sensory functionalities (e.g., vocal, and written instructions or tactile signs) in order to assist users with disabilities, as well as the ability to edit certain visual settings (elements/text size and contrast ratio of the platform or images).

TRUSTEE UI/UX GUIDELINES

Reference Information

A digital platform's structure and layout include the visual composition and the functionality. One of the main characteristics of a digital service is its aesthetics and overall appearance. Juliver [6] writes that a good user experience is also related to consistency of a web environment, responsiveness (adjustable to screen size) and accessibility (perceivable content, operable, understandable and robust - usable across different devices and technologies). Additionally, digital services should be conventional to other similar ones presented on the internet, keeping specific formations (like hover state), so that its functionality and layout is familiar to the user. A familiar layout adds to a service's credibility making the user feel safe and trust the platform. The most important guideline is that digital products should be user-centred, since they are created to serve the user and their needs.

Walburg [7] writes that a digital platform's structure requires the header that usually includes navigation option (bar and menu) and/or a minimal search bar, the body where most of the content is displayed and footer that can contain additional important information like contact details or navigation options. An additional structural option is the sidebar

which is an additional navigation tool. Of course, as stated by Hernandez [8], navigation options are crucial to a digital product's structure formation and functionality. As Juliver [6] writes, menu navigation and search bar both serve the purposes of exploration and information detection, while headings serve to visual hierarchy and thus navigation.

Hernandez in [8] proposes a clean and clear layout, familiar and simple. An important role to the visual outcome play colouring and typography. As suggested by Juliver [6], colouring should be restricted to 3-4 prominent colours, colours assist in aesthetics, navigation and even information of the user, thus colour selection should be carefully thought out and the colour palette should be related to the brand. Typography should consist of legible text, avoiding more than three different sizes and should be minimalistic in colour range, insisting on legibility and view of a digital platform, whitespaces are a "must" clearing up the space around text and important information, allowing the user to focus [6]. Whitespace doesn't have to be literally a white space but simply a muted space even if it is about a "muted" background image that is set in high contrast with the elements or text the user should focus on and that is set in low brightness assisting the contrast property.

Marlena Walburg [7] also refers to the fundamental elements of a digital service among which are buttons, landing page that is the first page that a user sees, hover state, links, drop-down menu which, as Walburg writes, reveal a platform's structure while also assist to a cleaner layout outcome and lastly responsive design. To these Juliver [6] adds Call-to-Action (CTAs), elements informing the user and promoting the user to engage in a specific action and whitespace that assists the purpose of CTAs and also declutters the amount of information provided in the platform, giving space and time to the user to digest the information.

End-user Requirements

Based on the UX research conducted under T2.3, which was presented, described, and analysed in the first version of D2.1 [1], several user-centred requirements were extracted that refer to the experience of the end-user of the TRUSTEE Platform. These User-centred requirements have been translated into technical functional and non-functional requirements, as presented above in Table 1 and Table 2. The non-functional requirements pertaining to the usability of the TRUSTEE platform and the DA are presented in Table 6, below:

Table 6: DA Usability Requirements

ID	Requirement
USR-047, USR-059	The TRUSTEE Dashboard SHOULD keep colours consistent across the UI
USR-058, USR-059	The TRUSTEE Dashboard SHOULD make sure that all essential Information is displayed in a visual hierarchy on one screen
USR-047, USR-062	The TRUSTEE Dashboard COULD provide items such as tables and/or charts in a minimised mode with the option to display further information in a modal window or navigate to a page with additional information.
USR-059 USR-062	The TRUSTEE Dashboard SHOULD display information clearly and efficiently, in a user-friendly way

USR-054 USR-058	The TRUSTEE Dashboard SHOULD make sure that all essential information is immediately accessible
USR-047, USR-049	The TRUSTEE Dashboard SHOULD provide a consistent and complete UI serving the goals of the end-user
USR-059	The TRUSTEE Dashboard COULD make sure that the purpose of each feature of the UI is clear to the end-user

A well-structured digital service provides a better user experience, which is important for the user as well as the digital product impact and future development. The following sections outline in detail the proposed UI/UX guidelines for the TRUSTEE Platform, based on the relevant literature discussed above, the non-functional usability requirements presented in Table X, W3C “Standards/Guidelines” [9] and W3C “Design and Develop” [10] sections, as well as Material design by Google²³ and Adobe Spectrum²⁴. TRUSTEE’s UI/UX guidelines are also based on articles and scientific literature. The guidelines are distributed in three categories, namely: Consistency, Functionality, and Accessibility.

Consistency

Aesthetics play an important role on user experience and platform’s effect on the user. A smooth navigation through the different parts of the platform as well as a pleasant digital environment and aesthetically enjoyable are elements that ensure consistency on a Platform. Consistency is important for the user, so that they stay focused, navigate easily through the digital platform and avoid feelings of frustration over overlapping visual information, e.g., one page of the platform having a vertical navigation bar and the other a horizontal. Consistency is related to visual aspects, but also placement and elements usage. The following subsections outline the TRUSTEE’s guidelines over consistency of the Platform. This can be achieved through a uniform visual layout where the user can perceive that every section is about the same topic and doesn’t have to conform with different layouts in terms of visual outcome, positioning and look of elements, text and more. The following paragraphs aim to provide specific guidelines that should be followed for a better outcome.

Visuals

Visuals include colour palette, fonts, and sizes. Christine Vallaure in a recent article [11] presents the colour use and options in UI design. There are four primary ways to note colour in UI; Pantone, CMYK, HEX, and RGB. HEX and RGB are the ones used mostly in UX design and especially HEX because of its efficient way of presentation, since it provides the exact same result as RGB, while the other two are used in printing. What is usually suggested is that a digital platform should have two to three colours for a better design and result, exploiting their variants to the fullest. After choosing the desirable colours, it is useful to name them according to their use as neutrals, primary and secondary

²³ <https://m2.material.io/>

²⁴ <https://spectrum.adobe.com/>

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

colours and to define their on-colours, namely the colours selected for typography or icons display on the relative colour background by holding on the accessibility regulations from WCAG [12].

Asha Rajput [13] refers to accessibility as one of the most important factors in web design and colouring is a key component to it. Colour is related to vision impairments and limitations, so colouring a website has to be a well-thought process. Colour selection also affects and impacts the user in different ways; it represents the brand's identity of the website and can provoke different emotions and imply the intentions of the page. Colours and colour combinations can be associated with a brand or a story, and thus every time one sees these combinations, they will be reminded of something specific. Also, colours can indicate hierarchy in a text or visual space by highlighting or emphasising with vibrant colours the text or the block that should stand out.

TRUSTEE's colour palette consists of the following colours in HEX format presented in the TRUSTEE D6.3 – Dissemination and Communication Strategy and Plan [14] (see Table 7);

Table 7: TRUSTEE's colour palette

Name	HEX Code	Colour
Blue	#1F4693	
Yellow	#FFD457	
Orange	#FF9E16	
Green	#718F40	
Light Blue	#77C5EA	

The intention is to provide a platform that will indicate TRUSTEE's colour combinations but will also be functional for the user. Thus, for the TRUSTEE Platform the following colours apply, according to the original ones (see Table 8):

Table 8: TRUSTEE's colour palette for the TRUSTEE Platform

Category	Colour Name	HEX Code	Colour
Neutrals	Black	#000000	
Neutrals	White	#FFFFFF	
Primary Text	Blue	#1F4693	

Secondary Text – 1	White	#FFFFFF	
Secondary Text – 2	Black	#000000	
Primary Colour	Blue	#1F4693	
Background Colour	White	#FFFFFF	
Secondary Colour – 1	Light Blue	#97D2EE Opacity 50%	
Secondary Colour - 2	Yellow	#FFD457	
Secondary Colour -3	Light Yellow	#FFF8E2	
Borders/Lines	Grey 1	# DEDEDE	
Borders/Lines – 1	Grey 2	#DEDEDE66 or #DEDEDE with 40% opacity	

Colour selection of a variant of the original colouring is because of practical and usability reasons within the TRUSTEE Platform.

Body

The guidelines as presented in Table 9 for the structure of body were based on literature and articles regarding research and experience for the design and structure of a web page or platform [15], [16], [17], [18]

Table 9: TRUSTEE Body Background and Text Guidelines

Background colour	White	#FFFFFF	
Font size [19]	14px		
Primary Text Colour	Black	#000000	

Font size for main text	14px
Font size for minor notes or footnotes	12px
Font family	Arial (sans-serif)
Line length	40–70 characters per line [20]

Similarly, text guidelines are based both on the experience and expertise of UX designers as delivered from related articles and on the publications related to accessibility and reading abilities of different people [16] [18] [21] [22] [23] [24]. For the font size of headings of Table 10 specifically, UX guidelines were formed and adjusted to normal text's font size that was set to 14px.

Table 10: TRUSTEE Headings - Guidelines

Headings		
Colour	Blue	#1F4693
Font-weight		Bold
Heading 1 – font size		24px
Heading 2 – font size		21px
Heading 3 – font size		18px
Heading 4 – font size		16px
Heading 5 – font size		14px

The suggested headings above are based on literature recommendations as well as heading and body text relations among different platforms or guideline material. It is advised for designers to use two to three headings styles.

Elements

Elements denote the different functional or informational elements/objects on the platform such as headers, footers, buttons²⁵, and navigation bar. The consistency of main elements in a platform can offer to uniform outlook and a smooth and easy navigation process. The present paragraph contains the must-follow UX guidelines as well as suggestions for non-mandatory components or parts of those components.

TRUSTEE Header

The header is the upper part of a web platform containing the logo and name of the platform and probably icons and elements like the user profile or accessibility points. For TRUSTEE header the mandatory guidelines are included in the Table 11 as well as an example of its outlook in Figure 17.

Table 11: TRUSTEE Header Guidelines

Border Line	Light grey	#DEDEDE	
Background Colour	White	#FFFFFF	
Icons Colour	Blue	#1F4693	
Icon Hover Background Colour	Light grey	#DEDEDE	
Icons Hover Colour	Blue	#1F4693	
Selected Background Colour	Blue	#1F4693	

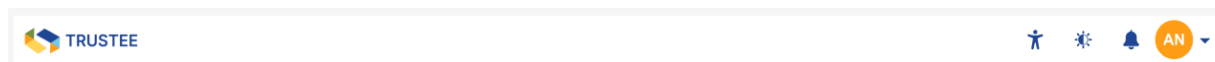


Figure 17: TRUSTEE Header

User Profile

The right side of the header can contain the User Profile element. This may include user's account page, settings and logout options and consists of a circle include the two first letters of the user's name and a drop-down arrow, which

²⁵ <https://m2.material.io/components/buttons#usage>

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

when clicked reveals the user profile levels. Table 12 and Table 13 presents the TRUSTEE user profile guidelines and icons while Figure 18 shows the actual element and its dropdown menu.

Table 12: TRUSTEE User Profile Guidelines

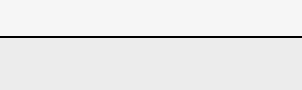
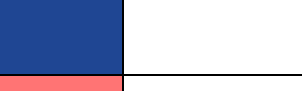
User Profile Guidelines			
Circle Colour	Orange	#FF9E16	
Circle Text Colour	White	#FFFFFF	
Circle Text Font Size	24px		
Circle Text Font - Weight	700 or Bold		
Down Arrow	Blue	#1F4693	
User Profile Dropdown Menu			
Background Colour	White	#FFFFFF	
Background Clicked Colour	Grey 6	#F6F6F6	
Border Live / Divider	Grey 2	#DEDEDE66 or #DEDEDE with 40% opacity	
Main Text colour	Blue	#1F4693	
Logout Text Colour	Red	#FF7474	

Table 13: TRUSTEE Icons for Dropdown Menu

Icons and Labels		
Label	Main Icon	Clicked Icon

My Account		
Settings		
Logout		

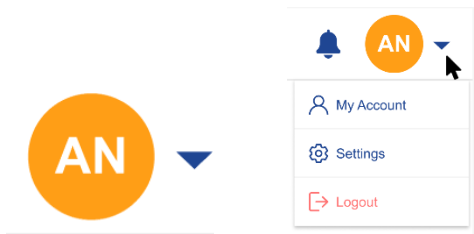


Figure 18: TRUSTEE User Profile Element and the Related Dropdown Menu

Header: Suggestions

The Table 14 below shows the icons used in the Header on the left of user profile element in order to inform the user about the functionalities of the platform, accessibility, dark mode and notifications section.

Table 14: TRUSTEE Header Suggested Icons

Icons	Colours			States	
	Main		Hover (Background)	Main	Selected
Accessibility	Blue	#1F4693	Light Grey #DEDEDE		
Light/Dark Mode	Blue	#1F4693	Light Grey #DEDEDE		
Notifications	Blue	#1F4693	Light Grey #DEDEDE		

Breadcrumbs

Breadcrumbs is called the text in the form of a simple string on top of the main page, indicating the user their place in the dashboard platform. Table 15 and Figure 19 show the guidelines for breadcrumbs and instance of dashboard depicting breadcrumbs.

Table 15: TRUSTEE Breadcrumbs Guidelines

Placements	Top left of main dashboard page and on the right of the navigation sidebar		
Background Colour	Transparent		
Text Colour	Blue	#1F4693	
Text Font Size	16px		
Text Font Weight	Normal		
Path	Use slash (/) to distinct the different pages starting with TRUSTEE e.g. TRUSTEE / Home Page		

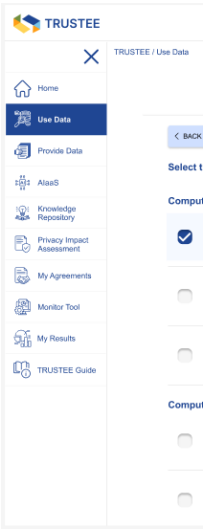


Figure 19: TRUSTEE Breadcrumbs placed on the top left beside the navigation sidebar

TRUSTEE Footer

The footer is the part at the end of a web platform containing the logo, informational tags and labels that could navigate the user to other parts of the platform and social media accounts. For TRUSTEE footer the mandatory guidelines are included in Table 16 below while Figure 20: TRUSTEE Footer presents its outlook.

Table 16: TRUSTEE Footer Guidelines

Border Line	Light grey	#DEDEDE	
Colour	White	#FFFFFF	
Text/Icons Colour	Blue	#1F4693	
Hover Colour	Blue	#1F4693	
Text/Icons Hover Colour	White	#FFFFFF	
Text Font Size for Labels	16px		
Text Font Size for EU message	14px		

Text Font Weight	Regular
------------------	---------

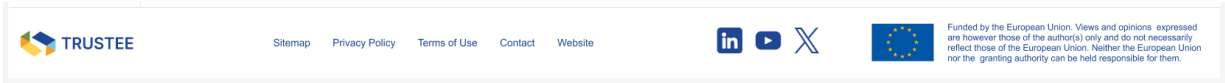


Figure 20: TRUSTEE Footer

All the Icons as well as labels and text referred must be included in the TRUSTEE footer, as shown in Table 17 and Table 18.

Table 17: TRUSTEE Footer Icons and EU Flag

Icons	Colours						States	
	Main			Hover (Background)			Main	Hover
LinkedIn	Blue	#1F4693		Light Grey	#DEDEDE			
YouTube	Blue	#1F4693		Light Grey	#DEDEDE			
Twitter	Blue	#1F4693		Light Grey	#DEDEDE			
EU	-			-				-

Table 18: TRUSTEE Footer Labels and EU Text

Footer Labels and EU Message
Sitemap
Privacy Policy

Terms of Use
Contact
Website
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Navigation Bar

Navigation in a Platform is the ability to move to different sections of the page, visit specific pages, and be able to go through the different information sections included. The most common navigation elements are navigation bar and menu [16], [18], [21], [25], [26] structured in a vertical position on the left side of the platform as a navigation drawer opening and closing in relation to user's wishes. The presentation and construction guidelines are given by Figure 21 and Table 19 relatively. Crucial to the navigation process is the semantics in the platform in general and in the navigation bar particularly, namely the meanings used to name a process, section or domain [27]. The main points of focus for the semantics would be the following:

1. logical relations of words to commands/function of the element and the section it is linked with
2. consistent wording (all words should be verbs or nouns)

Table 19: TRUSTEE Navigation Sidebar Guidelines

Navigation Bar			
Bar Placement	Left Side Bar		
Border Line	Light grey	#DEDEDE	
Divider Line	Light grey	#DEDEDE66 (#DEDEDE 40% opacity) or #E8E8E8 or RGBA(222,222,222,0.4)	
Bar Colour	White	#FFFFFF	
Text Font size	14px		

Text/Icon Colour	Blue	#1F4693	
Hover			
Background Colour	Light grey	#DEDEDE	
Text/Icon Colour	Blue	#1F4693	
Selected			
Background Colour	Blue	#1F4693	
Text/Icon Colour	White	#FFFFFF	

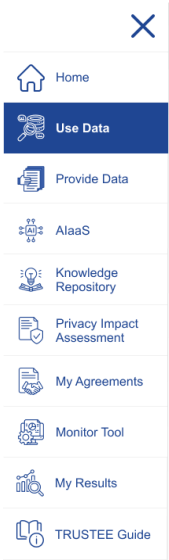


Figure 21: TRUSTEE Navigation Sidebar

Following, Table 20 contains all icons that are used in the navigation bar for each label which might be further updated in in the future.

Table 20: TRUSTEE Navigation Sidebar Icons

Label	Icon	
	Main	Selected

Home		
Use Data		
Provide Data		
AIaaS		
Knowledge Repository		
Privacy Impact Assessment		
My Agreements		
Monitor Tool		
My Results		
TRUSTEE Guide		
Burger Menu		
Close Navigation Bar		

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Navigation Bar: Suggestions

In some cases, the labels included in a navigation bar may not be the only level of navigation in a platform. There can be cases where a platform level contains sublevels, structuring a hierarchy formation and distinguishing the different parts of the specific page. In that case and in need of including sublevels to a label UX guidelines suggestions are as in Table 21 below.

Table 21: TRUSTEE Suggestions on Menu Tabs

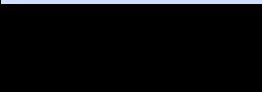
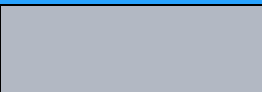
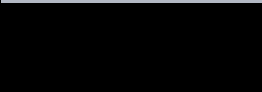
Menu Tabs			
Placement	Open on the right of the label		
Border Line	Light grey	#DEDEDE	
Divider Line	Light grey	#DEDEDE66 (#DEDEDE 40% opacity) or #E8E8E8 or RGBA(222,222,222,0.4)	
Colour	Off white grey	#FDFDFD	
Text Colour	Black	#000000	
Hover Colour	White	#FFFFFF	
Text Hover Colour	Dark Grey	#404040	
Text Font Size	14px		
Menu levels should be limited for better efficiency, namely not more than three levels of nested menus			

TRUSTEE Buttons

Table 22 below presents the four types of buttons found on the TRUSTEE platform as shown in Figure 22, Figure 23, Figure 24 and Figure 25. The desirable On Click action of the button is to include a ripple effect.

Table 22: TRUSTEE Main Button Guidelines



Active/Hover state			
Type	Contained Button		
Colour	Light Blue	#CFDFFF	
Text Colour	Black	#000000	
Font size	14px – Upper Case		
Text weight	Regular		
Font family	Sans-serif		
Size	dynamically fit to text label size		
Box shadow	0px 2px 4px 0px		
Box shadow Colour	Light Grey	RGBA (0, 0, 0, 0.25)	
In case there needs to be a submit and cancel button for a specific task place submit on the left and cancel on the right side [19].			
Disabled button (Desirable)			
Colour	Grey	#B2B8C3	
Text Colour	Black	#000000	
Font size	14px – Upper Case		
Text-weight	Regular		

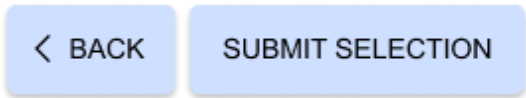


Figure 22: TRUSTEE Main - Buttons



Figure 23: TRUSTEE Disabled Buttons

Buttons indicating next step of the workflow for the user, have to be different than the main button and obvious to the user as for their use as for their crucial role in the platform and their actions. Such buttons of positive workflow message are instructed in Table 23 and an instance of a button in that category is shown in Figure 24.

Table 23: Green Button Guidelines - Positive Messaging

Addition/Move forward Button			
Colour	Green	#79DE69	
Text Colour	Black	#000000	
Font size	14px – Upper Case		
Text-weight	Regular		



Figure 24: TRUSTEE Add - Button.

Buttons denoting a very definitive move that will change dramatically the flow of user’s actions such as deletion, removal and so on, shall be distinctive and thus red is the recommended choice in such cases. Instructions to construct such buttons are presented in Table 24 and instances are shown in Figure 25.

Table 24: Red Button Guidelines - Cancellation

Delete/Cancel Button			
Colour	Red	#FF7474	

Text Colour	Black	#000000	
Font size	14px		
Text-weight	Regular		



Figure 25: TRUSTEE Delete, Remove and Cancel buttons

Forms

Forms components in TRUSTEE should have a consistent layout and colour palette, to assist the user into the component’s intentions. All form components should have an outline. (Table 25). An example is presented in Figure 26. On the other hand, disabled form components should be easily identified as not being able to be edited as showcased in Figure 27.

Table 25: TRUSTEE Select and Input (Forms)

Select and Input (Forms)			
Type	Select		
Background Colour	White	#FFFFFF	
Label Colour	Blue	#1F4693	
Text Colour	Black	#000000	
Font size	14px – Upper Case		
Text weight	Regular		
Font family	Sans-serif		

Outline (stroke)	Blue	#1F4693	
Disabled select or input (Desirable)			
Colour	Grey	# FCFCFC	
Text Colour	Grey	#9E9E9E	
Outline (Stroke) Colour	Grey	#9E9E9E	
Font size	14px		
Text-weight	Regular		

* Dataset Name

* Domain of Dataset

* Dataset Size

* No. of Records

Figure 26: Select and Input (Forms)

ID

Figure 27: Disabled Form component

Table of DA Colours

Table 26 displays all TRUSTEE colours of dashboard.

Table 26: All DA Colours

Category	Colour Name	HEX Code	Colour
Neutrals	Black	#000000	
Neutrals	White	#FFFFFF	

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Primary Text	Blue	#1F4693	
Secondary Text - 1	White	#FFFFFF	
Secondary Text - 2	Black	#000000	
Primary Colour	Blue	#1F4693	
Background Colour	White	#FFFFFF	
Secondary Colour -1	Light Blue -1	#97D2EE Opacity 50%	
Secondary Colour -2	Yellow	#FFD457	
Secondary Colour -3	Light Yellow	#FFF8E2	
Borders/Lines	Grey -1	#DEDEDE	
Borders/Lines – 1	Grey -2	#DEDEDE66 or #DEDEDE with 40% opacity	
Additional			
Disabled	Grey -3	#9E9E9E	
Disabled Button	Grey -4	#B2B8C3	

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Submit/Accept	Green	#79DE69	
Cancel/Delete	Red	#FF7474	
Table Row	Blue Grey -1	#D4DAE6	
TRUSTEE Button	Light Blue -2	#CFDFFF	
TRUSTEE Input	Grey -5	#F5F5F5	
Table Row	Blue Grey -2	#EDF0F6	
Checkbox Row	Light Blue -3	#CFDFFF - Opacity 20%	
OnHover Row	Grey -7	#E5E5E5 Opacity 30%	
Disabled Input	Grey -8	#FCFCFC	
Table Row dark blue	Blue Grey -3	#D4DAE6	
Other Colours			
TRUSTEE Orange	#FF9E16		
Checkbox Disabled	#9E9E9E		

Functionality

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

There are specific functionality traits that have to be used across the platform and critically affect the user and their experience. In this category error messages are included as well as language formation.

Informative / Indicative Messages for the User

A good User Experience includes the best possible communication between the user and the platform. This translates to a visual as a textual or audio types of communication between the two parts so that the user can be informed for the different work stages and functions of the platform, as the platform can receive messages from the user in the form of buttons or text more often.

These messages can be informative such as reporting that the performed action is successfully completed or that the form is correctly filled in or that the actions performed are heading to a specific action in the platform. Usually, messages as such may be of a green colour, giving the “green light” as said, for the user to continue with their actions in the platform. The message may contain an indicative icon and a functional form of text allowing users to close the message and move on with their navigation performance. Figure 28 shows the proposed informative TRUSTEE message in green colour tone #79DE69.

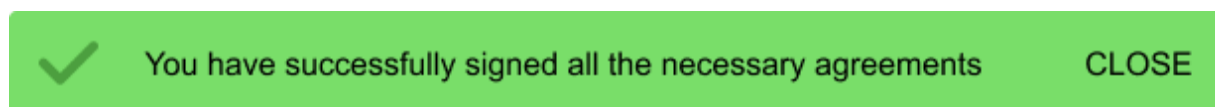


Figure 28: TRUSTEE indicative information message

Other form of messages are the warning indications. The platform shall inform the user that their actions may conclude to a functional dead end or cause unnecessary complications to platforms performance. The instructions are similar to those of the informative green message though the change of colour is demanded to indicate the different purpose of the message as to catch user's eye and push them to pay attention. As in nature such messages are colour yellow. Figure 29 shows the proposed warning TRUSTEE message in yellow colour tone #FFD457.

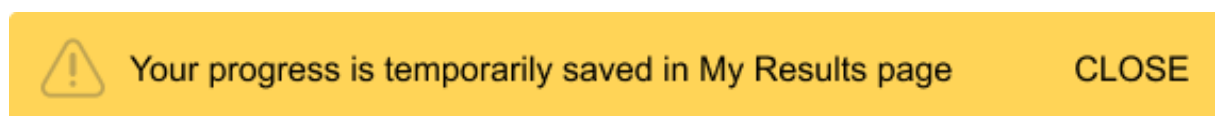


Figure 29: TRUSTEE indicative warning message

Errors

Errors²⁶ constitute a very important part of a web page or platform, since they report a functional problem occurring through use of the platform or block the workflow of the user indicating a flawed action. It is very important that the error message is as informative as indicative for the required reformative actions the user has to make to resolve the issue. The two main points that a UI developer and designer shall follow are:

1. Single word or non-descriptive alerts are not very efficient or informative. Error messages should be descriptive and provide information for the user to understand and solve the problem.

²⁶ W. W. A. Initiative (WAI), “Designing for Web Accessibility – Tips for Getting Started,” *Web Accessibility Initiative (WAI)*, Mar. 28, 2023. <https://www.w3.org/WAI/tips/designing/> (accessed Mar. 30, 2023).

2. Colour: Red - # FF7474

In Figure 30 there is presented the error message of TRUSTEE including the “TRY AGAIN” choice – letting the user restructure or rewrite their action – and the CLOSE choice.

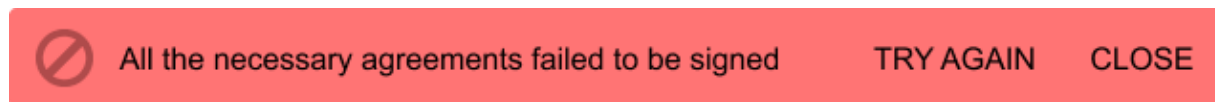


Figure 30: TRUSTEE indicative error message

Writing

Writing is the main form of communication between the user and the platform interface. Thus, the text of a platform has to be legible, understandable, specific, descriptive and clear as to its information and purpose. The main points found in literature [22] would be summarised to the following:

1. Language used on TRUSTEE UI should be simple and easy to understand, while simultaneously descriptive.
2. Use headings and font-styles to set hierarchy on the screen, providing a more efficient and smooth navigation experience for the user.
3. Divide text in paragraphs for better readability with a break paragraph line that should be.
4. Take advantage of the many heading options to create a visual text hierarchy in text but there should not be more than three font sizes.
5. Single line on the next page (orphans) or windows (single word line) should be avoided.

Accessibility

Accessibility is one of the most important characteristics of a Platform since it enables users with impairments or difficulties to access the platform and be equally able to use it. These guidelines consider visual accessibility rules as far as it concerns the contrast ratio between text and background for the best possible quality according to the W3C guidelines and “Luminosity Colour Contrast Ratio Analyser” tool [28].

MAIN DASHBOARD UI

The development of the main UI of the TRUSTEE Platform follows an iterative process that begins with the creation of low fidelity wireframes and continues with the creation of high fidelity wireframes; the latter serve as the basis, along with the UI/UX Guidelines previously mentioned, for the creation of the interactive UI Mock-ups that will be used as the basis for the frontend development of the DA [29]. Wireframes are the first step of a product's appearance and functioning. TRUSTEE wireframes are intended to illustrate the platform's overview, how it will be built – an initial sketch of the design, how the various components will be arranged or displayed, as well as their connections [30].

Interactive UI Mock-ups

TRUSTEE UI Mock-ups include figures that depict the appearance of the DA's Log-in, Sign-up, Home page, Use data page, Provide data page, Knowledge Repository page, My results page, and My datasets page, aiming to set the goal

and feel of the desired outcome. The UI mock-ups were created with the Figma software, and they are presented and described in more detail in the following sub-sections.

Login Page

The first page to be presented is the login page. A user will be able to use the platform only after being authenticated, therefore, the user has to use the login page in order to be authenticated through the TRUSTEE authentication and authorization mechanism. Figure 31 below depicts how the platform's connection will supposedly appear.

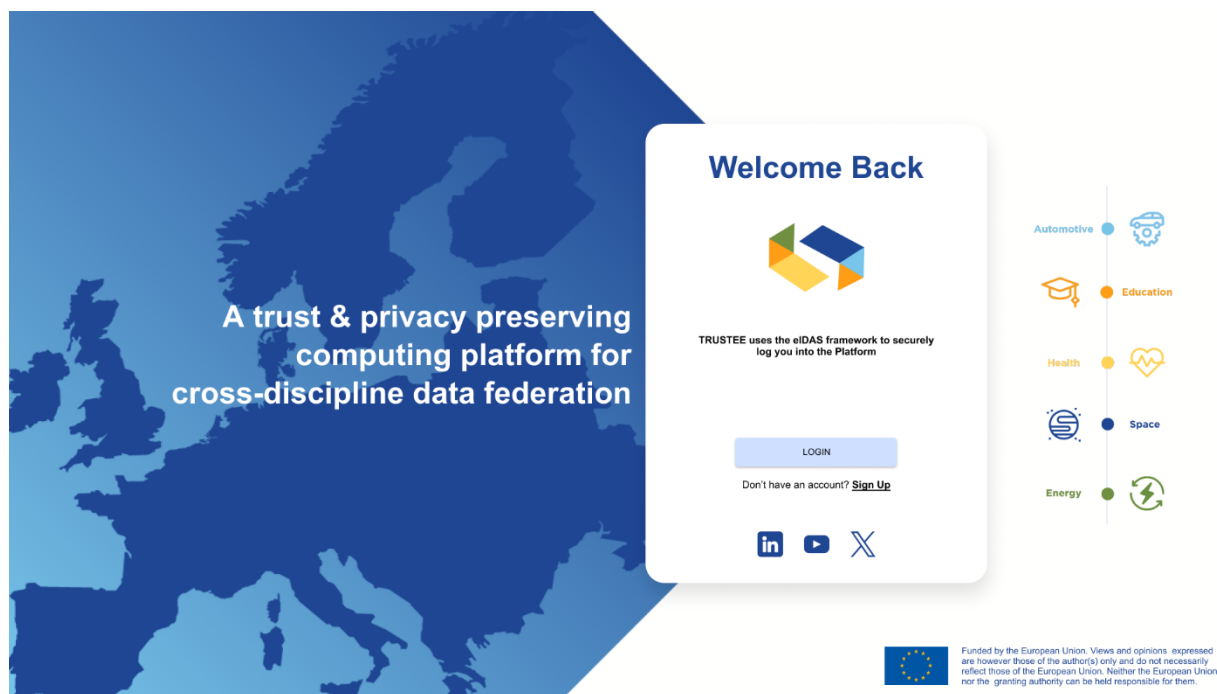


Figure 31: DA Log-in page

Sign-up Page

The Sign-up mock-up page shows how the user registration process in the TRUSTEE Platform will supposedly appear and is presented in Figure 32.



Figure 32: DA Sign-up page

Home Page

Figure 33 depicts the home page of the TRUSTEE. In this page there is a specific set-up that will stay consistent for all DA pages (except login and sign up), giving a sense of unity and continuity. That includes the upper heading line that includes the TRUSTEE logo and the profile info and settings, the navigation side bar, the main page, and the lower space (footer) where there is information presented relating to the TRUSTEE platform and project.

In more detail, the footer includes: i) the sitemap of the platform, ii) links to the TRUSTEE's Privacy Policy and use of terms, iii) a link to the contact information of the TRUSTEE, iv) a link to the TRUSTEE website, where the user may find available information for the project, v) links to the social media of TRUSTEE, where the user can be informed about news related to the project, and vi) the acknowledgement of European Union (EU) funding and the EU flag, according to the Communication and Visibility rules for European Union funding programmes for 2021-2027 [31].

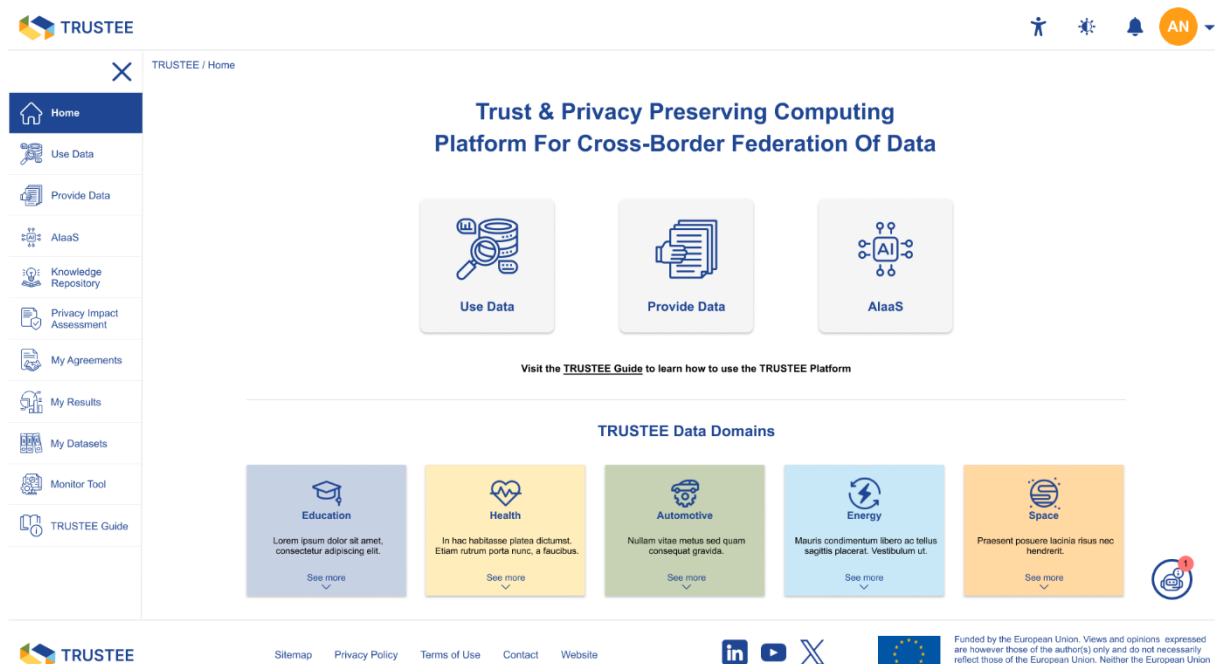


Figure 33: DA Home Page

Use Data Page

After entering the Home page, the users can select to access one of the components from the navigation bar. When the users select the Use Data tab, they are landed to the page into which they can insert and submit their search criteria regarding the datasets they wish to use. As depicted in Figure 34 below, this page informs the users of the “Use Data” workflow. In addition, Figure 35 presents the page that is provided to the user in order to begin the process of data use by submitting their search criteria in order to construct their own unique search query. More details about the mock-ups of the subsequent pages presented to the user during this process will be provided further below in the mock-up flow for the TRUSTEE Data Consumer.

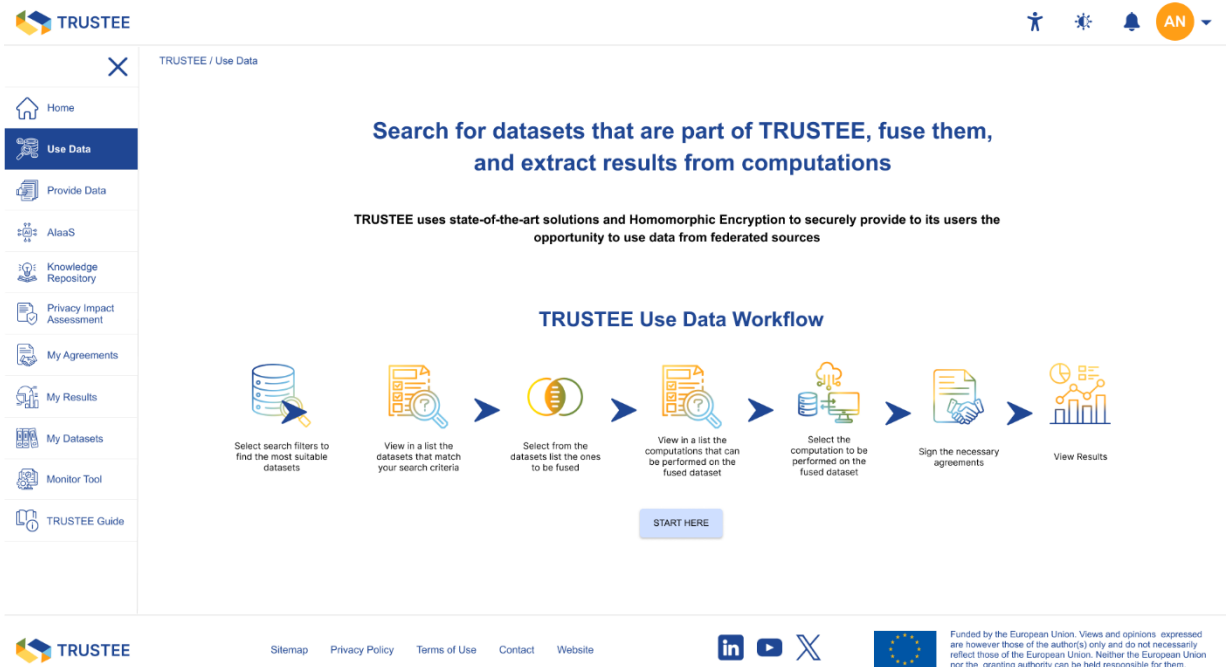


Figure 34: DA Use Data Page Entry Point

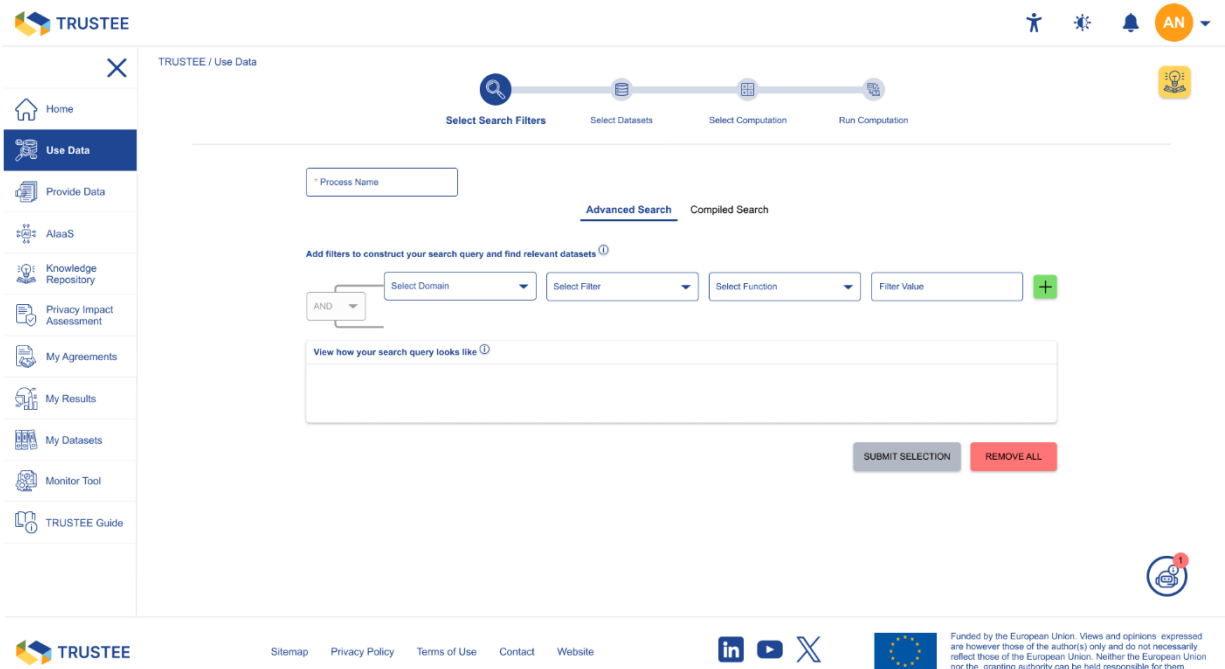


Figure 35: DA Use Data Page Search point

Provide Data Page

Instead of selecting the Use Data page, the user can select the Provide Data page from the navigation sidebar, which lands them to the initial page of the Provide Data process, providing the user with information about the “Provide Data” workflow, as depicted in Figure 36. As soon as the user clicks the Provide Data button they are redirected to the My Agreements page through a redirection to the UI of the STM ABB, which handles the next steps of the process.

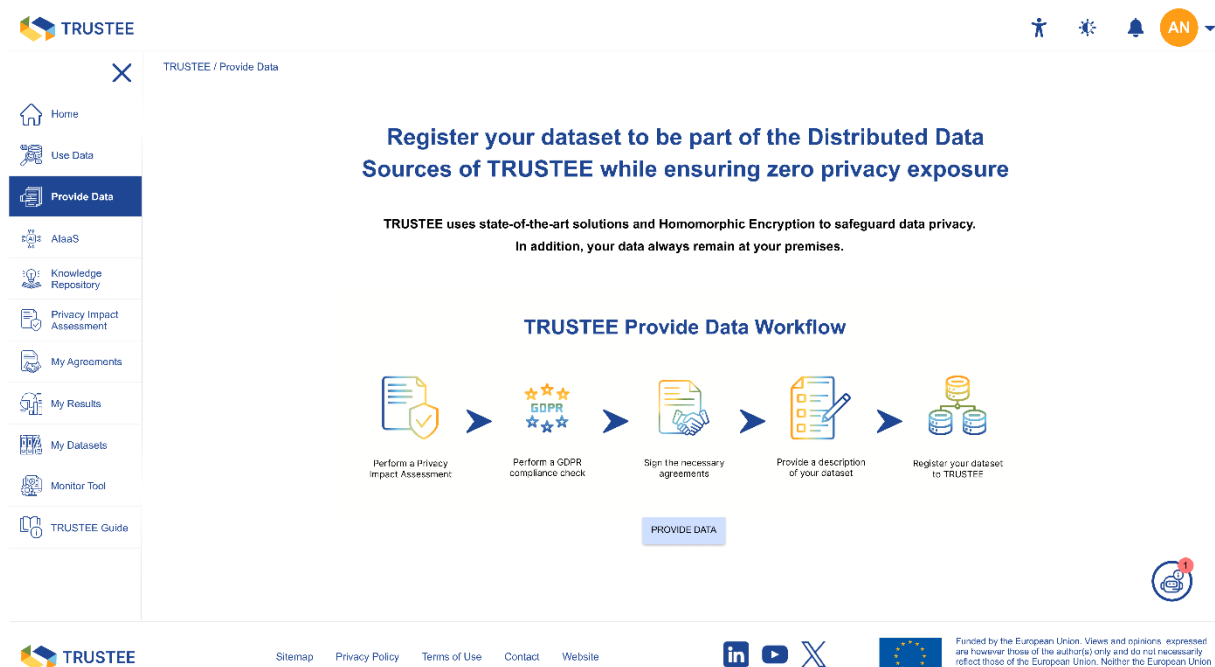


Figure 36: DA Provide Data Page Entry Point

Knowledge Repository Page

Figure 37 below shows the mock-up of the Knowledge Repository page of the DA, where the user will be able to explore and/or use past search query, fusion, and/or computation results of “Use Data” processes conducted by other Data Consumers of the TRUSTEE Platform and published, upon their consent, in the KR to be accessible to all users of the Platform.

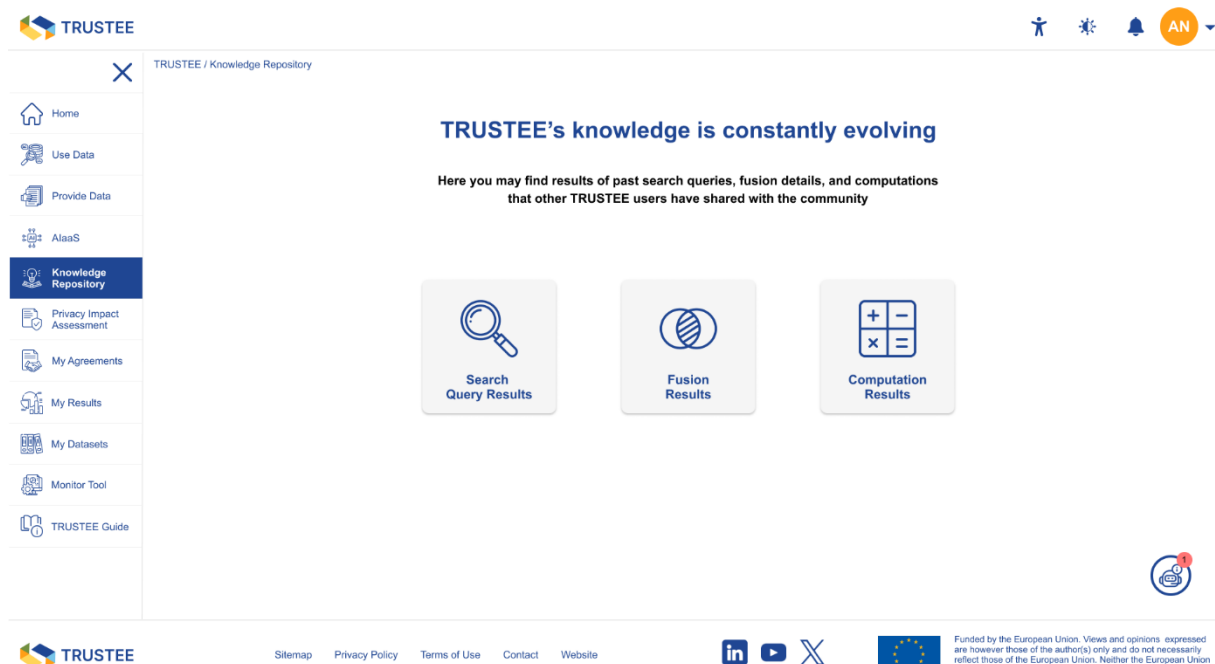


Figure 37: DA KR Page

My Results Page

At the beginning of the user's session, every time they enter the TRUSTEE Platform, the My Results page is empty, as it is showcased in Figure 38, until they perform a "Data Use" workflow and populate it with results. During the user's session the results of the "Data Use" processes they conduct are temporarily saved and displayed in this page; however, such results are discarded when the session of the user is deactivated and only results that the user consented to be published in the KR will be stored within TRUSTEE and be accessible for future use, not only by the user who stored them but also by other TRUSTEE users. In addition, Figure 39 illustrates the appearance of the My Results page when it contains "Data Use" workflow results.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

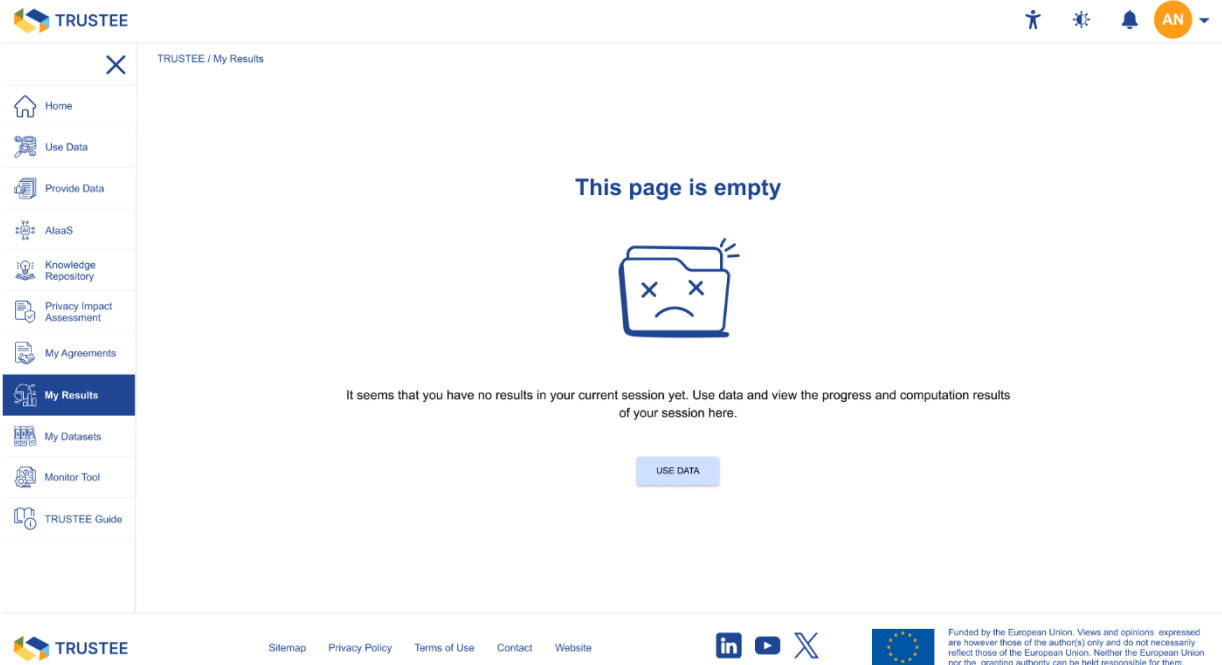


Figure 38: DA My Results Page without results

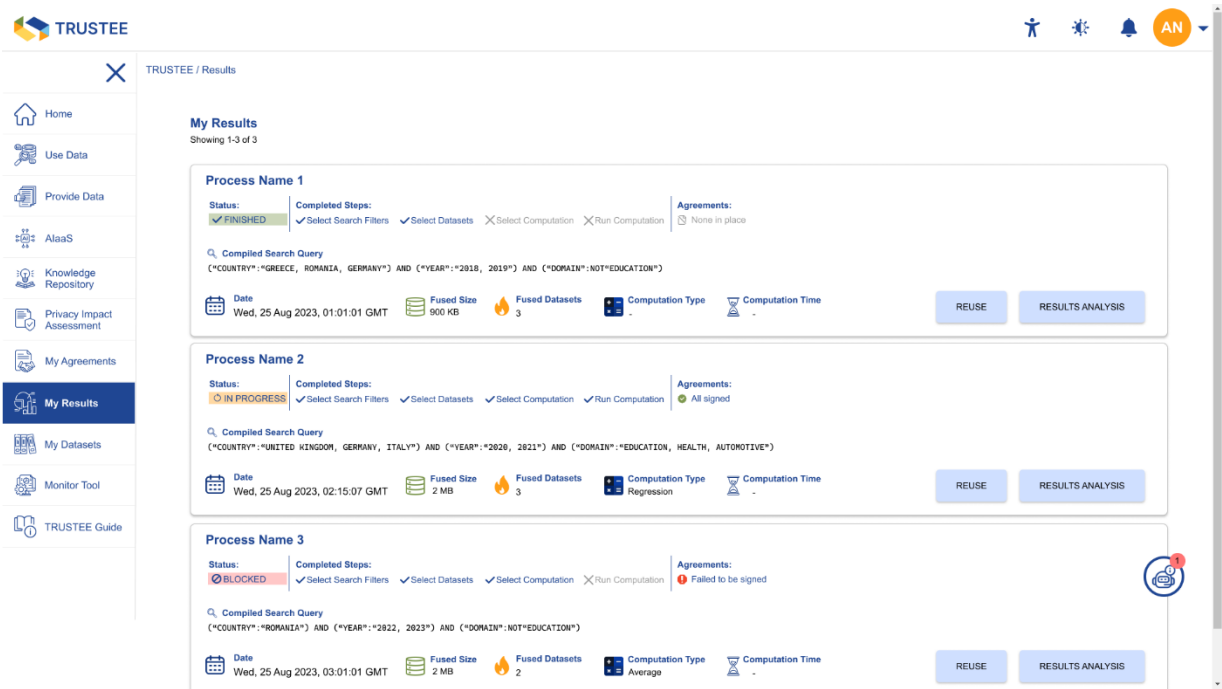


Figure 39: DA My Results Page populated with results

My Datasets Page

As long as a Data Provider has not registered any of their datasets to be part of the Distributed Data Sources of TRUSTEE, the My Datasets page remains empty as depicted in Figure 40. When a dataset has been registered then the page is populated as showcased in Figure 41.

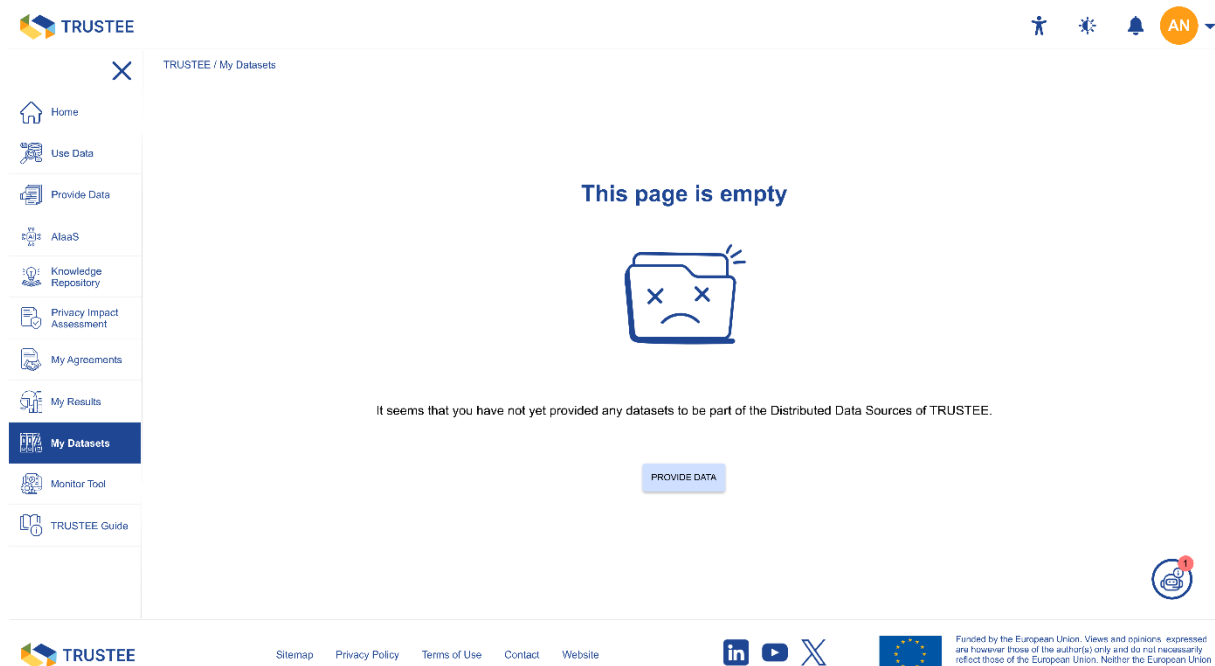


Figure 40: DA My Datasets Page empty

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

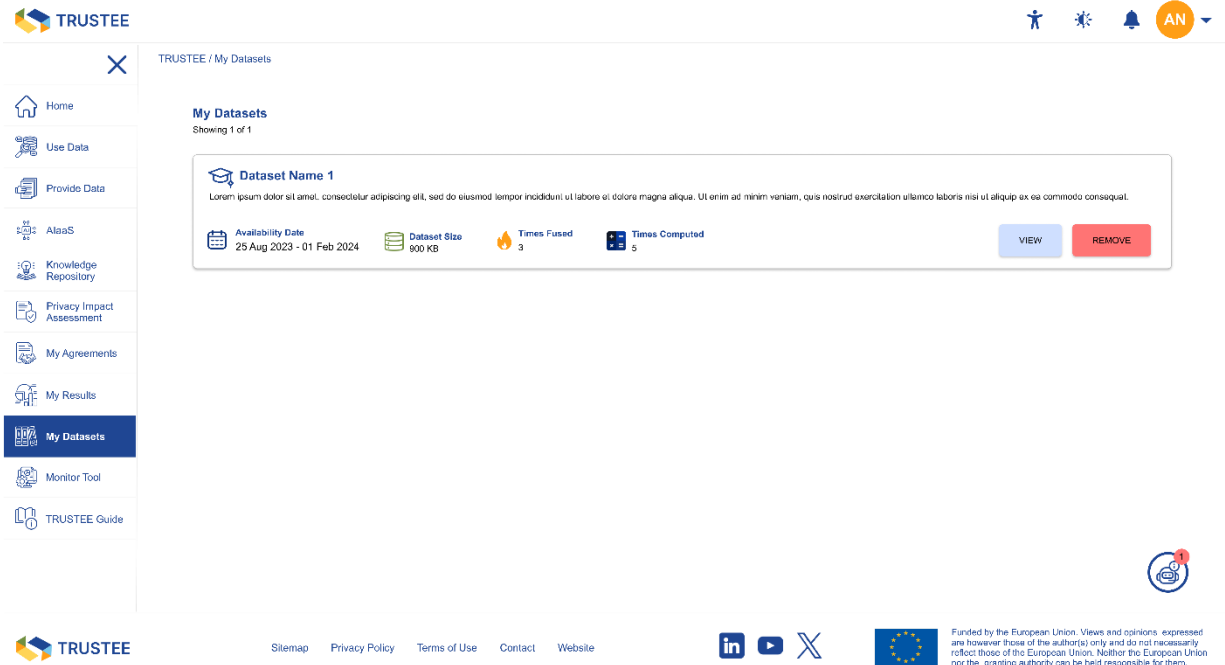


Figure 41: DA My Datasets Page populated with datasets

TRUSTEE Data Consumer Use Case Flow (Mock-up UI Frames sequence)

This section illustrates the step-by-step process of the TRUSTEE Data Consumer Use Case flow (Figure 42 - Figure 61), as described in the “Operational Description” sub-section of the “Implementation” chapter of this document, using the developed UI mock-ups presented above.

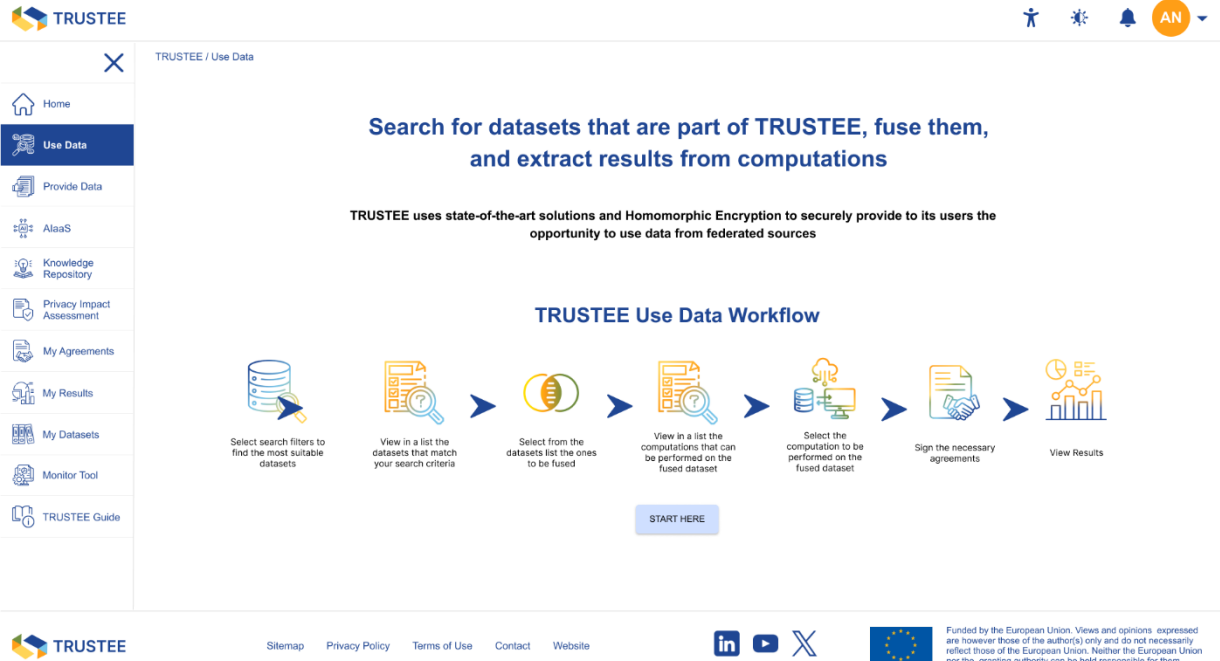


Figure 42: DA Use Data Initial Page

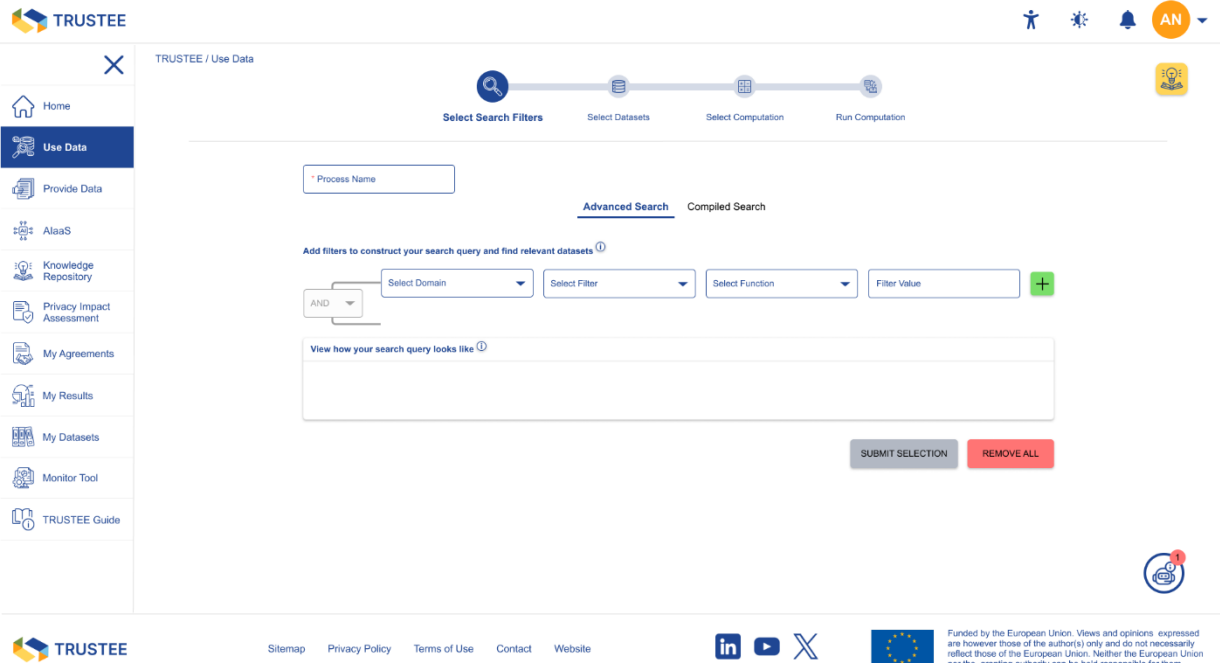


Figure 43: DA Use Data Page Filtering

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The screenshot shows the TRUSTEE web application interface. The top navigation bar includes the TRUSTEE logo, a user profile icon, and a notification bell. The left sidebar contains a menu with options: Home, Use Data (selected), Provide Data, AlaaS, Knowledge Repository, Privacy Impact Assessment, My Agreements, My Results, My Datasets, Monitor Tool, and TRUSTEE Guide. The main content area is titled "TRUSTEE / Use Data" and features a progress bar with four steps: Select Search Filters, Select Datasets, Select Computation, and Run Computation. Below the progress bar, there is a "Process Name" input field. The "Advanced Search" tab is active, showing a section titled "Add filters to construct your search query and find relevant datasets". This section includes a table with columns for "AND", "Select Domain", "Select Filter", "Select Function", and "Filter Value". The first row shows "AUTOMOTIVE" for the domain, "COUNTRY" for the filter, "IS EQUAL TO" for the function, and "GREECE" for the filter value. A second row is partially visible with "Select Domain", "Select Filter", "Select Function", and "Filter Value". Below the table, a preview box shows the generated query: "(((DOMAIN: 'AUTOMOTIVE')) AND (COUNTRY: GREECE)) AND ()". At the bottom right of the search area are buttons for "SUBMIT SELECTION" and "REMOVE ALL". The footer contains the TRUSTEE logo, a sitemap, privacy policy, terms of use, contact, and website links, along with social media icons and a European Union funding notice.

Figure 44: DA Use Data Page Filtering Operation in Advanced Search mode

The screenshot shows the TRUSTEE web application interface in the "Compiled Search" mode. The top navigation bar and left sidebar are identical to the previous figure. The main content area is titled "TRUSTEE / Use Data" and features a progress bar with four steps: Select Search Filters, Select Datasets, Select Computation, and Run Computation. Below the progress bar, there is a "Process Name" input field with the text "My Process-01-AUTO". The "Compiled Search" tab is active, showing a section titled "Enter keywords, phrases, or Boolean expression". This section includes three dropdown menus for "Select Domain", "Select Filter", and "Select Function". Below these, a text box contains the compiled query: "(*COUNTRY: 'GREECE, GERMANY') AND (*YEAR: '2020, 2021') AND (*COUNTRY: NOT 'ROMANIA')". At the bottom right of the search area are buttons for "SUBMIT SELECTION" and "CLEAR". The footer contains the TRUSTEE logo, a sitemap, privacy policy, terms of use, contact, and website links, along with social media icons and a European Union funding notice.

Figure 45: DA Use Data Page Filtering Operation in Compiled Search mode

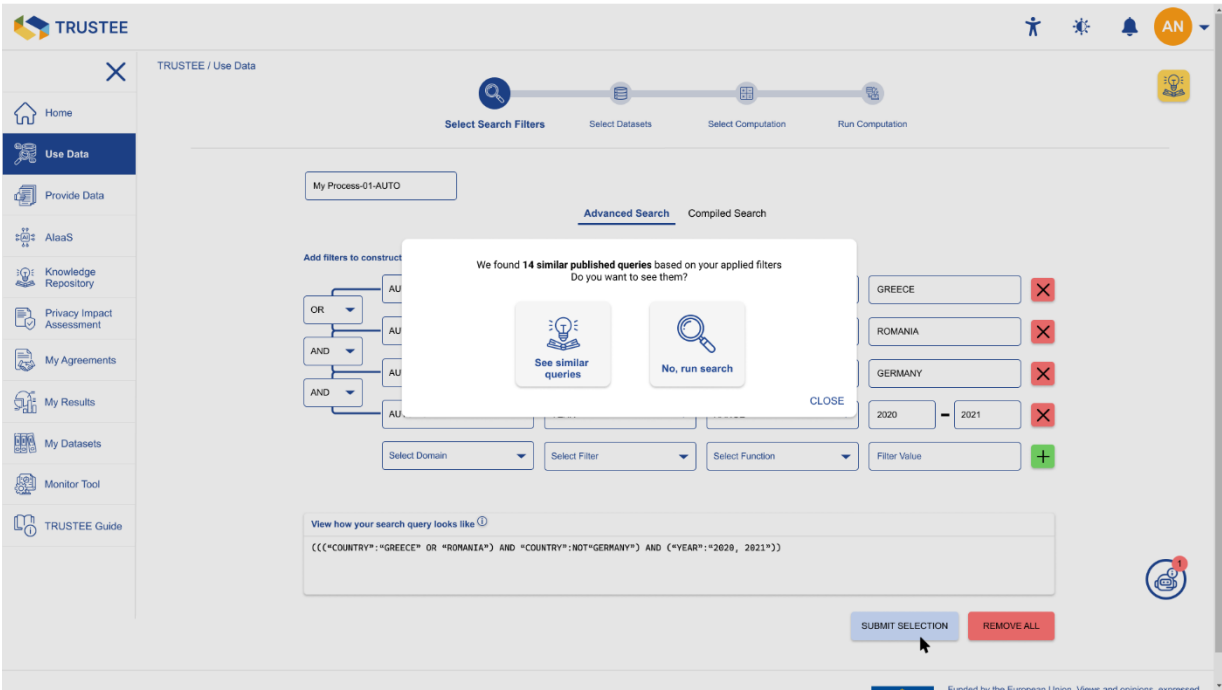


Figure 46: DA Use Data Page Similar published Queries Popup

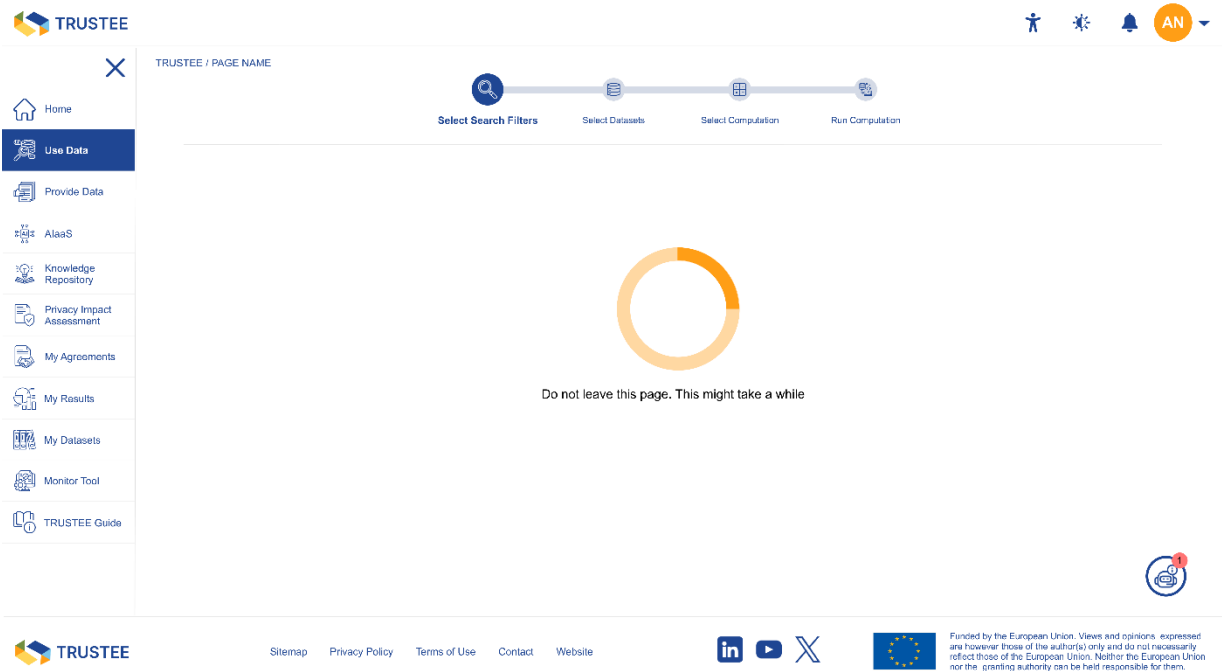


Figure 47: DA Page Loader - Loading Search Query Results

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

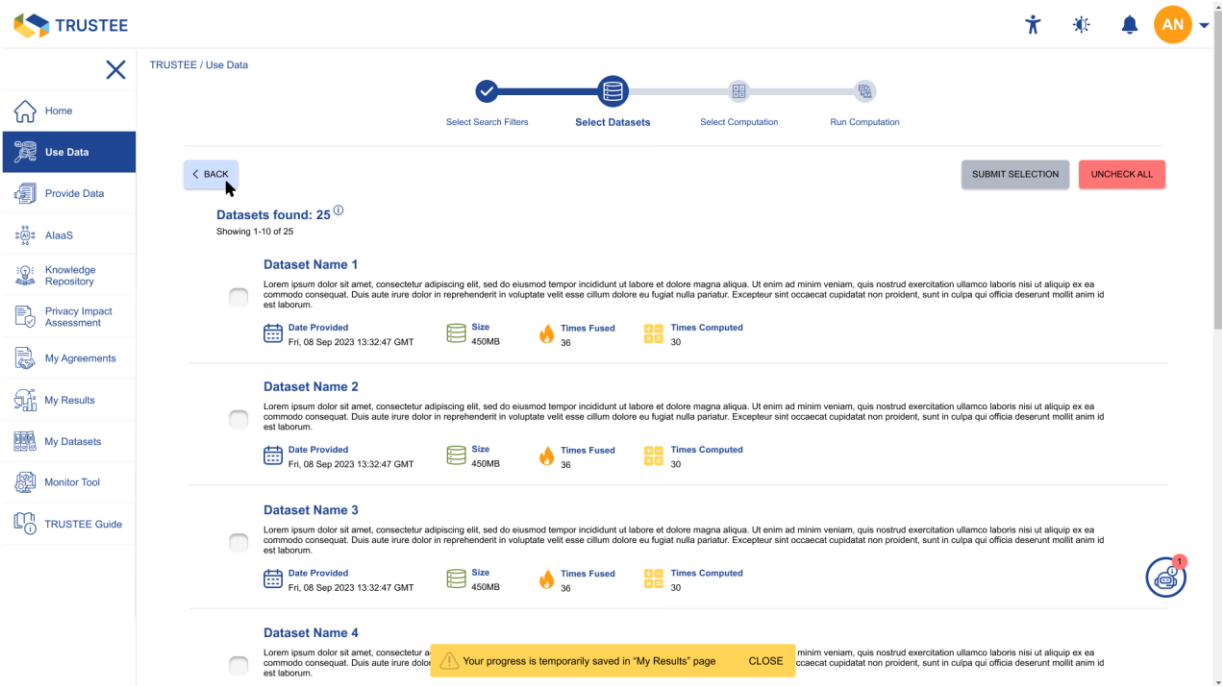


Figure 48: DA Use Data Page Select Datasets for fusion

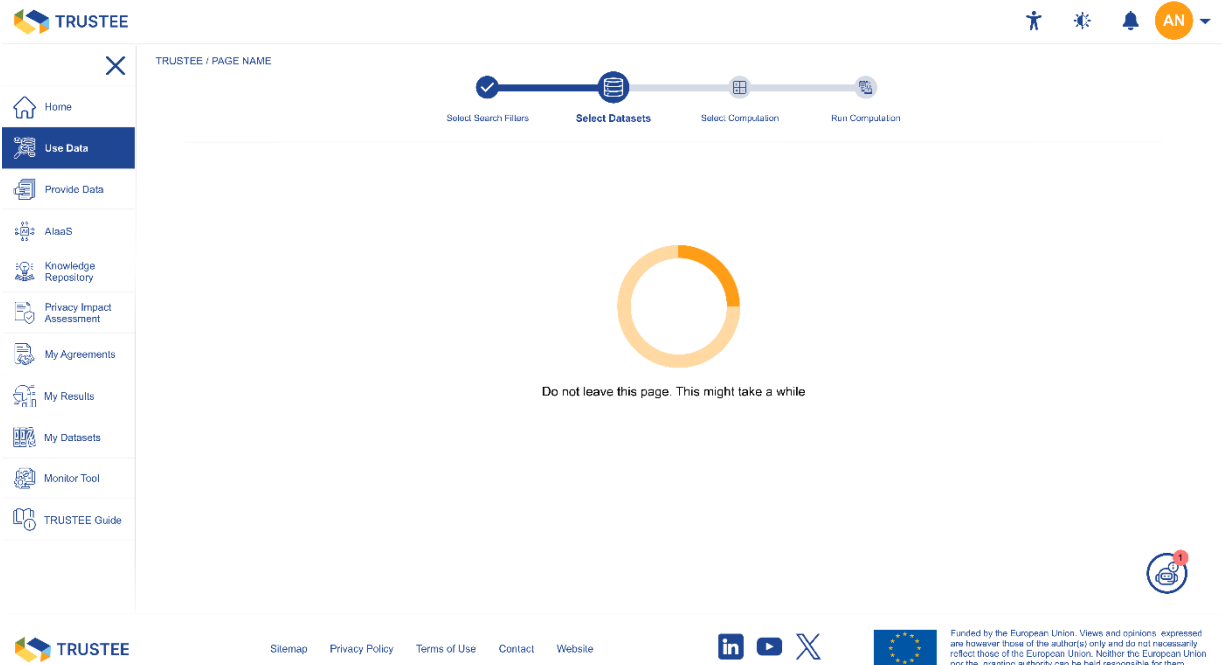


Figure 49: DA Page Loader – Fetching available computations that can be performed on the fused dataset

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

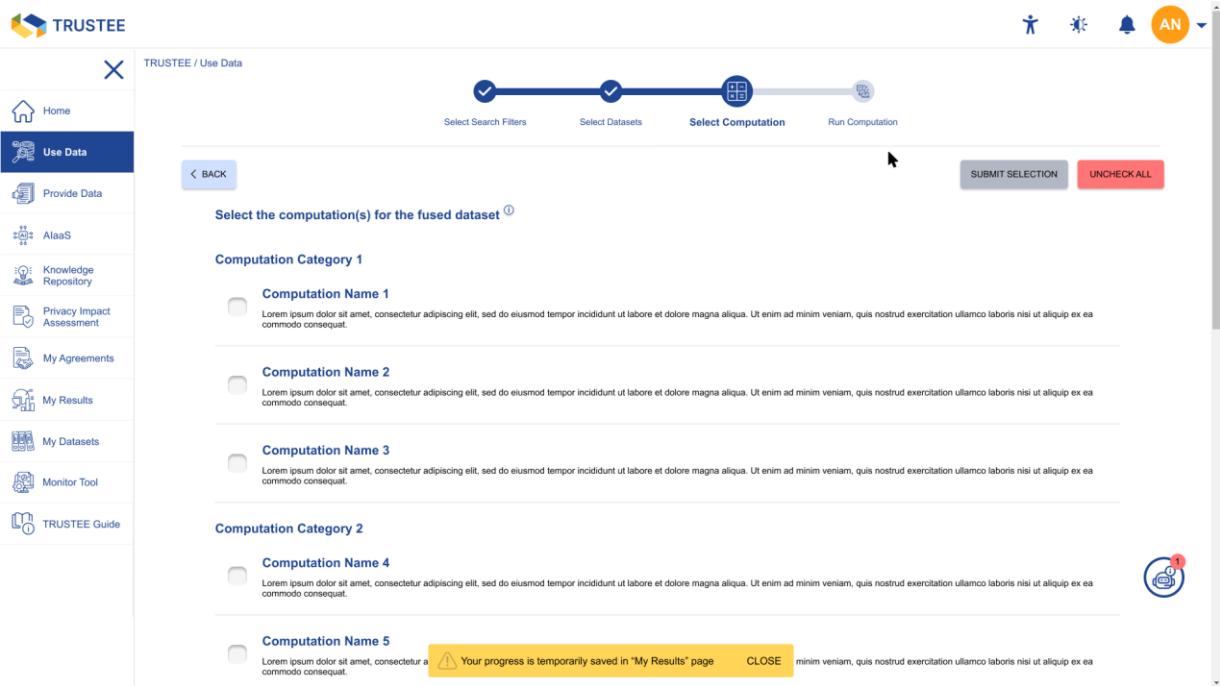


Figure 50: DA Use Data Page Select Computation

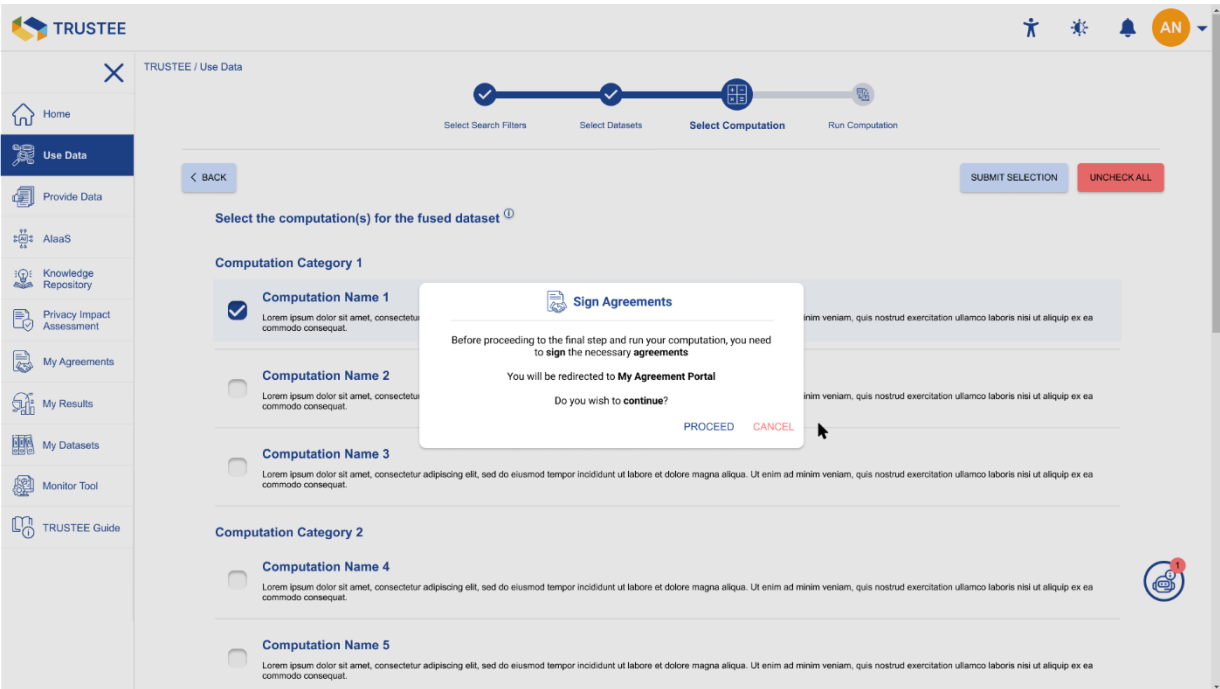


Figure 51: DA Use Data Page STM Redirection

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

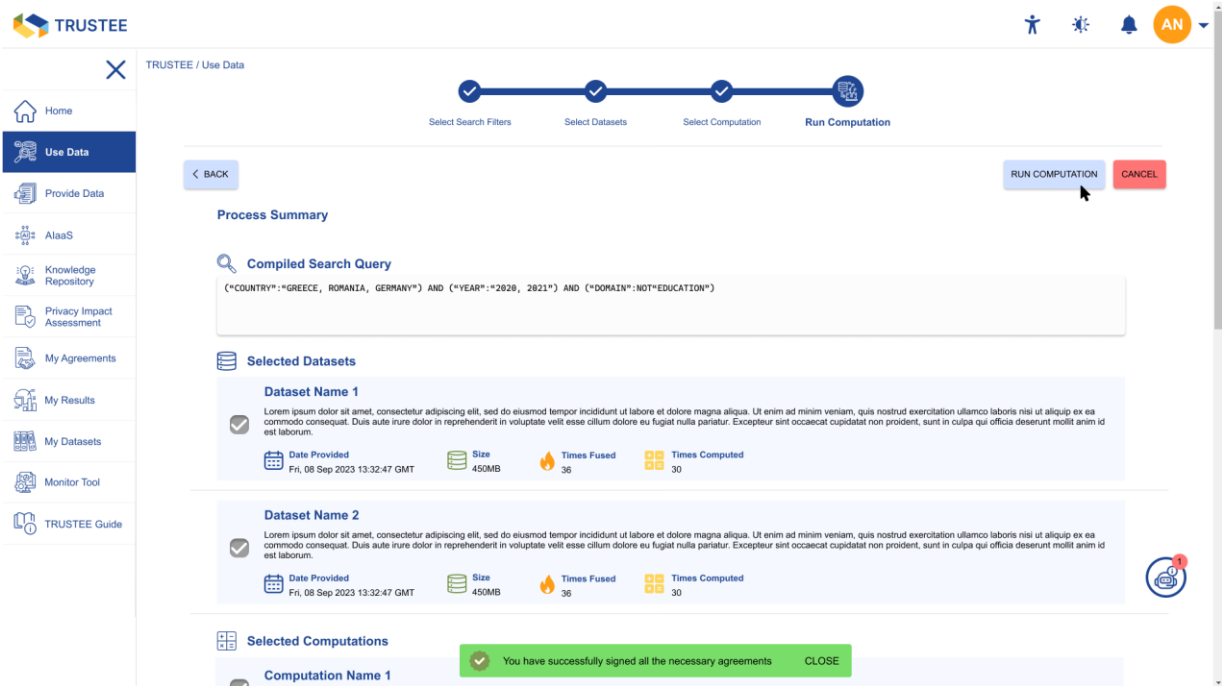


Figure 52: DA Use Data Page Agreements Success

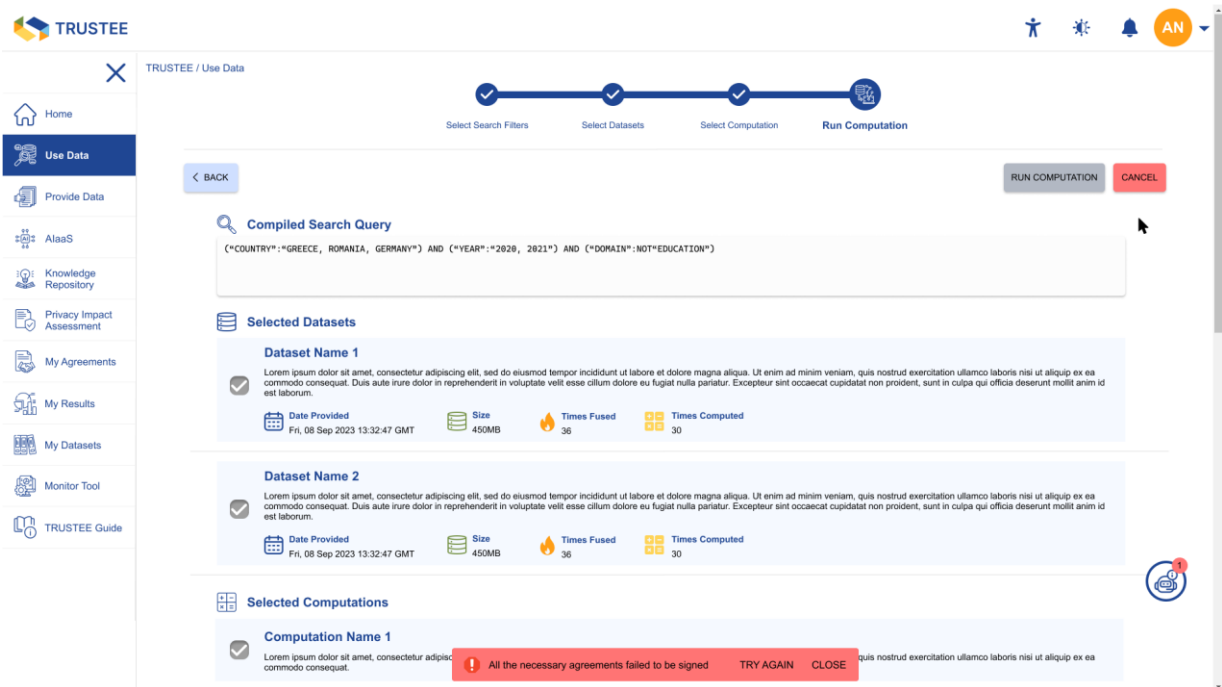


Figure 53: DA Use Data Page Agreements Failure

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

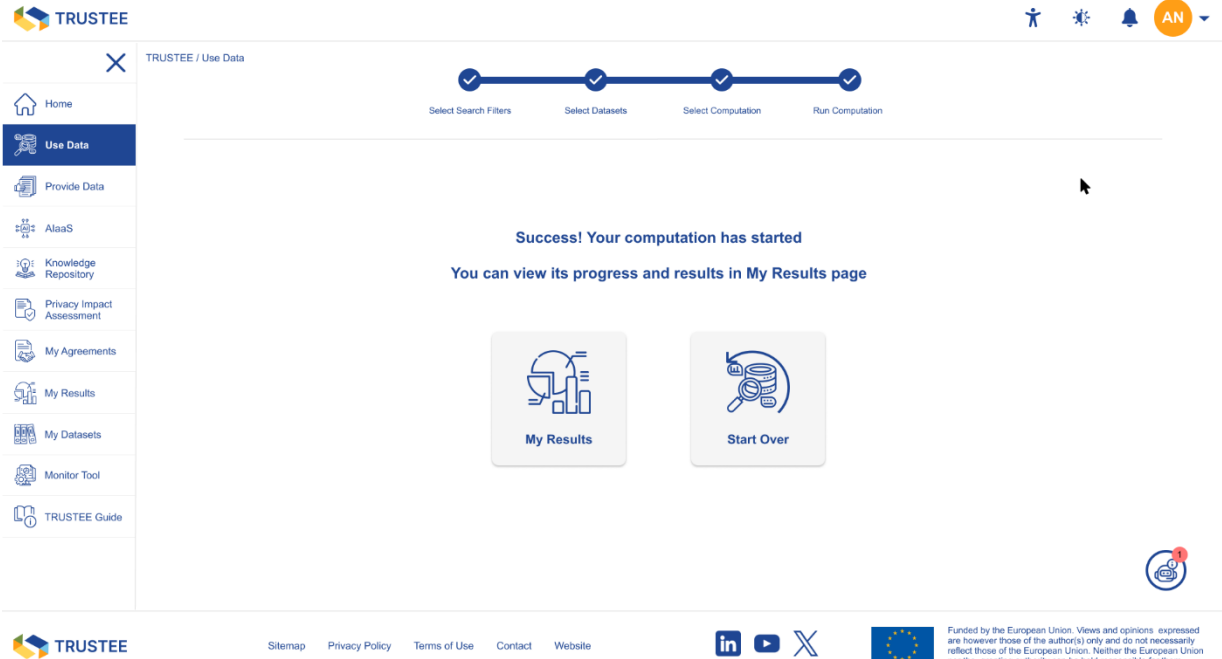


Figure 54: DA Use Data Page Computation Start

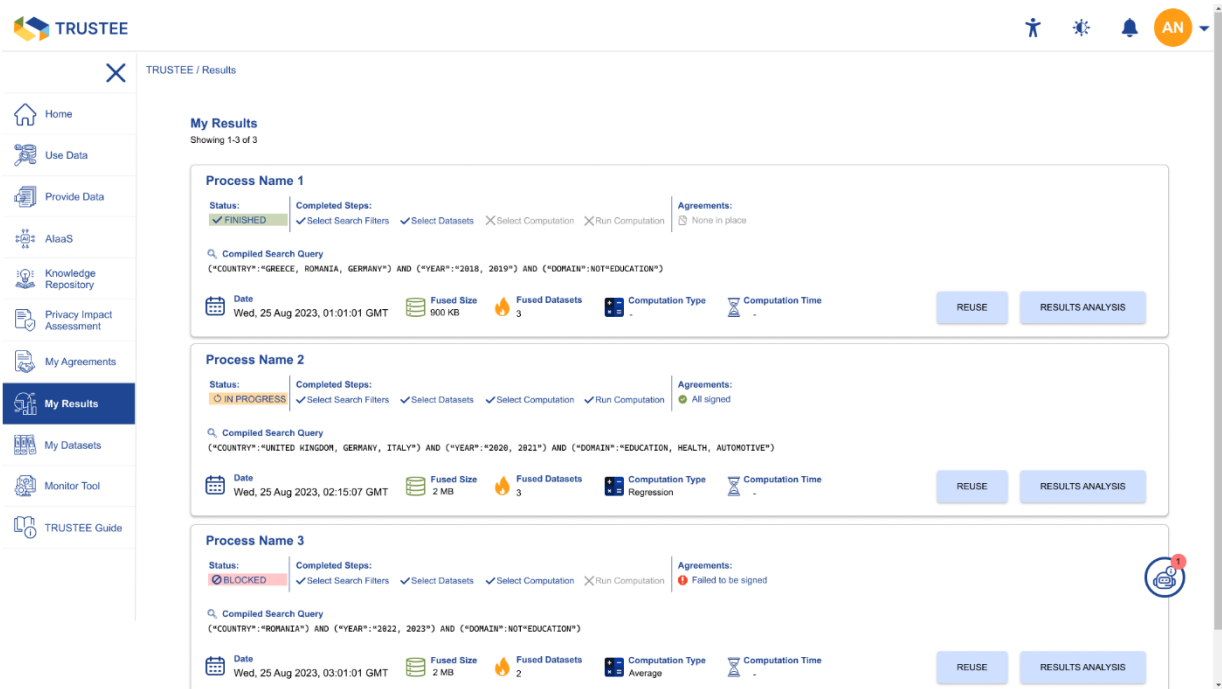


Figure 55: DA My Results Process In Progress

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

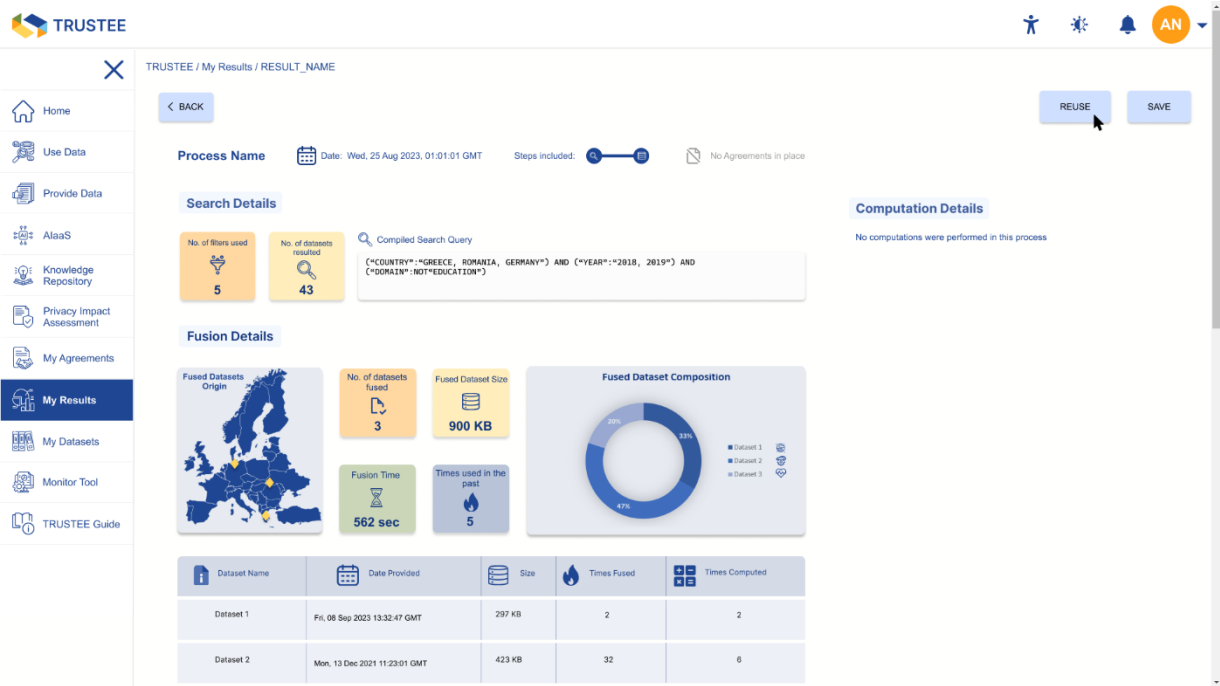


Figure 56: DA My Results Page Selected Process without computation details

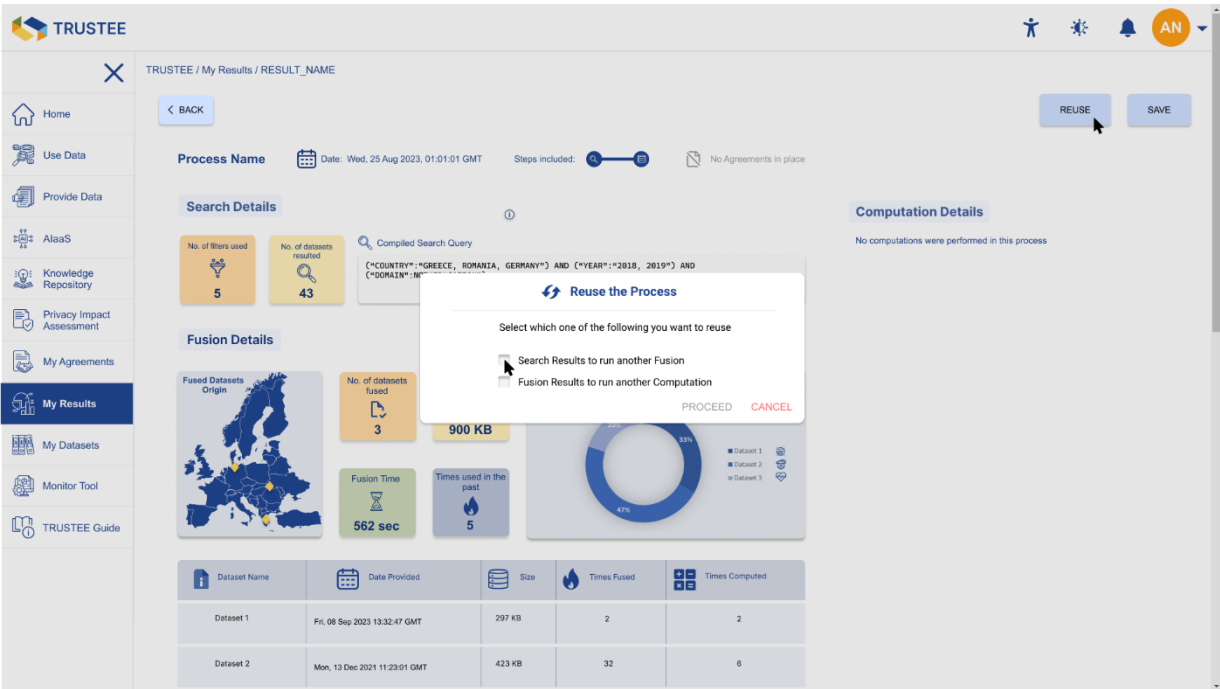


Figure 57: DA My Results Page Reuse Popup



Figure 58: DA My Results Page with computation details

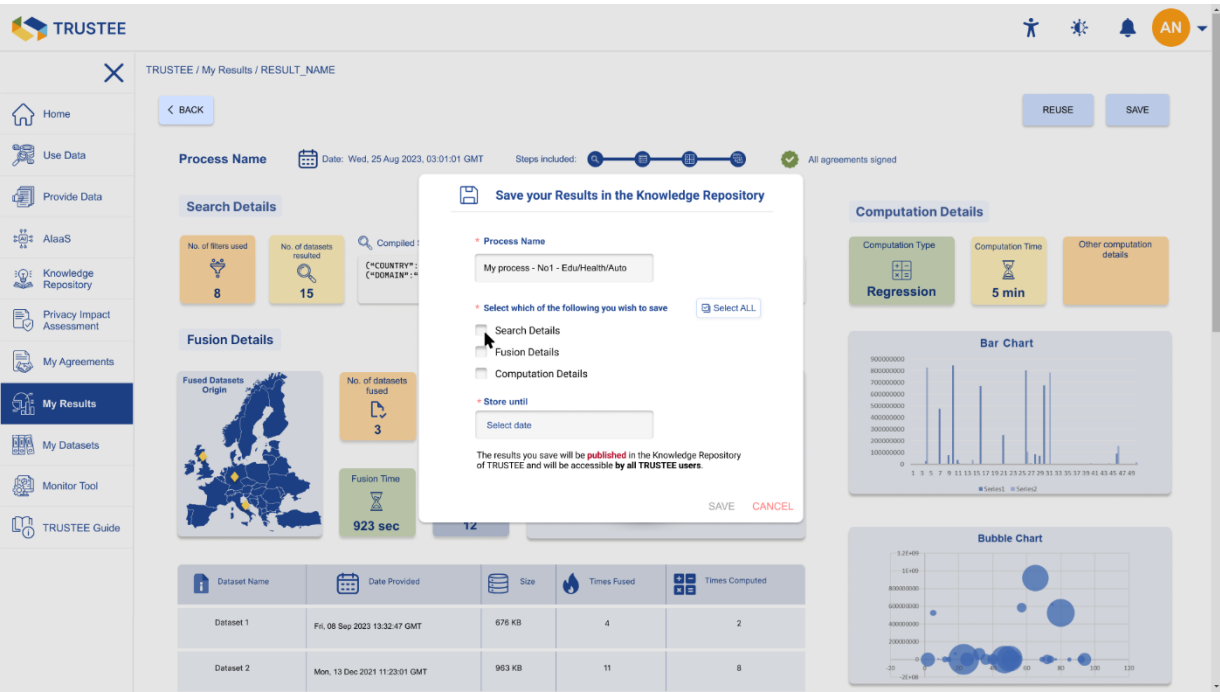


Figure 59: DA My Results Page Save Process in the Knowledge Repository pt.1

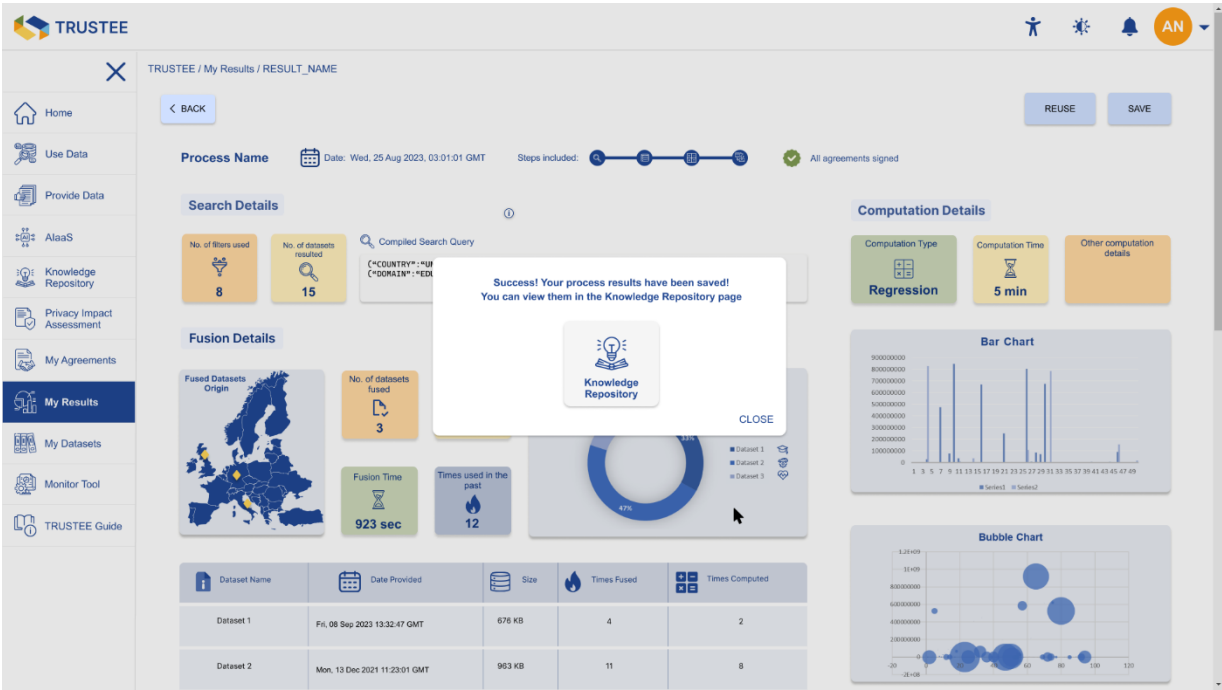


Figure 60: DA My Results Page Save Process Results in the Knowledge Repository pt.2

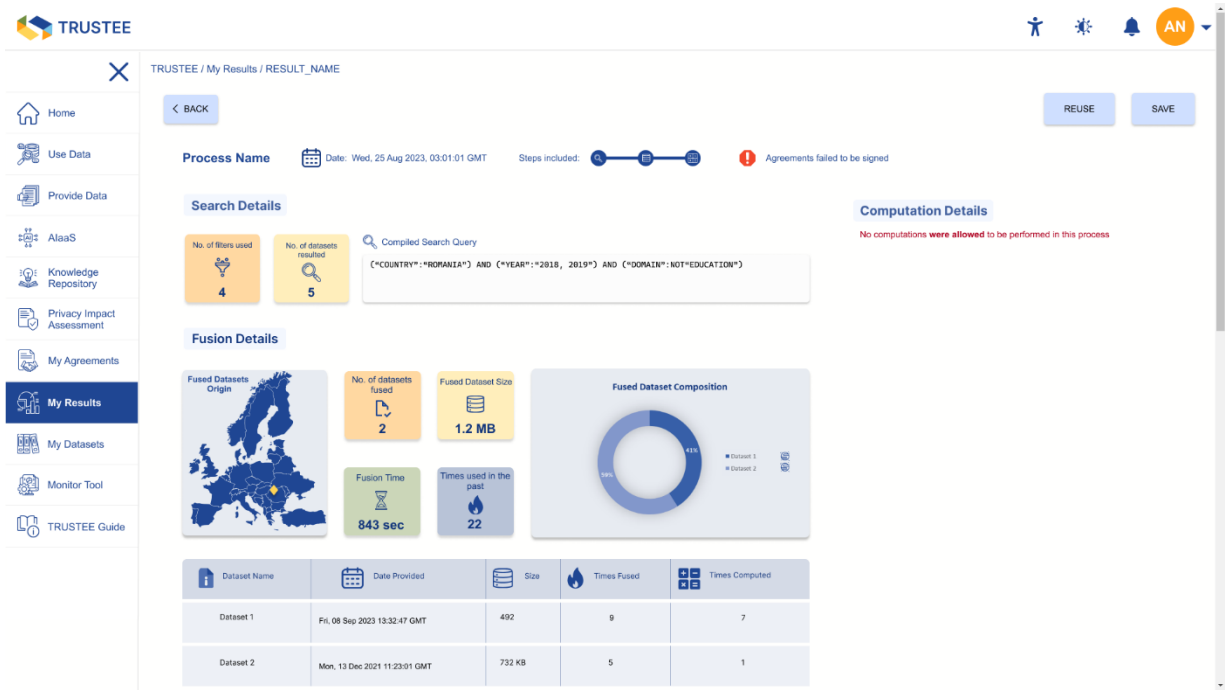


Figure 61: DA My Results Page Agreements not signed – computation now allowed to be performed

TRUSTEE Data Provider Use Case Flow (Mock-up UI Frames sequence)

This section illustrates the step-by-step process of the TRUSTEE Data Provider Use Case flow (Figure 62 - Figure 64), as described in the “Operational Description” sub-section of the “Implementation” chapter of this document, using the developed UI mock-ups presented above.

The screenshot displays the TRUSTEE web application interface. On the left is a sidebar with navigation links: Home, Use Data, Provide Data (highlighted), AlaaS, Knowledge Repository, Privacy Impact Assessment, My Agreements, My Results, My Datasets, Monitor Tool, and TRUSTEE Guide. The main content area is titled 'Dataset Description' and contains four sections: 'Dataset General Information' (with fields for Dataset Name, Domain of Dataset, Dataset Size, Unit, No. of Records, and a text area for Dataset Description), 'Dataset Access Information' (with fields for Access URL, Access Start Date, and Access End Date), 'Metadata Schema' (with a dropdown for Select Metadata Attributes and a button for Create another Attribute ...), and 'Dataset Structure' (with a table for columns and value types, and a button for Create another Column ...). At the bottom of the form are buttons for BACK, SUBMIT, and CLEAR ALL. The footer includes the TRUSTEE logo, a sitemap, privacy policy, terms of use, contact, and website links, social media icons, the European Union flag, and a disclaimer: 'Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.'

Figure 62: DA Provide Data Page Submit Dataset Description

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Home

Use Data

Provide Data

AlaaS

Knowledge Repository

Privacy Impact Assessment

My Agreements

My Results

My Datasets

Monitor Tool

TRUSTEE Guide

Dataset Description

Dataset General information

Provide some general information about your dataset and a short description

Dataset Name 1

EDUCATION

900

KB

3000

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Dataset Access Information

Provide the URL where your dataset can be found, alongside a start and an end date during which you permit TRUSTEE to access your dataset

www.dataset_access_url.com

25 Aug 2023

01 Feb 2024

Metadata Schema

Select the metadata attributes that best characterize your dataset from the provided list. If you can not find the attribute you want, you may create it.

Country, Year, School Name, School Address

Create another Attribute ...

Dataset Structure

Provide the names of the columns of your dataset, in the order of appearance, as well as the their corresponding value types from the lists provided. If you can not find the column name or the value type you want you may create it.

0001

Grade Level

Create another Column ...

String

Create another Value Type ...

0002

Pedagogical Approach

Create another Column ...

String

Create another Value Type ...

ID

Column Name

Create another Column ...

Value Type

Create another Value Type ...

BACK

SAVE

CLEAR ALL

TRUSTEE

Sitemap Privacy Policy Terms of Use Contact Website

in

EU

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Figure 63: DA Provide Data Page Edit Dataset Description

TRUSTEE

TRUSTEE / My Datasets

Home

Use Data

Provide Data

AlaaS

Knowledge Repository

Privacy Impact Assessment

My Agreements

My Results

My Datasets

Monitor Tool

TRUSTEE Guide

My Datasets

Showing 1 of 1

Dataset Name 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Availability Date

25 Aug 2023 - 01 Feb 2024

Dataset Size

900 KB

Times Fused

3

Times Computed

5

VIEW

REMOVE

TRUSTEE

Sitemap Privacy Policy Terms of Use Contact Website

in

EU

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Figure 64: DA My Datasets Page

For demonstration purposes of the VA's high fidelity wireframes, a "Navigation" use-case will be showcased using the figures below.

High fidelity Wireframes

As presented in Figure 65, the user has not yet interacted with the VA, thus, it is in a collapsed state on the bottom right corner of the screen. When the user clicks on the icon or the popup message, the VA expands into its full view, as presented in Figure 66. There, the VA welcomes the user with a message and presents them with their main options to start the conversation. Then, the user clicks on the "Navigation" option and the message gets sent to the VA and based on its decision tree. After that, VA replies accordingly, asking the user where they want to navigate to as shown in Figure 67 and Figure 68 respectively. There the user is presented with more options on where to navigate and clicks on the "Knowledge Repository" where the message gets sent once again to the VA as shown in Figure 69. The VA then replies back and performs the redirection as presented in Figure 70. In contrast, Figure 71 shows that if the user decides to close the options box without choosing one, the VA cancels the navigation procedure and starts the conversation all over.

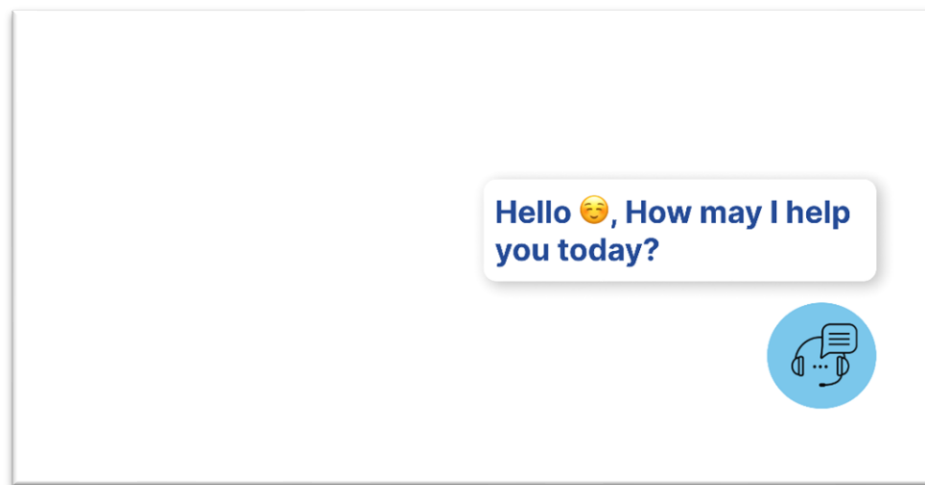


Figure 65: Virtual Assistant hidden state. Initial view

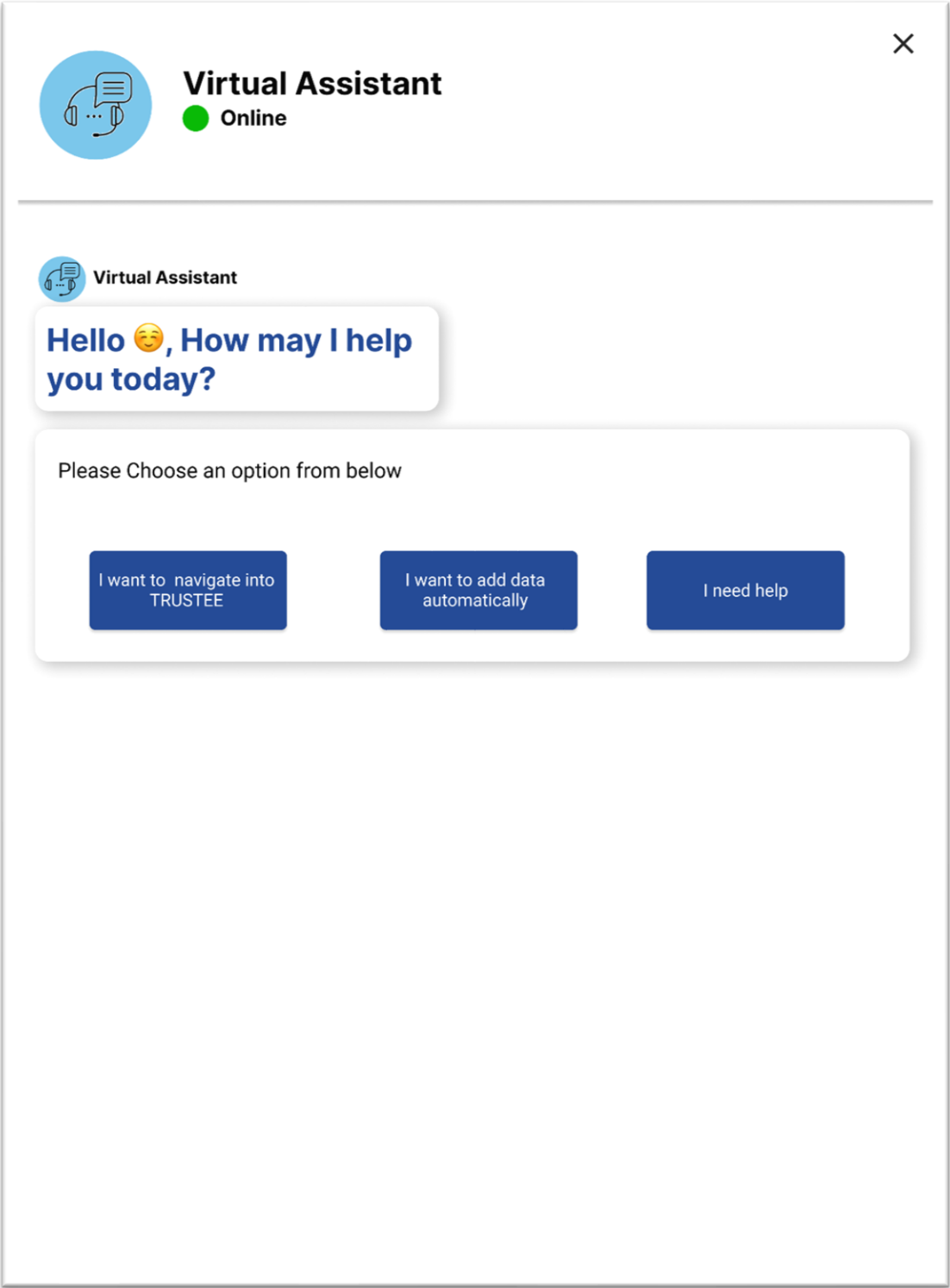


Figure 66: Virtual Assistant expanded state - First message

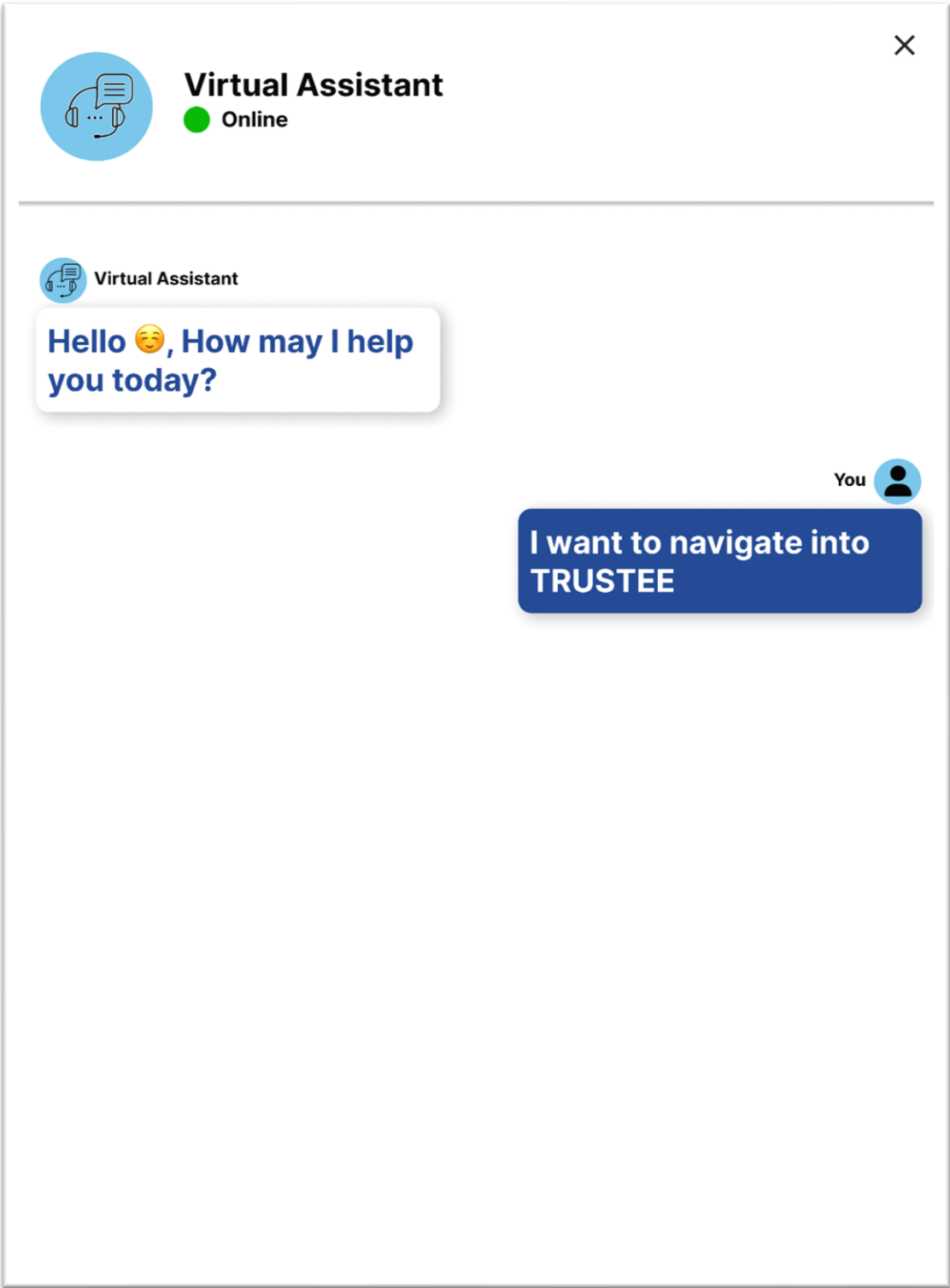


Figure 67: Virtual Assistant – Navigation use case #1

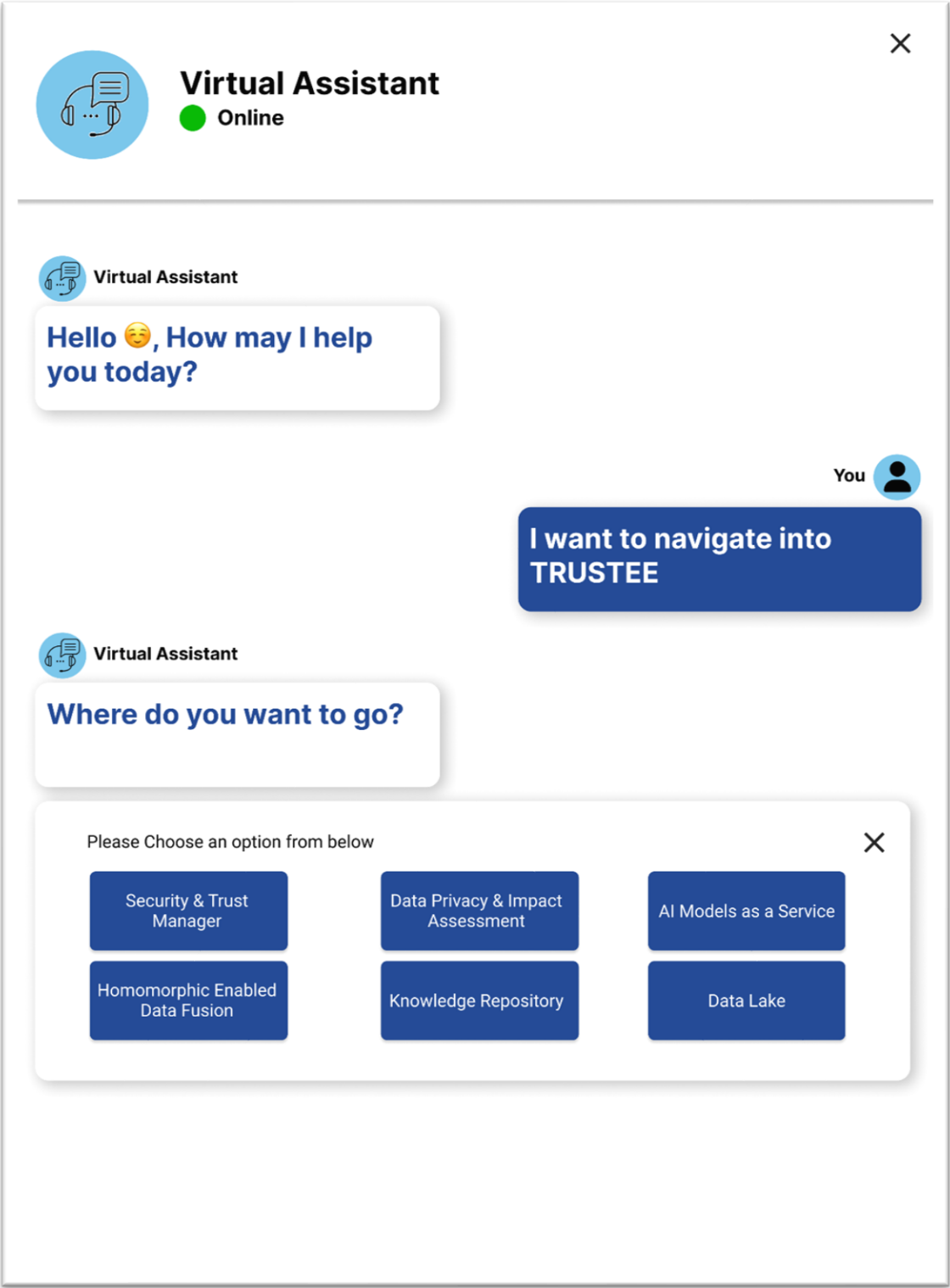


Figure 68: Virtual Assistant – Navigation use case #2

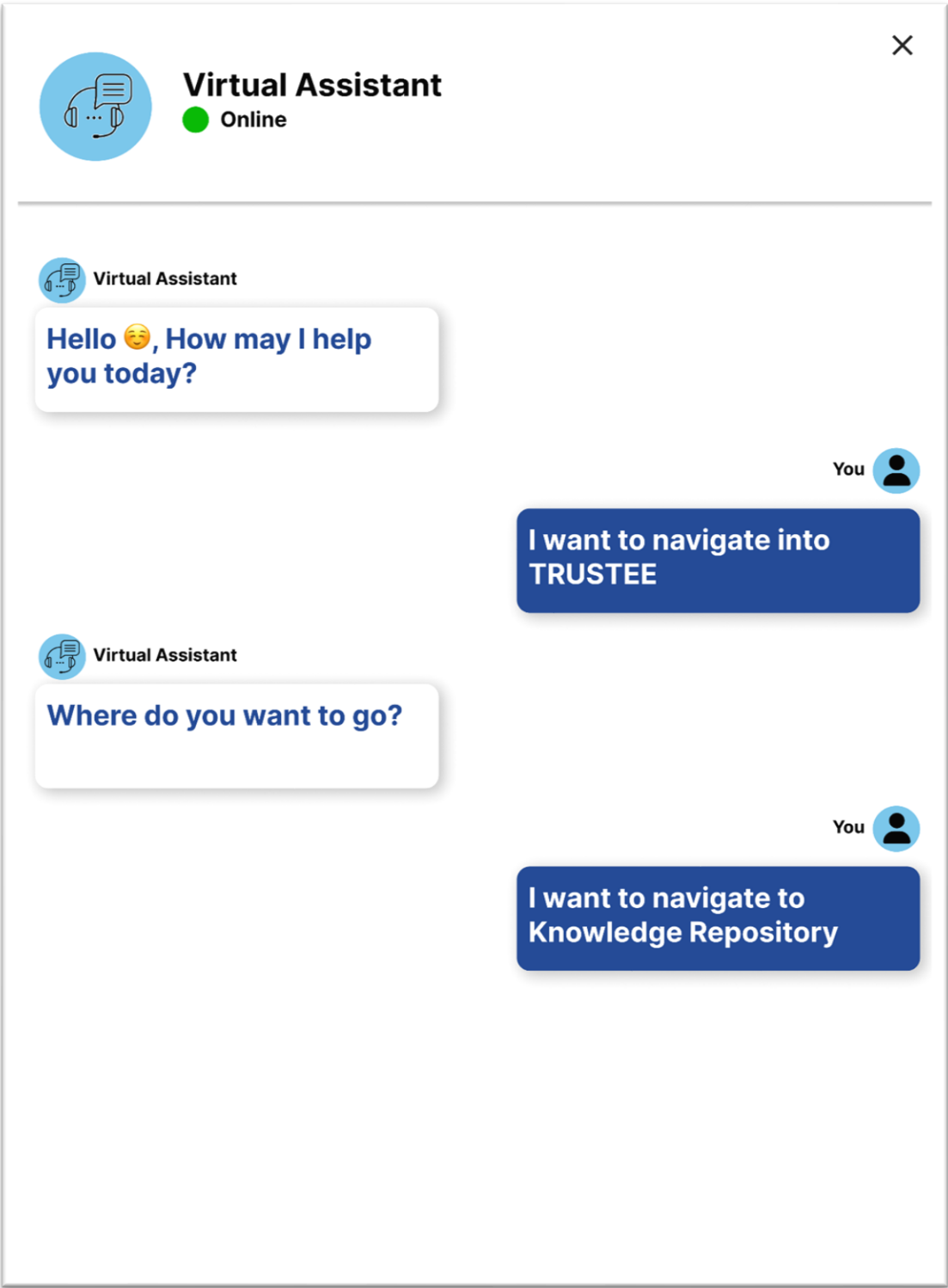


Figure 69: Virtual Assistant – Navigation use case #3

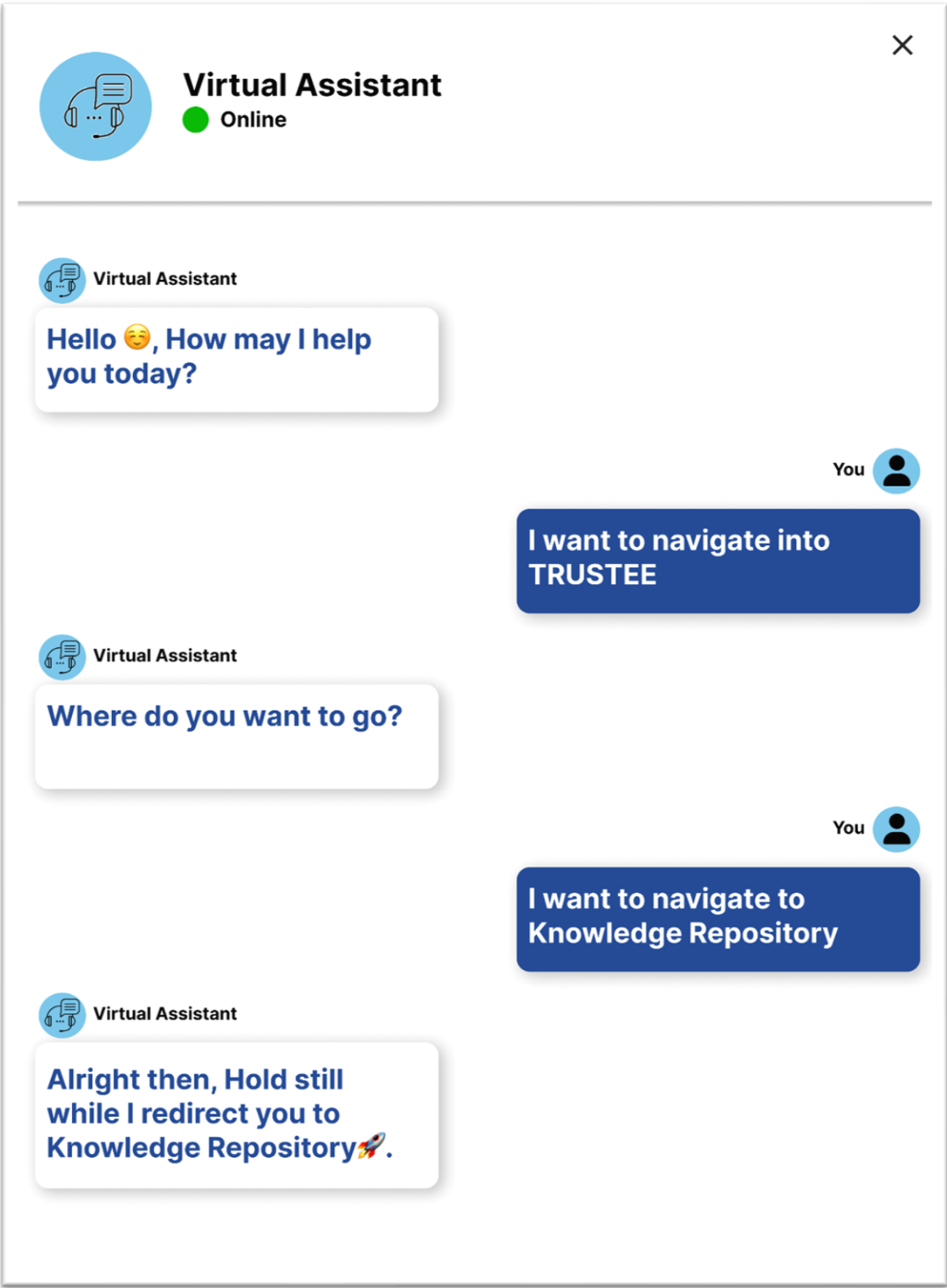


Figure 70: Virtual Assistant – Navigation use case #4

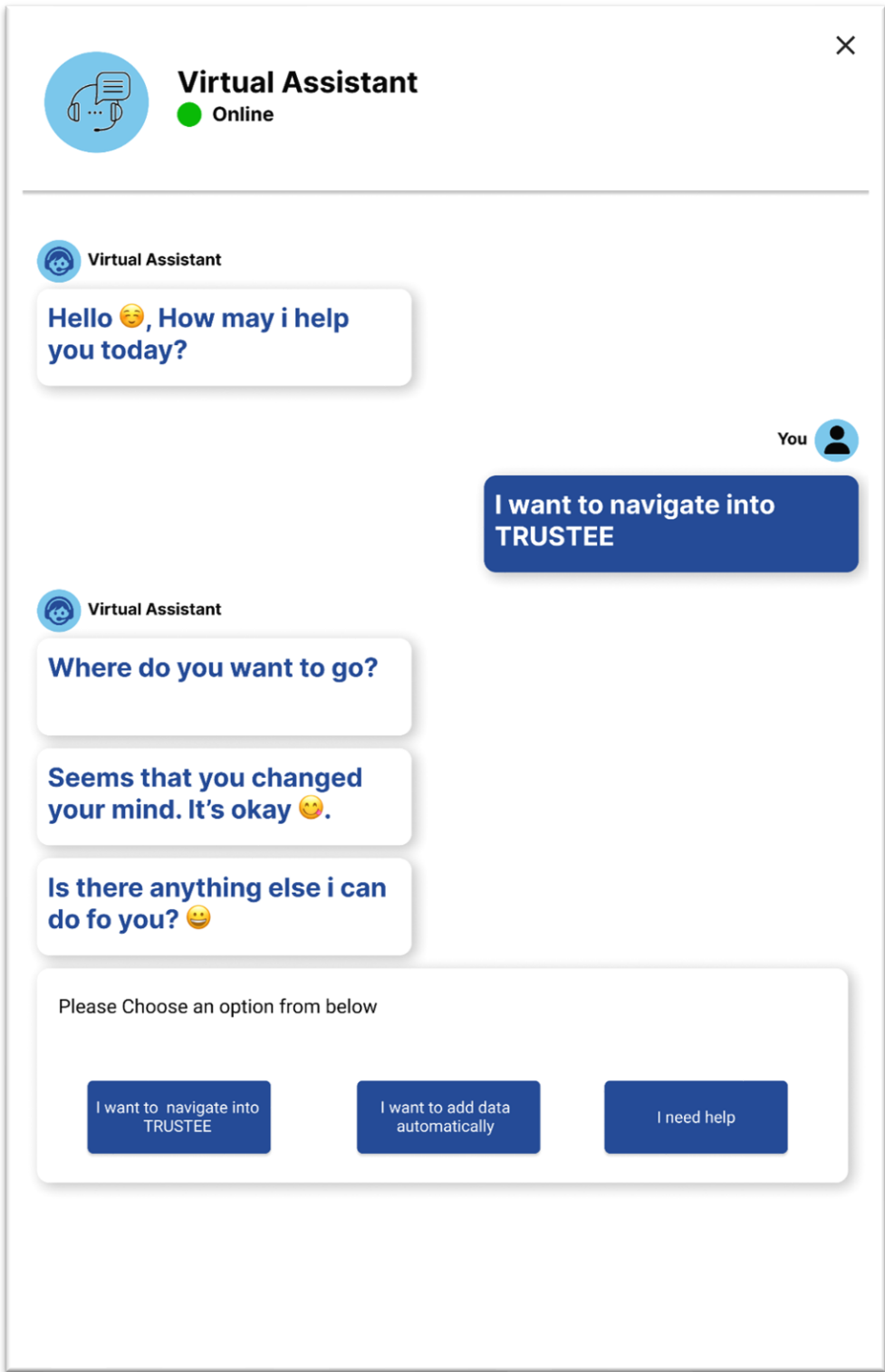


Figure 71: Virtual Assistant – Navigation use case #5

3.5.4. TECHNICAL DETAILS

The Technical Details section provides an in-depth overview of the development, installation, and deployment aspects of the TRUSTEE platform. This section is crucial for technical stakeholders and development teams to understand the tools, technologies, and infrastructure required to successfully implement and deploy the DA ABB. As seen in Table 27, Table 28, and Table 29 the development, installation and deployment aspects of the main UI, the RPA and VA sub components are shown respectively.

3.5.4.1. MAIN UI

Table 27: Main UI technical details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	ANY
Database and databases tools	-
GUI	VueJS
Programming language	JavaScript, HTML, CSS
Development Environment	Chromium
INSTALLATION AND DEPLOYMENT	
Software Requirements	NodeJS, VueJS, Pinia, Vue Router
Hardware Requirements	-
Containerization	-
Communications	HTTP RESTful

3.5.4.2. RPA

Table 28: RPA Technical details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	Windows NT
Database and databases tools	-
GUI	-
Programming language	C++, JavaScript, Shell
Development Environment	Windows 10
INSTALLATION AND DEPLOYMENT	
Software Requirements	RabbitMQ, Windows API, MinGW64, Microsoft UIA
Hardware Requirements	Minimum requirements: 1 core CPU, 60MB RAM, 100MB hard disk space.
Containerization	-
Communications	AMQP

3.5.4.3. VIRTUAL ASSISTANT

Table 29: Virtual Assistant technical details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	ANY
Database and databases tools	-
GUI	VueJS, PrimeVue, Vanilla Javascript
Programming language	JavaScript, HTML, CSS
Development Environment	Chromium
INSTALLATION AND DEPLOYMENT	
Software Requirements	NodeJS, VueJS
Hardware Requirements	-
Containerization	-
Communications	AMQP, HTTP RESTful

3.6. VALIDATION

Since the beginning of the TRUSTEE project, several work has been done into the development of the DA. This work includes the RPA Robot capabilities to interpret the automation process file, decode the automation process and execute the respective actions using the mouse of the user. Additionally, the main UI and the VA have been conceptualized and designed into the high fidelity wireframes presented above. Because of the immaturity of the components, in this section the proposed methodology to conduct the validation is presented. This methodology will be employed in upcoming deliverables, namely D4.2 and D4.3, as the DA components will have reached a suitable level of development.

3.6.1. VALIDATION METHODOLOGY

The validation methodology outlined below provides a framework for the validation of the RPA component and VA. While these components are currently not mature enough for complete validation, this methodology will be employed in D4.2 and D4.3 deliverables as they will have reached a suitable level of development.

- Validation Objectives
 - Define the validation objectives to ensure compliance with functional requirements, performance benchmarks, security standards, and usability criteria.
- Validation Criteria:
 - Establish specific criteria, including accuracy, response time, error rates, scalability, compliance, and user satisfaction, to determine the success of the validation process.
- Test Plans and Execution:
 - Develop comprehensive test plans covering functional, performance, security, and usability aspects. Execute the test cases and document the results, including any issues or discrepancies encountered.
- Integration and User Acceptance Testing:

- Conduct integration testing to evaluate the interaction between the DA and external ABBs. Engage end-users or representative stakeholders for user acceptance testing and gather feedback on performance, usability, and effectiveness.
- Evaluation and Iterative Improvement:
 - Analyze the test results, compare against the validation criteria, and identify areas for improvement. Address any issues, bugs, or performance gaps through iterative development and additional testing.
- Documentation and Review:
 - Prepare documentation of the validation process, including test plans, test cases, results, and modifications. Seek review and approval from relevant stakeholders to ensure alignment with project objectives.

3.6.2. VALIDATING THE WORKFLOW AUTOMATION

An initial validation assessment of the RPA Robot has been conducted to ensure the stability and functionality at the development stage. This assessment involves a simple test automation process acting upon the Microsoft Windows calculator application to visually validate the functioning robot. In Figure 72, the robots internal sequence diagram is shown. The automation starts using a command from the TRUSTEE System developer to the robot to execute the process called “Calculator”. The robot interprets the process and detects the specified element on the screen. Once the detection is completed successfully, the robot moves the mouse over the detected element and performs a mouse click on it. This procedure gets repeated three more times, each time with a different element. After the robot execution, a success message is returned to the TRUSTEE System developer indicating that the robot runs successfully. To verify the accuracy, a quick glance at the calculator's user interface while the robot executes the process is sufficient to determine whether the actions were conducted as anticipated.

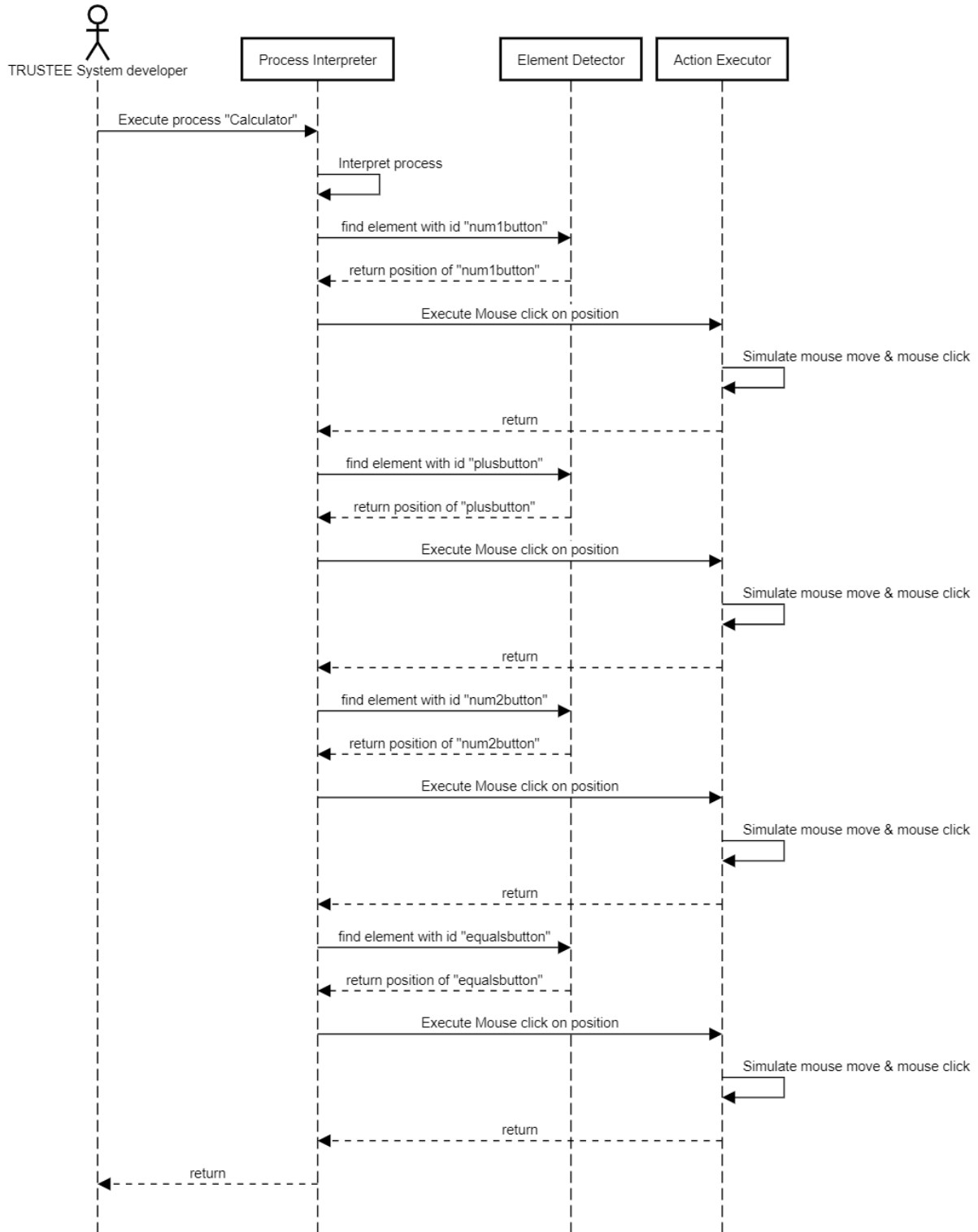


Figure 72: Workflow automation test on Windows Calculator

3.7. NEXT STEPS

Up to this point, the architecture and the development level of the DA ABB has been showcased. In this section, a roadmap that outlines the next steps to propel the development of DA ABB will be presented.

- Add keyboard event simulation functionality to RPA Robot.
This functionality will enable the RPA Robot the users keyboard to type into text entries and perform more complex automations.
- Define the available Questions & Answers of the VA.
Define which will be the available paths into the decision tree of the VA and the appropriate messages to be exchanged throughout the chat.
- Define and construct the VA decision tree.
- Encode the above mentioned paths into the actual decision tree and add it to the Knowledge base of the VA.
- Create the high fidelity mock-ups of the main UI and linking them together to create an interactive experience.
- Progress with the development of the 1st version of the mock UI of the DA
- Create the interactive high fidelity mock-ups of the VA.
- Further implement the Automation Manager.
- Progress with the development of the RPA Automation Manager sub-component.

3.8. CONCLUSIONS

In conclusion, this chapter provided the technical details about the DA ABB and its internal sub-components. First, all the associated user-centred, technical, legal, and socio-ethical requirements were showcased. The overall Architecture of the ABB was presented as well as the Operational principle of the main UI, the RPA, and the VA. Furthermore, an initial version of the GUI of the components was presented and explained using use case scenarios. Alongside with the Technical details and the TRUSTEE UI/UX guidelines and recommendations, some external interfaces and the involved input-output data models were outlined. Finally, a comprehensive validation methodology was proposed to access the internal components of the DA, which will be used as well as revised and enhanced in upcoming version of this deliverable, namely D4.2 and D4.3.

4. PART II: DATA PROTECTION IMPACT ASSESSMENT

4.1. INTRODUCTION

As the digital landscape grows, the integration and significance of data protection cannot be overstated. Central to TRUSTEE's vision is the provision of an efficient and easily comprehensible framework and tool for conducting DPIAs. This necessity arises from the critical role DPIAs play in ensuring the protection of personal data, in line with the European Union's GDPR.

The objective of the DPIA in TRUSTEE is to aid users in determining if a DPIA is mandatory for their data processing activities and, in such cases, guide them through the assessment. The assessment will take into account the description of the processing operations, their necessity and proportionality, the risks involved, and the measures taken to mitigate these risks.

By delineating the importance, process, and execution of DPIAs within TRUSTEE, we aim to empower our stakeholders with an easy-to-use tool and a robust framework. With this, they will be better equipped to safeguard personal data, thereby advancing the cause of data protection in the digital landscape.

4.1.1. NEED AND OBJECTIVES

The TRUSTEE project identifies several crucial needs for the development of its DPIA ABB, with the selection of the Commission Nationale de l'Informatique et des Libertés (CNIL) tool. These needs include creating a user-friendly and easily comprehensible tool, which is paramount for lowering the application barriers for EU stakeholders. The tool is also needed to provide a comprehensive and multidimensional assessment framework, encompassing user-centered, technical, legal, and socio-ethical requirements of conducting a DPIA, thereby ensuring all aspects of data protection are adequately met. Integration with cybersecurity risk assessment frameworks is another essential need, aiming to amalgamate considerations of security and privacy in data protection endeavors. Furthermore, the tool is required to offer operational transparency by delineating its high level and low level architectures, thus providing clear insights into its operational dynamics and procedural structures.

Addressing the outlined needs leads to the pursuit of several pivotal objectives in the TRUSTEE project. The CNIL tool aims to systematically aid users in identifying whether conducting a DPIA is mandatory and guide them meticulously through the subsequent assessment, providing clear, coherent descriptions of processing operations and their underlying purposes. Assessing the necessity and proportionality of processing operations and evaluating the congruence of processing activities with their intended purposes are central objectives to ensure rational and justified processing. The tool also focuses on rigorous risk assessments to evaluate and mitigate risks to the rights and freedoms of data subjects and proposes efficacious measures to address identified risks. Additionally, a prime objective is to facilitate organizations in demonstrating compliance with GDPR, emphasizing safeguarding the rights and legitimate interests of data subjects. The enhancement and refinement of the CNIL tool, aligning with GDPR guidelines and principles, signify the project's commitment to advancing the standards of data protection in the EU digital landscape.

4.1.2. INNOVATIONS

The TRUSTEE's DPIA ABB, leveraging the CNIL tool, brings forward several innovations including a user-centric and easily understandable design aimed at lowering application barriers for EU stakeholders. It integrates a multidimensional assessment framework and cybersecurity risk assessments, ensuring comprehensive evaluations of privacy and security risks. The tool provides enhanced operational transparency through detailed architecture delineations and facilitates rigorous necessity and proportionality assessments of processing operations, enhancing reliability and effectiveness. Additionally, it enables organizations to effectively demonstrate compliance with GDPR, while its continual refinement and expansion align it with evolving data protection standards and principles.

4.2. BACKGROUND

4.2.1. LEGAL AND REGULATORY FRAMEWORKS

The Regulation 2016/679 (GDPR) [32], along with its Article 35, establishes the concept of a DPIA. It mandates that data controllers who engage in data processing which likely results in a high risk to the rights and freedoms of data subjects must conduct a DPIA. A similar obligation is also outlined in Directive 2016/680 [33].

It is important to stress that such a controller is required to carry out an assessment prior to the processing, to assess if the risks associated with planned data processing activities are appropriately addressed even before their start.

A DPIA is “a process designed to describe the processing, assess its necessity and proportionality and help manage the risks to the rights and freedoms of natural persons resulting from the processing of personal data by assessing them and determining the measures to address them.” [34]. According to the Article 35(3) of the GDPR [32], a DPIA is in particular required in the situations when the controller conducts (a) a systematic and extensive evaluation of personal aspects relating to individuals, based on automated decision making, which includes profiling, and on which decisions are based which produce legal effects which concern the individual, or in a similar manner significantly affect individual; (b) processing of personal data representing special categories of data or data relating to criminal convictions and offences on a large scale; (c) a systematic monitoring of an area, which is publicly accessible, on a large scale.

Nevertheless, this catalogue is not exhaustive. The GDPR sets out that a supervisory authority at the national level may establish a list of processing operations, which require a DPIA, article 35 GDPR [35]. Such a list should be made publicly available.

The Article 29 Working Party (“WP29”) [35] provided its further guidelines concerning the necessity of the DPIA, in what situations it should be considered, and what should include and address.

Following the guidelines, when a data controller is planning on processing personal data, has to check whether this processing activity falls in the categories listed by the WP29, i.e. categories of processing which may have an impact on the rights and freedoms of the data subject:

- a) **evaluation or scoring** – this includes including profiling and predicting. An example here could be building marketing or behavioural profiles on the basis of usage and navigation of the data subjects on the company’s website,
- b) **automated-decision making with legal or similar significant effect** – this includes, inter alia, processing which may lead to exclusion or discrimination against the data subjects,

- c) **systematic monitoring** – this processing includes the type defined in Article 35(3)(c), as well as observing, monitoring or controlling individuals, especially in the context where they do not know who is collecting the data and how they will be used,
- d) **sensitive data or data of highly personal nature** – this includes special categories of data (defined in Article 9 GDPR) and data concerning criminal convictions or offences (defined in Article 10 GDPR), as well as processing the data which may be considered as increasing the risk to the rights and freedoms of the data subjects. Such data are, e.g., data linked to the household or private activities, related to the exercise of fundamental rights or data whose infringement could seriously influence the individual's life (e.g., financial data used for fraud),
- e) **data processed on a large scale** – the term “large scale” is not defined in the GDPR, however, in the assessment process it is recommended to take into account: the number of individuals concerned; the volume or range of data processed; the permanence and duration of the processing activity and the geographical scope of the processing activity,
- f) **matching or combining datasets** originating from data processing operations which were performed for different purposes or different data controllers in a manner which would go beyond the expectations of the individual,
- g) **data concerning vulnerable data subjects** – the group of vulnerable data subjects may include, e.g., children, employees, elderly, patients, asylum seekers and any case in which an imbalance between the position of a data controller and a data subject can be observed,
- h) **innovative use or applying new technological or organizational solutions** – where a new technology should be assessed according to the state of technical knowledge. The use of AI for data processing, Internet of Things (IoT) applications, application of novel technologies increasing the data computing capabilities may trigger a need to conduct a DPIA,
- i) **when the processing in itself “prevents data subjects from exercising a right or using a service or a contract”** – This includes processing operations that aims at allowing, modifying or refusing data subjects' access to a service or entry into a contract [32].

In cases where it is not clear whether a DPIA is required, the WP29 recommends carrying one out nonetheless, since a DPIA is a useful tool to help controllers comply with data protection law [34], p.8, especially negate risks, and demonstrate compliance with the GDPR [36]. A DPIA is a living instrument, it has to be established before the processing is initiated but it should be reviewed and reassessed on a regular basis as the processing evolves, pursuant to the requirements of privacy-by-design and privacy-by-default set by the GDPR [32], Article 25.

Conducting a DPIA includes several steps to both assess the risks attached to the processing and the documentation of compliance. CNIL, which is taken into account as the basis for the DPIA ABB development in TRUSTEE, designed by the French data protection authority, encompasses all of them. However, it only deals with personal data and does not include other type of data that may require different protection (such as intellectual property protection) [37].

CNIL provided the method on how to conduct a DPIA with a four-tier assessment: (i) determining the context of the processing, (ii) fundamental principles, (iii) risks attached to the processing, and (iv) validation of the DPIA.

1. The first step is to describe the context of the processing (i):
 - a. Overview the processing,
 - b. Detailed description of the processing: data, processes and supporting assets
2. Once the description of the foreseen processing is set, the controller needs to provide the information how fundamental principles established in the GDPR will be satisfied (ii) and assess the necessity and proportionality of the processing regarding the purposes identified in the description:
 - a. Details concerning necessity and proportionality of processing
 - b. Information about applied controls to protect and satisfy the rights of the data subjects

3. When both the description and the necessity and proportionality have been determined, then the risk assessment begins (iii).
 - a. Identification of the risks to the rights and freedoms of data subjects.
 - b. Each risk is associated with a level (low, intermediate, high).
 - c. The risk assessment allows the controller to determine what measures would be appropriate to implement in order to mitigate the risk that has been identified. For instance, to prevent data leakage, an organisational measure could be access restriction to the data with a habilitation policy strictly determined and enforced. Additionally, to ensure confidentiality of the data a technical measure would be encryption. In the case of TRUSTEE, this would mean the use of homomorphic encryption to reinforce privacy.
4. When the DPIA is completed, the controller – or the Data Protection Officer if the entity has one – must review and validate the DPIA before starting the processing (iv).

In the case of TRUSTEE and the creation of a DPIA tool, the CNIL tool is used as a model of reference while acknowledging that only personal data are included in the latter. For non-personal data, the impact created by the processing and/or data sharing would be of economic or social value [38]. Therefore, it is recommended to take into account sensitive data other than personal data to protect from intellectual property breaches and/or theft and other vulnerabilities that may lead to judicial proceedings, loss of reputation with economic repercussions, and business failures. This will be further explored in the lifecycle of the TRUSTEE project and more information will be provided in future versions of this live document.

4.2.2. DPIA PROCESSING ACTIVITIES

A DPIA is a structured process that serves multiple purposes. It aims to describe the processing activities, evaluate the necessity and proportionality of those activities, and manage the risks to the rights and freedoms of individuals associated with the processing. By assessing the risks involved and determining appropriate measures to address them, DPIAs contribute to accountability in data processing. They not only assist data controllers (depending on the use case scenario – Data Providers or Consumers) in meeting the requirements of the GDPR but also demonstrate that adequate measures have been implemented to ensure compliance with the Regulation. In essence, a DPIA is a tool for establishing and showcasing compliance. Failure to comply with DPIA requirements can result in penalties imposed by the relevant supervisory authority. The term "Privacy Impact Assessment" (PIA) is often used interchangeably with DPIA, referring to the same concept.

Initiating the DPIA as early as possible during the design phase of the processing operation is highly recommended, even if some processing activities are yet to be determined. This approach is beneficial for implementing data protection by design principles, as the DPIA results will identify areas requiring privacy consideration and implementation. By regularly updating the DPIA throughout the project lifecycle, data protection and privacy aspects are consistently addressed, encouraging solutions that facilitate compliance. It may be necessary to revisit specific assessment steps as the development process advances, as the selection of certain technical or organizational measures can influence the severity or likelihood of the risks associated with the processing.

TRUSTEE's DPIA adopts a methodology which incorporates a Privacy and security Assessment for Cross-discipline Trust and Sovereignty (PACTS) framework. Despite not being universally recognized as a standard, the proposed PACTS methodology remains robust and viable for the task at hand, meticulously aligning with the stringent criteria delineated by the Article 29 Data Protection Working Party (WP29) guidelines [32], [33]. This alignment underscores the framework's credibility and applicability in addressing and conforming to the outlined data protection mandates and principles.



Figure 73: PACTS iterative process for conducting out a DPIA

The methodology employed in the PACTS framework (Figure 73) ensures a systematic and reliable use of personal data during processing through an iterative approach. The steps involved in this methodology are as follows:

- i. Broadly describe the objectives and processing activities of each TRUSTEE ABB and TRUSTEE persona, summarizing the rationale behind the need for a DPIA.
- ii. Describe the nature of data processing carried out during each use case, including data sources, collection, use, storage, sharing, and deletion practices.
- iii. Identify relevant stakeholders and consult with them throughout the DPIA process, leveraging their expertise in information security.
- iv. Engage relevant data controller parties to assess the necessity and proportionality of their data processing activities, ensuring compliance with GDPR provisions (Article 5 and Article 6). Consider data protection compliance measures as recommended by the Article 29 guidelines.
- v. Assess the potential impact on individuals and evaluate any physical, emotional, or material harm that may arise from the intended processing.
- vi. Evaluate options for reducing the identified risks.
- vii. Conclude the DPIA, taking into account the findings and recommendations from the previous steps.

The proposed methodology in PACTS enhances the criteria outlined by WP29 by incorporating the following key elements:

- **Systematic Description of Processing:** The methodology provides a structured and comprehensive description of the processing activities, as mandated by Article 35(7)(a) of the GDPR.
- **Necessity and Proportionality Assessment:** The methodology assesses the necessity and proportionality of the processing, ensuring compliance with Article 35(7)(b) of the GDPR.
- **Risk Management for Data Subjects' Rights and Freedoms:** The methodology focuses on effectively managing risks to the rights and freedoms of data subjects, in accordance with Article 35(7)(c) and (d) of the GDPR.
- **Involvement of Interested Parties:** The methodology encourages the participation of relevant stakeholders, such as Data Protection Officers (DPOs) as outlined in Article 35(2) of the GDPR, as well as the inclusion of data subjects or their representatives, as specified in Article 35(9) of the GDPR.

The performance of the DPIA is primarily the responsibility of the data controller or joint controllers, as applicable in the case of PACTS, who collectively determine the purpose and means of processing in accordance with Article 26 of the GDPR. The main objective of the DPIA is to ensure the implementation of privacy protection principles within the project, allowing data subjects to maintain control over their personal data. This process enables data controllers to demonstrate their compliance approach and the selected controls, emphasizing the concept of accountability outlined in Article 5(2) and Article 25 of the GDPR. Similarly, application providers can showcase their adherence to Privacy by Design, as described in Article 25 of the GDPR.

Upon completing the DPIA, it is essential to integrate the obtained outcomes back into the project plan. This involves identifying actionable points and assigning responsibilities for their implementation. By employing standard project management processes, the execution of these action points can be ensured while simultaneously monitoring the ongoing effectiveness of the DPIA. In cases where there are significant changes to the processing's nature, scope, context, or purposes, it is necessary to repeat the DPIA process.

DPIA ABB enables clear communication of processing intents, risks, and mitigation strategies to data subjects while ensuring a structured approach to documenting compliance, addressing risks, and implementing responsible data processing practices. A tight couple between DPIA ABB and STM ABB as well as, TAI-SDF ABB ensures that the outcomes of the DPIA will be utilised as part of the overall cybersecurity risk assessment framework of TRUSTEE.

4.3. REQUIREMENTS

In this chapter, the compliance of the DPIA ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding with the aforementioned requirements addressed during the development and by the implementation of the DPIA and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements have been considered. As the implementation of the DPIA progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

User-centred requirements were identified through the UX Research that was conducted under T2.3. These requirements were reported in D2.1 [1], while they were also utilized in order to extract technical, functional and non-functional requirements, that were also outlined in D2.1. Table 30 below presents the interplay between user-centred requirements and technical requirements, that is the functional and non-functional requirements that address each user-centred requirement. The user-centred requirement ID is used to reference the respective requirement reported in D2.1 [1] while the requirement's description is also provided alongside whether the DPIA is currently fully or partially compliant with the user-centred requirement, i.e., indicating whether technical requirements have been identified in order to address the respective user-centred requirement. The ID of the technical requirement, as reported in D2.1 [1], that is addressing each user-centred requirement is also provided in the last column of the table. Additionally, it is possible than more than one technical requirement addresses a single user-centred requirement.

Table 30: DPIA – User-centred requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
USR-001	TRUSTEE must enable collaboration among domains and enable access and use of data for secondary use from multiple domains	Yes	Req-DPIA-FUNC-1 Req-DPIA-FUNC-2
USR-003	Users must provide access to their data in a secure and privacy aware way without sending their raw data to others	Yes	Req-DPIA-FUNC-1
USR-005	The data provided must be used with respect to privacy regulations, legal frameworks, and ethical considerations and without the risk of data loss	Yes	Req-DPIA-FUNC-1 Req-DPIA-FUNC-2 Req-DPIA-FUNC-3

USR-006	The user must always be aware of how their data are used and they must be able to terminate the procedure if any violation of their interests may happen	Yes	Req-DPIA-FUNC-1 Req-DPIA-FUNC-4
USR-007	Users must always keep control over the data to which they provide access	Yes	Req-DPIA-FUNC-2 Req-DPIA-FUNC-4
USR-009	TRUSTEE must protect education data from misuse that aims in profiling students	Yes	Req-DPIA-FUNC-2
USR-010	TRUSTEE must protect students from being automatically subjected to companies' laws and conditions	Yes	Req-DPIA-FUNC-2 Req-DPIA-FUNC-4
USR-029	TRUSTEE should offer collaboration across multi-disciplinary data domains	Yes	Req-DPIA-FUNC-1
USR-033	TRUSTEE should enable computations and analysis over the accessed data	Yes	Req-DPIA-FUNC-1
USR-036	TRUSTEE should allow efficient data querying	Yes	Req-DPIA-FUNC-2
USR-038	TRUSTEE should enable transfer of learning without IPR and data privacy issues	Yes	Req-DPIA-FUNC-2
USR-040	TRUSTEE should provide protection of personal data	Yes	Req-DPIA-FUNC-2 Req-DPIA-FUNC-4
USR-041	TRUSTEE should allow the process of data without IPR and privacy issues	Yes	Req-DPIA-FUNC-2
USR-042	TRUSTEE should be compliant with all security requirements and legislations regarding data manipulation and querying	Yes	Req-DPIA-FUNC-2 Req-DPIA-FUNC-3
USR-045	TRUSTEE should securely share commercial and private data	Yes	Req-DPIA-FUNC-2 Req-DPIA-FUNC-4
USR-046	TRUSTEE should eliminate the correlation of data with personal data	Yes	Req-DPIA-FUNC-1
USR-054	TRUSTEE should have easy access and permissions	Yes	Req-DPIA-FUNC-2
USR-057	TRUSTEE should be an efficient and highly secure platform	Yes	Req-DPIA-FUNC-1
USR-100	The user could be able to decide the level of access they are willing to provide for their data	Yes	Req-DPIA-FUNC-2
USR-107	Through TRUSTEE data providers could be able to avoid pseudonymization and be able to share data	Yes	Req-DPIA-FUNC-4

Table 31 presents the preliminary results of the investigation regarding the compliance of the DPIA with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether DPIA complies with the

requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 31: DPIA – Functional and non-functional technical requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
Req-DPIA-FUNC-1	The tool MUST allow the user to carry out the risk assessment based on CNIL methodology.	Yes	The tool CNIL PIA complies with the CNIL methodology.
Req-DPIA-FUNC-2	The DPIA tool MUST assess the privacy impact of the data communication(s) between the TRUSTEE components.	Partially	Depends on the maturity of the development of the ABBs and the interfaces between them.
Req-DPIA-FUNC-3	The DPIA tool MUST be provided from an EU member public authority and abide to the legal knowledge base of the EU.	Yes	CNIL (Commission Nationale de l'Informatique et des Libertés) is a member public authority of the EU.
Req-DPIA-FUNC-4	The tool MUST include the module allowing to describe the purpose of processing the data, and, where applicable to the user, legitimate interest in processing.	Partially	Scenario specific. Depends on the maturity and development of pilot specific use cases, data availability.

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

4.4. ARCHITECTURE

This section presents the tools and processes used to perform the DPIA, including the high level and low level architecture of the DPIA tool.

4.4.1. HIGH LEVEL

In the TRUSTEE project, the DPIA ABB goes beyond functioning as a standalone tool and integrates with other ABBs of the TRUSTEE Platform to enhance its capabilities and facilitate seamless communication and collaboration. The DPIA ABB interacts with several other ABBs, enabling the exchange of information and leveraging shared functionalities. Currently DPIA ABB communicates with the ATR and the STM ABBs sharing the results of the DPIAs. It will also interact with the AM ABB for the users' authentication and lastly, the DPIA tool will be accessed through the DA ABB for a seamless user experience. External interfaces and communication between building blocks is an ongoing process stemming from TRUSTEE overall architecture definition and will be detailed in forthcoming deliverables, namely D4.2 and D4.3, as well as in D2.2.

The high-level architecture of the DPIA ABB is depicted in, showcasing the primary internal components and their connections with other ABBs through external interfaces. By illustrating the high level architecture, Figure 74 provides an overview of how the internal components of the DPIA ABB interact with external interfaces, promoting effective communication and collaboration within the TRUSTEE project.

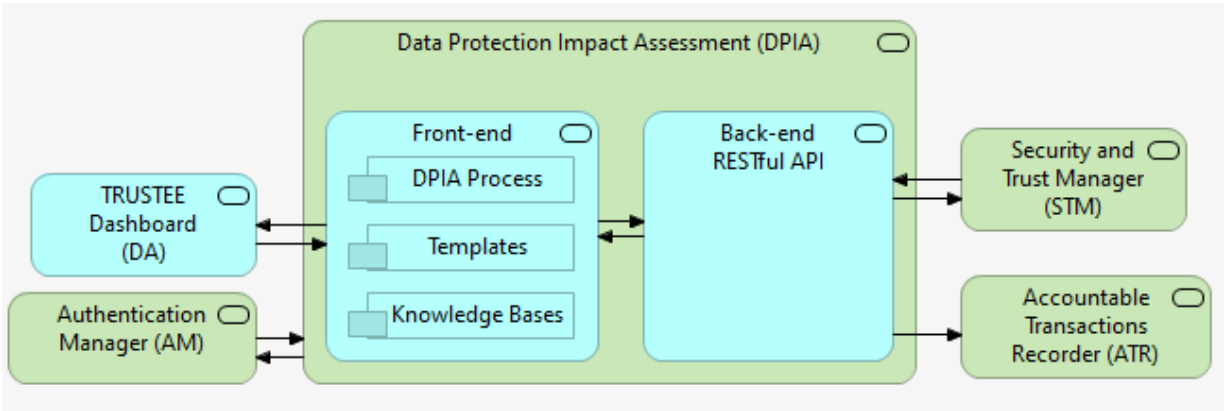


Figure 74: High Level Architecture of the DPIA ABB (D2.1 [1]- Figure 108)

4.4.2. LOW LEVEL

The DPIA ABB consists of the following main components: a) the DPIA Interface (frontend), b) the DPIA Server (backend), and c) a database. Figure 75 illustrates the low level architecture of the DPIA ABB, providing a detailed view of its internal structure.

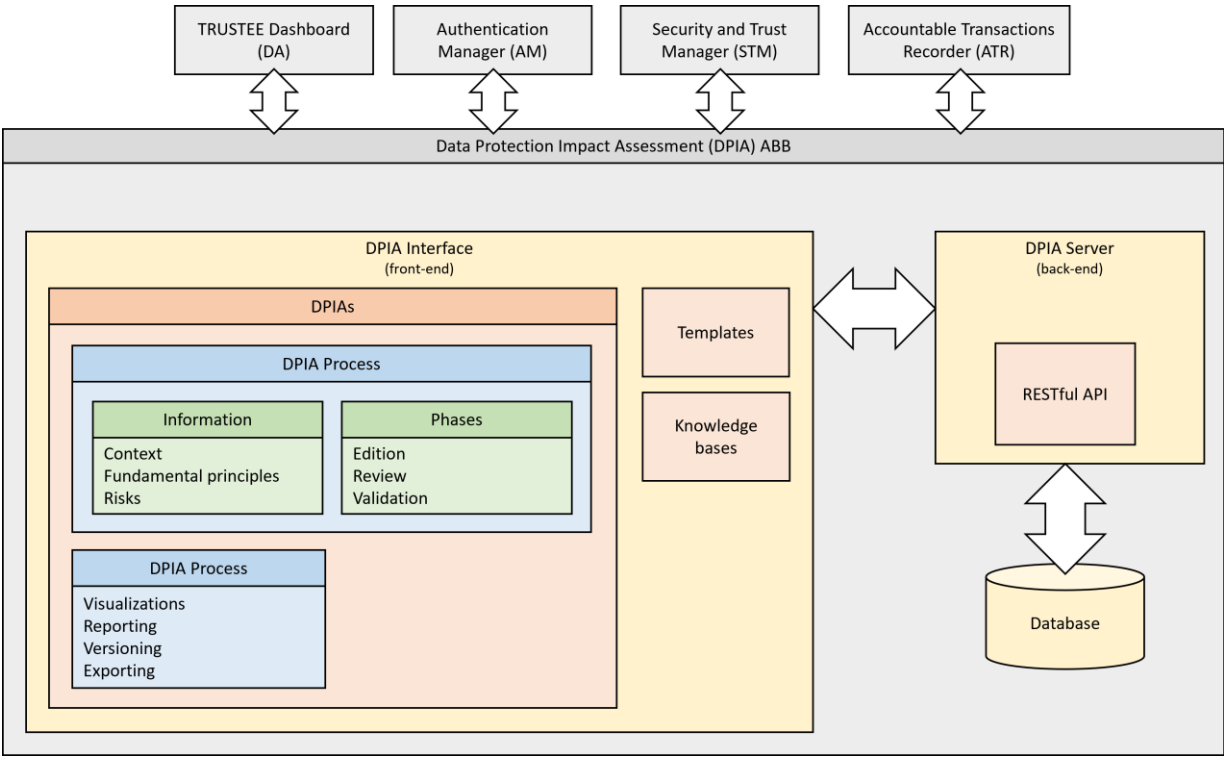


Figure 75: Low Level Architecture of the DPIA ABB (D2.1 [1]- Figure 109)

DPIA Interface

The DPIA Interface is the user interface that gives users access to all the functionalities needed to perform a data protection impact assessment including the information and phases of the DPIA process and additional useful functions as well as the templates that can act as a starting point for a DPIA and the legal and technical knowledge bases. The users will be redirected to the DPIA Interface through the DA, for a seamless experience within the TRUSTEE ecosystem. The DPIA interface contains the DPIA module, Templates as well as a Knowledge Base modules.

- **DPIAs module**

The DPIA process requires various sections of information to be filled in by the users. The first section is called “Context” where users should identify and state the object of the DPIA study, while at the same time define and describe the scope of the processing in detail. The second section that should be populated is named “Fundamental principles”. In this section, users should demonstrate that they have implemented the necessary means to enable the persons concerned to exercise their rights. The last section, named “Risks”, is the section where users should assess the privacy risks, considering existing or planned controls. The DPIA process has three different phases. The first phase is the “Edition” phase where users populate the sections described above. Once a section or sub-section is completed, users can request a review for this specific section/sub-section. This phase is called “Review” phase. Lastly, there is the “Validation” phase. Once the reviewer has reviewed a section in the previous phase, they can then ask for overall DPIA validation. The final step of the DPIA process is the DPIA validation where the validator prepares and formalises the validation of the DPIA. The DPIA process also includes various functions like visualizations, reporting, versioning and exporting.

- **Templates**

The “Templates” module enables users create and use their own templates with pre-filled information or additional questions/measures. They can create a different template to be used in different use cases and processes.

Knowledge Base

The “Knowledge Base” module gives the ability to users to create a knowledge base that can then be consulted during the DPIA process. This knowledge base can be both legal and technical and can include elements like definitions, principles and methodologies. These elements can be linked to specific sections/sub-sections of the DPIA process which means that they will be visible only when users populate those sections.

DPIA Server

The DPIA Server (backend) works as a RESTful API and includes all the functionalities of the DPIA ABB. It provides various endpoints which can be requested by the DPIA Interface and execute all the required actions.

Database

The “Database” module contains results and parameters of ongoing or finished DPIAs. This module is connected to the DPIA Server back-end to retrieve and deposit data.

4.5. IMPLEMENTATION

4.5.1. FUNCTIONAL DESCRIPTION

The DPIA ABB encompasses several functionalities, including the following:

- DPIA Creation
- DPIA Edition

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- DPIA Review
- DPIA Validation
- DPIA Importing/Exporting
- DPIA Visualisation & Reporting
- DPIA Templates Creation
- DPIA Knowledge Base Creation

The DPIA ABB methodically unfolds the intricate process of conducting a DPIA. It utilizes a user-friendly interface that enables users to manage their assessments effectively. This platform is vital for individuals or organizations that are keen to implement robust data protection mechanisms.

- **Initiating the Process** - To kick-start a DPIA process, users are required to input various details including a title, and the names of the editor, reviewer, and validator. Additionally, they can select a DPIA template and designate the DPIA category. Refer to the Figure 76 for a visual representation of the process initiation.
- **Editing and Evaluating Sections** - Post-initiation, users can systematically edit each section utilizing the supplied forms. The first section, "Context," mandates users to pinpoint the object of the PIA study and elaborate on the processing scope meticulously. Following this, the "Fundamental Principles" section concentrates on illustrating the incorporation of mechanisms that empower individuals to exercise their rights effectively. The concluding section, "Risks," engages users in an evaluative process of the privacy risks, keeping in view the existing or forthcoming controls. It includes identifying controls that bolster data security and evaluating privacy risks in specific areas such as illegitimate data access, unwanted data modification, and data disappearance.
- **Risk Assessment and Review Process** - In this phase, users thoroughly assess privacy risks, specifying potential impacts, main threats, risk sources, and selecting the anticipated controls to mitigate the risks. After populating all sections and estimating the risk severity and likelihood with valid justifications, users can forward the document for evaluation. Reviewers, at this juncture, have multiple options based on the section under review; these include "To Correct," "Acceptable," or "Improvable," allowing them to make precise evaluations. The selections lead to various subsequent actions, including corrections, validations, and the creation of action plans, depending on the evaluation outcome.
- **Visualizations and Validations** - The DPIA ABB provides users with three distinctive visualizations: "Risks Overview," "Risk Mapping," and "Action Plan." These visualizations are pivotal in enhancing the understanding of risks, the implications of controls, and are instrumental in strategic planning for risk mitigation and compliance enhancement. Each visualization serves a unique purpose, offering insights, comparisons, and planning for implementing additional controls identified during the DPIA process. Upon concluding the review of all sections, the reviewer solicits overall DPIA validation, marking the transition to the final phase of the DPIA process.
- **Finalization and Additional Features** - To finalize a DPIA, the validator inputs the Data Protection Officer's (DPO's) opinion and the opinions of concerned individuals and then either approves or rejects the DPIA. Users have the liberty to export/import DPIAs in JSON format and to download reports in various layouts. Moreover, the platform allows users to fabricate new templates, pre-fill questions for future DPIAs, and concoct knowledge bases by specifying titles and the names of authors and contributors. This feature is pivotal for users aiming to create comprehensive and detailed knowledge bases.

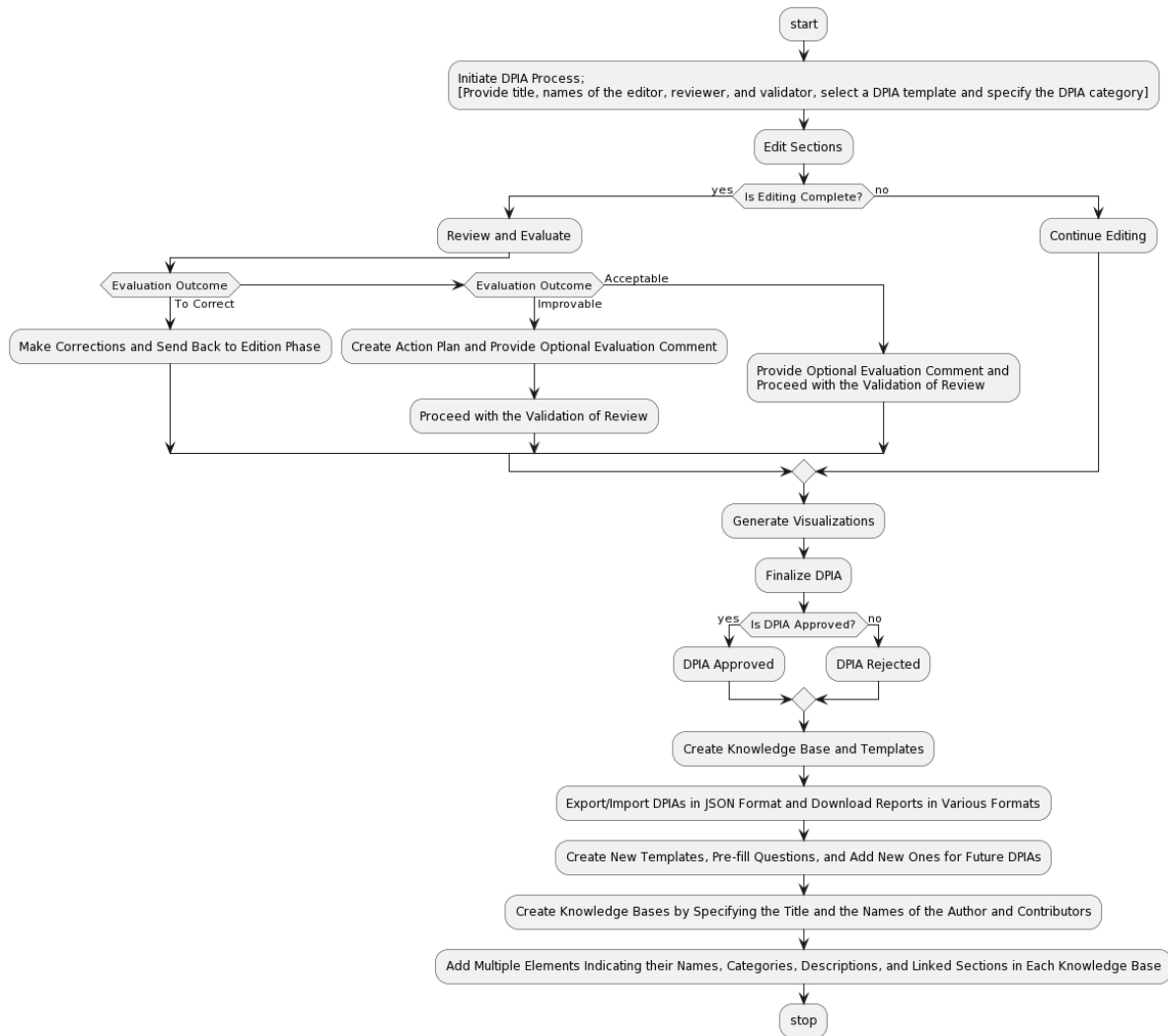


Figure 76 DPIA Workflow

4.5.1. OPERATIONAL DESCRIPTION

This section provides an overview of the internal operation steps of the DPIA ABB. The identification of users is a crucial first step in the functioning of the DPIA ABB. Based on the basic use case interaction flows defined in D2.1 [1], two user personas have the responsibility of conducting a data protection impact assessment using the DPIA ABB. These personas are the Data Providers and Consumers. The Data Provider Persona aims to register their data as part of the Distributed Data Sources of TRUSTEE. As part of the registration process, they are required to conduct a DPIA using the DPIA ABB. The DPIA ABB serves as a tool for assessing and managing the potential risks and impacts on data privacy associated with their data contributions. The Consumer Persona, which is the most significant, must perform DPIAs after specifying what kind of computations they intend to perform on the data to ensure GDPR compliance (Data Consumer). By utilizing the DPIA ABB, these users ensure compliance with data protection regulations and promote responsible data handling within the context of their involvement with TRUSTEE.

The Developer Persona, on the other hand, seeks to utilize the functionalities offered by TRUSTEE as stand-alone functionalities. As they engage with TRUSTEE and its services, developers can perform DPIAs using the DPIA ABB if needed. This assessment allows them to evaluate the potential risks and privacy implications associated with their development activities within the TRUSTEE ecosystem.

The sequence diagram in Figure 77 depicts the sequence of actions/operations related to the implementation of a DPIA and the order in which they occur. It visualizes the dynamic behaviour of the DPIA ABB and the relationships between the DPIA ABB and the broader TRUSTEE infrastructure. The DPIA process encompasses several key activities, including access to the DPIA ABB through the DA, authentication via the AM, creation of a new DPIA, edition of an existing DPIA, review of a completed DPIA, overall validation of the DPIA, and its final approval or rejection. Additionally, the diagram illustrates the exchange of DPIA status information between the DPIA ABB and the DA, as well as the recording of the DPIA in the ATR.

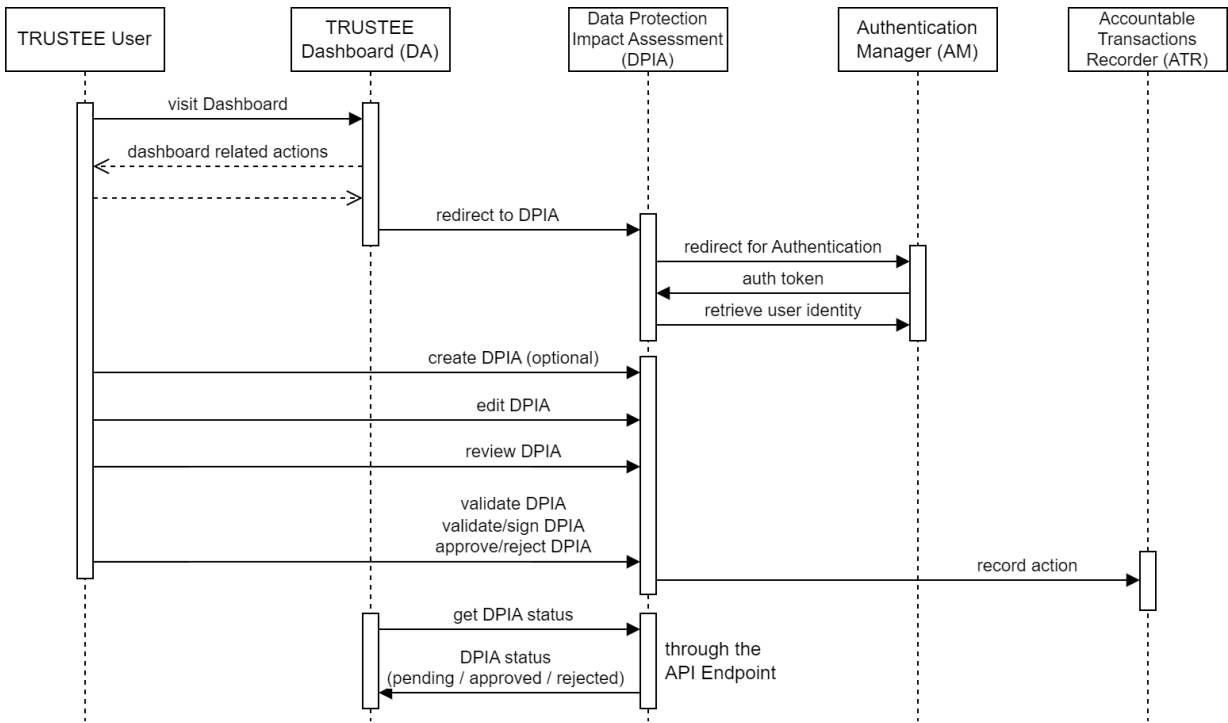


Figure 77: Sequence Diagram of the DPIA ABB including external ABBs (D2.1 [1]– Figure 110)

4.5.2. TECHNICAL DESCRIPTION

4.5.2.1. IMPLEMENTATION DETAILS

The DPIA ABB utilizes the CNIL PIA, an open-source privacy impact assessment software, serving as a foundational tool for structuring assessments. Given its publication under a free license, it is adaptable and can be modified to suit the requirements of any organization or project. It has been meticulously customized to cater to the nuanced needs of both the DPIA ABB and the broader TRUSTEE project.

In its implementation, the DPIA ABB is structured in a client-server architecture. The DPIA Interface, acting as the client or front-end, is meticulously developed utilizing the Angular web application framework. Conversely, the DPIA Server forms the server or back-end, operating through the Ruby on Rails web application framework, offering a seamless interaction and data exchange with the client via a RESTful API.

The data pertinent to DPIA ABB is securely harbored in PostgreSQL, a robust and open-source relational database system. This server, pivotal in processing client's requests, orchestrates necessary operations, and interfaces with the database to store or retrieve data. It unveils a spectrum of endpoints through the API, facilitating other TRUSTEE components to make requests and receive responses seamlessly.

This architecture enshrines a separation of concerns, allowing the DPIA Interface to concentrate on presentation and user interaction, while the DPIA Server manages business logic, data processing, and storage, thereby ensuring a harmonious and efficient workflow between the API and the connected database.

The client-server application's RESTful API streamlines a multitude of tasks including routing, request processing, and response rendering, paving the way for an efficient and seamless interaction between the DPIA Server and Interface.

- **DPIA Server Functionalities:**

The DPIA Server embeds multifarious functionalities. Initially, it defines resources exposed by the API, outlining accessible data entities. Subsequently, routes for API endpoints are configured, associating HTTP verbs—GET, POST, PATCH, PUT, DELETE—with specific controller actions for each resource. These actions, aligning with standard CRUD operations, handle API requests, process the data, and return apt responses. The predominant response format desired is JSON, given the server's primary functioning as an API.

Models within the DPIA Server represent data and integrate logic, serving as liaisons between the application and database tables. They abstract the intricacies of the data source, ensuring data integrity, expressing model relationships, and facilitating data interaction. These models thus establish a coherent representation of data, enabling consistent and efficient interaction and management throughout its lifecycle.

DPIA Interface Components:

The DPIA Interface is constructed using diverse components, services, and models. Components encapsulate user interface parts and their behavior, defined structurally by HTML templates, logically by TypeScript files, and stylistically by CSS or SCSS files. Services facilitate the sharing of data and business logic across components, managing tasks like API data fetching, calculations, and authentication. They align with models to fetch or update data from the DPIA Server's RESTful API, ensuring structured and consistent data interaction.

User Interface and Routes:

The user interface of the DPIA ABB affords access to various main routes, each tethered to specific components such as HomeComponent, EntriesComponent, PiaComponent, StructureComponent, and BaseComponent. Each route corresponds to a unique URL or endpoint, guiding users through the various components of the DPIA Interface, thereby enabling them to conduct DPIAs effectively.

Table 5 elucidates the components of the DPIA Interface and their corresponding routes, offering a panoramic view of their functionality. These refined components, services, and templates have been meticulously modified to incorporate

the envisaged features, reinforcing the effectiveness and user-friendliness of the DPIA ABB in conducting comprehensive data protection impact assessments.

Table 32: DPIA Interface's Components and Routes

HomeComponent	
URL(s)	"/" or ""
EntriesComponent	
URL(s)	• /entries • /entries/archive • /entries/structure • /entries/knowledgebase
PiaComponent	
URL(s)	• pia/{pia_id} • pia/{pia_id}/section/{section_id}/item/{item_id}
StructureComponent	
URL(s)	• structures/{structure_id} • structures/{structure_id}/section/{section_id}/item/{item_id}
BaseComponent	
URL(s)	knowledge_bases/{id}

4.5.2.1. DATA MODELS

User-provided information is stored in a PostgreSQL database using various data models converted into tables. The DPIA ABB's database includes the following tables: a) answers, b) attachments, c) comments, d) evaluations, e) knowledge_bases, f) knowledges, g) measures, h) structures, i) pias and j) revisions. Each table serves a specific purpose in the DPIA process.

The database's intricate architecture accommodates various tables, each serving a distinct purpose in storing diverse elements related to DPIAs.

The "answers" table archives responses to the myriad of questions presented during the DPIA, providing an insightful reflection of users' interactions and inputs. For user-supplied attachments, the "attachments" table serves as a reservoir, maintaining all files appended by users throughout their DPIA journey.

User-extended comments addressing specific questions find their home in the "comments" table, capturing the nuances of user insights and evaluations. The "evaluations" table enshrines reviewers' responses pertinent to a specific DPIA, serving as a testament to the assessment's comprehensive review process.

Legal and technical discernments curated in knowledge databases by users reside in the "knowledge_bases" table. It's here that corresponding elements align within the "knowledges" table, establishing a symbiotic relationship for coherent interaction and retrieval.

User-conceived measures are meticulously preserved in the "measures" table, reflecting the diverse approaches and methodologies employed throughout the DPIA process. The table of paramount significance is the "pias" table, embracing all-encompassing information related to DPIA, acting as a cornerstone for the entire database architecture.

Meanwhile, the evolution and variations of a DPIA are chronicled in the "revision" table, reflecting the dynamism and adaptability of the assessments.

Every table, with the exception of “pias,” is equipped with dedicated columns to record timestamps of both creation and last modification, ensuring meticulous tracking and recording of each entry’s inception and subsequent modifications. A common link, the column “pia_id,” interconnects tables “measures,” “evaluations,” “revisions,” “answers,” “attachments,” and “comments” with the central “pias” table. This linkage serves as the linchpin for associating and retrieving coherent information related to distinct DPIAs.

Every entry within the “pias” table inherently possesses a “structure_id” column, establishing a seamless connection with the “structures” table and fortifying the relationship between each DPIA entry and its corresponding structure or template.

Lastly, the interconnectedness between the “knowledge_bases” and “knowledges” tables is fortified through the “knowledge_base_id” column, allowing for structured organization and retrieval of knowledge elements within delineated knowledge bases.

Table 33 provides a comprehensive list of the database tables utilized in the DPIA ABB, along with their corresponding columns and the associated data types while Figure 78 provides the precise database schema of DPIA ABB, illustrating the relationships between the tables. The schema captures the structure and connections between the tables, enabling a comprehensive understanding of the database organization.

Table 33: Database Tables of the DPIA ABB

TABLE: answers	
id (PK)	integer
reference_to	character varying
data	jsonb
pia_id	integer
created_at	timestamp
updated_at	timestamp
TABLE: attachments	
id (PK)	integer
attached_file	character varying
pia_signed	boolean
pia_id	integer
created_at	timestamp
updated_at	timestamp
comment	text
TABLE: comments	
id (PK)	integer
description	text

reference_to	character varying
for_measure	boolean
pia_id	integer
created_at	timestamp
updated_at	timestamp
TABLE: evaluations	
id (PK)	integer
status	integer
reference_to	character varying
action_plan_comment	text
evaluation_comment	text
evaluation_date	timestamp
gauges	jsonb
estimated_implementation_date	timestamp
person_in_charge	character varying
pia_id	integer
created_at	timestamp
updated_at	timestamp
global_status	integer
TABLE: knowledge_bases	
id (PK)	bigint
name	character varying
author	character varying
contributors	character varying
is_example	boolean
created_at	timestamp
updated_at	timestamp
TABLE: knowledges	
id (PK)	bigint
name	character varying
slug	character varying
filters	character varying
category	character varying
placeholder	character varying
description	text

items	integer[]
knowledge_base_id	bigint
created_at	timestamp
updated_at	timestamp
TABLE: measures	
id (PK)	integer
title	character varying
content	text
placeholder	text
pia_id	integer
created_at	timestamp
updated_at	timestamp
TABLE: structures	
id (PK)	integer
name	character varying
sector_name	character varying
data	jsonb
created_at	timestamp
updated_at	timestamp
TABLE: pias	
id (PK)	integer
status	integer
name	character varying
author_name	character varying
evaluator_name	character varying
validator_name	character varying
dpo_status	integer
dpo_opinion	text
concerned_people_opinion	text
concerned_people_status	integer
rejection_reason	text
applied_adjustments	text
created_at	timestamp
updated_at	timestamp
dpos_names	character varying

people_names	character varying
is_example	integer
concerned_people_searched_opinion	boolean
concerned_people_searched_content	character varying
structure_id	integer
structure_name	character varying
structure_sector_name	character varying
structure_data	jsonb
is_archive	boolean
category	character varying
TABLE: revisions	
id (PK)	integer
export	jsonb
pia_id	bigint
created_at	timestamp
updated_at	timestamp

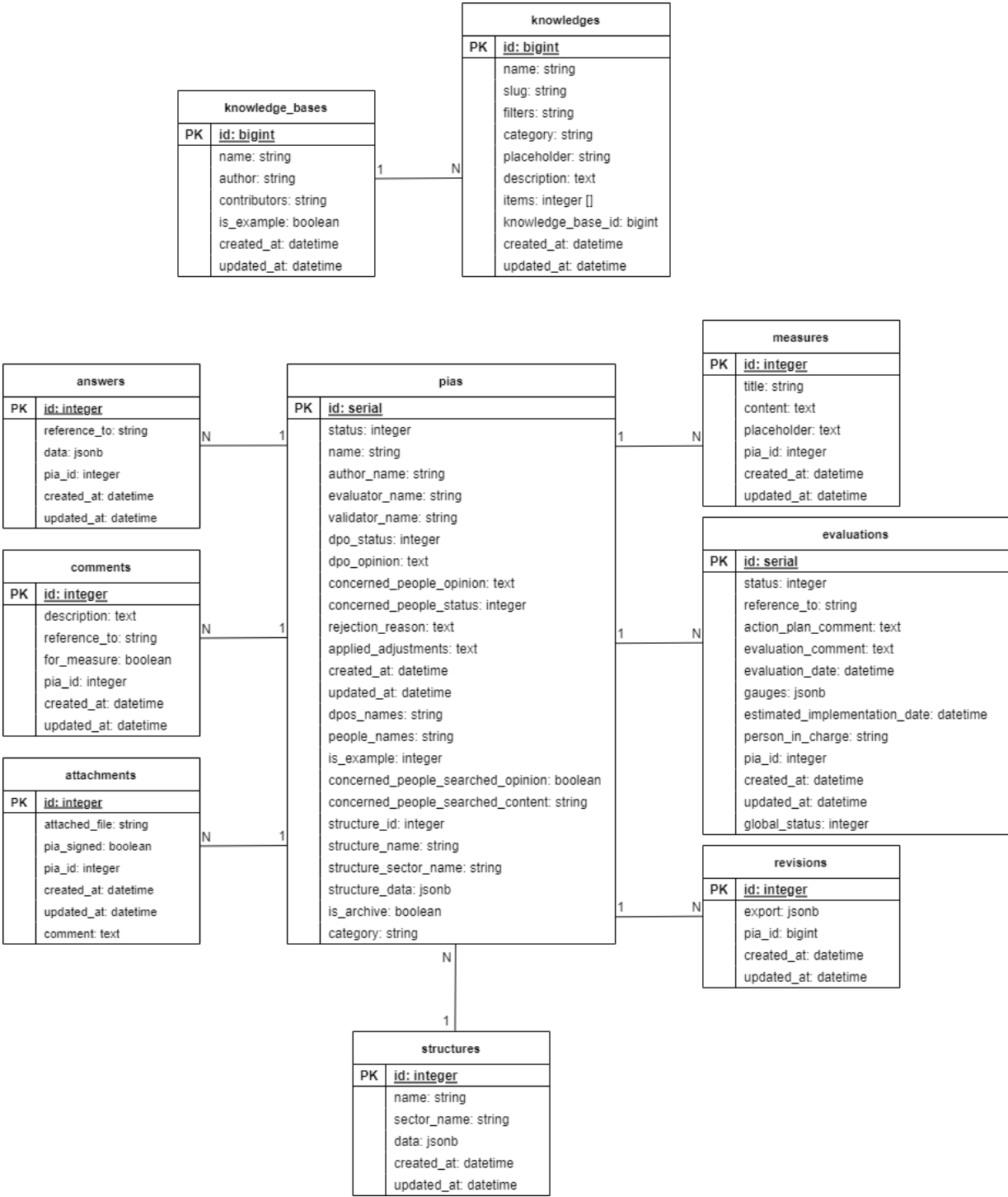


Figure 78: Database Schema of the DPIA ABB

Each DPIA is built upon a template that provides a standardized structure consisting of sections, items (sub-sections), and questions. The composition of a DPIA template's structure is represented by the required columns listed in Table

34 along with their data type. These fields outline the essential elements necessary for defining the structure of a DPIA template.

Table 34: Structure of Sections, Items and Questions

Section	
id	integer
items	list
title	string
short_help	string
Item	
id	integer
title	string
questions	list
short_help	string
evaluation_mode	string
Question	
id	integer
title	string
active	boolean
answer	string
answer_type	string
placeholder	string
link_knowledge_base	list

4.5.2.2. GRAPHICAL USER INTERFACE

The DPIA GUI currently follows the TRUSTEE UI/UX guidelines partially. But as this is an ongoing process, enhancement of the overall UI/UX and alignment with the platform’s guidelines will be reported in the following deliverables.

The welcome screen, depicted in Figure 79, serves as the initial screen of the tool, providing users with an introductory interface to the application. It serves as the starting point where users can orient themselves and navigate further into the tool's functionalities by pressing the “start” button.

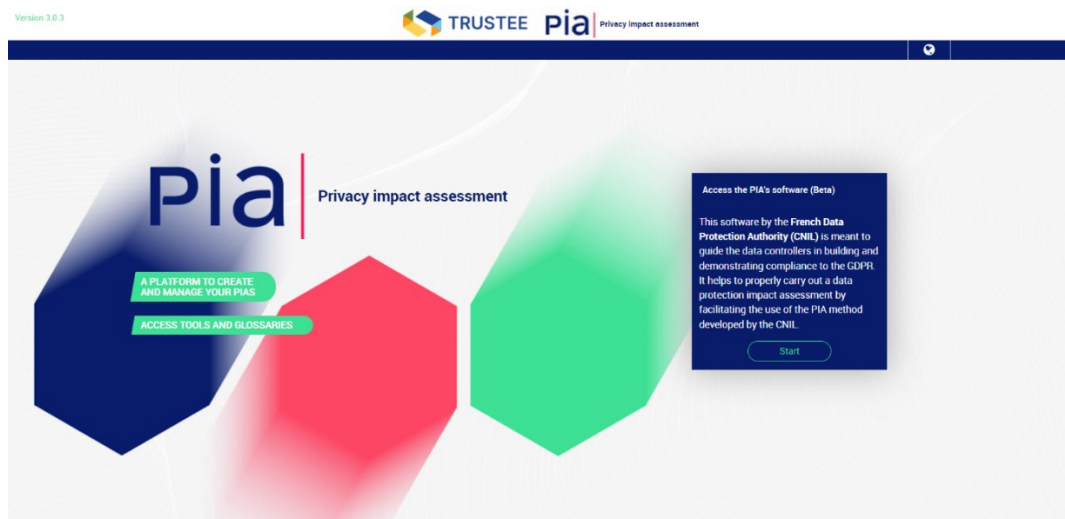


Figure 79: DPIA Interface Welcome Screen

All the screens of the user interface of the tool contain a header with the TRUSTEE and CNIL PIA logos, a navigation bar that is the menu of the tool and the main area.

The user interface of the tool features a consistent layout across all screens. It includes a header section displaying the TRUSTEE and CNIL PIA logos, providing visual branding. The navigation bar serves as the tool's menu, allowing users to access various functionalities. The main area of the interface is dedicated to displaying the relevant content and interactive elements based on the selected menu option. This consistent structure ensures easy navigation and a unified user experience throughout the tool. The menu encompasses a set of navigation items that facilitate easy access to different sections of the tool. These include: a) My PIAS, b) PIA Templates, c) Knowledge Base, d) Settings, e) Help and f) Language.

Figure 80 showcases the DPIAs screen (My PIAs), which displays all the DPIAs in progress or completed. The screen offers users various options for managing their DPIAs effectively. Users can create/start a new DPIA process or import a previously exported DPIA by clicking on the "plus" icons. It also provides flexibility in viewing DPIAs, offering both list and card view options. Furthermore, the screen includes a search functionality to help users easily locate specific DPIAs within their collection. Sorting options are also available to facilitate organizing the DPIAs based on various criteria.



Figure 80: DPIAs Screen

Figure 81 displays the DPIA Archives screen, which provides similar functionality to the DPIAs screen. However, there is a distinction in that the DPIA Archives screen specifically showcases the archived DPIAs and users cannot create new ones. Users can leverage the DPIA Archives screen to view and access the DPIAs that have been previously archived. While the ability to create new DPIAs is disabled, users can still benefit from functionalities such as searching, sorting, and different viewing options. This enables users to effectively manage their archived DPIAs and retrieve relevant information as needed. This screen ensures a clear separation between active DPIAs and archived ones, facilitating organization and maintaining a comprehensive record of completed or inactive assessments.



Figure 81: DPIA Archives Screen

Figure 82 presents the screen showcasing the available DPIA templates. Users can explore the various templates displayed on this screen, gaining insights into the predefined structures for different types of DPIAs. The screen allows users to create new templates or import existing ones, providing flexibility in template management. Users can utilize essential functionalities such as searching, sorting, and different viewing options. These features enable users to easily locate specific templates, organize them based on criteria, and explore template details as needed. By offering a dedicated screen for DPIA templates, the tool empowers users to leverage predefined structures, save time in creating new assessments, and ensure consistency in conducting DPIAs across different projects or scenarios.

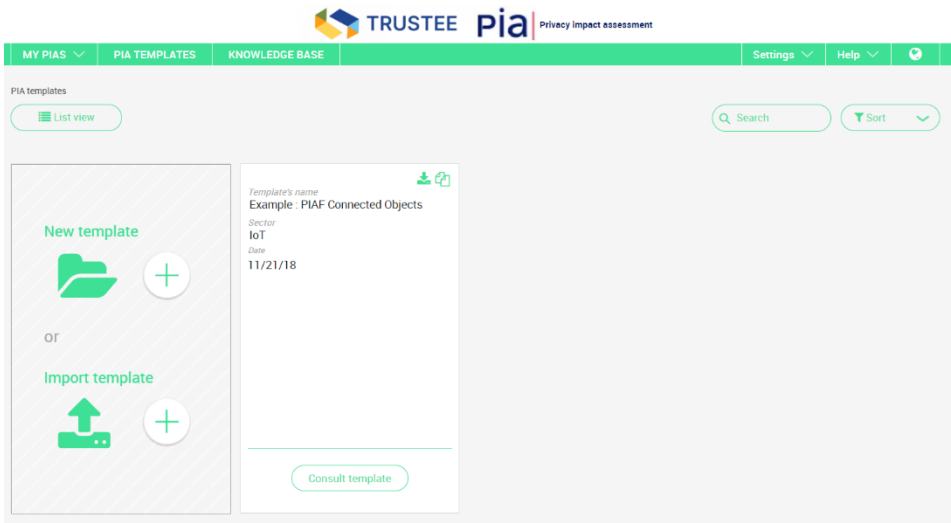


Figure 82: DPIA Templates Screen

Figure 83 illustrates the knowledge bases screen, providing users with an overview of the available legal and technical knowledge bases. On this screen, users can explore the existing knowledge bases, gaining access to valuable resources. Additionally, users have the option to create new knowledge bases or import existing ones, enhancing the versatility of knowledge management within the tool. Similar to previous screens, the knowledge bases screen incorporates a user-friendly approach. Users can utilize functionalities such as searching, sorting, and different viewing options to efficiently locate specific knowledge bases, organize them according to their preferences, and explore their contents. By presenting a dedicated screen for knowledge bases, the tool facilitates knowledge sharing and ensures that users can access relevant legal and technical information easily.

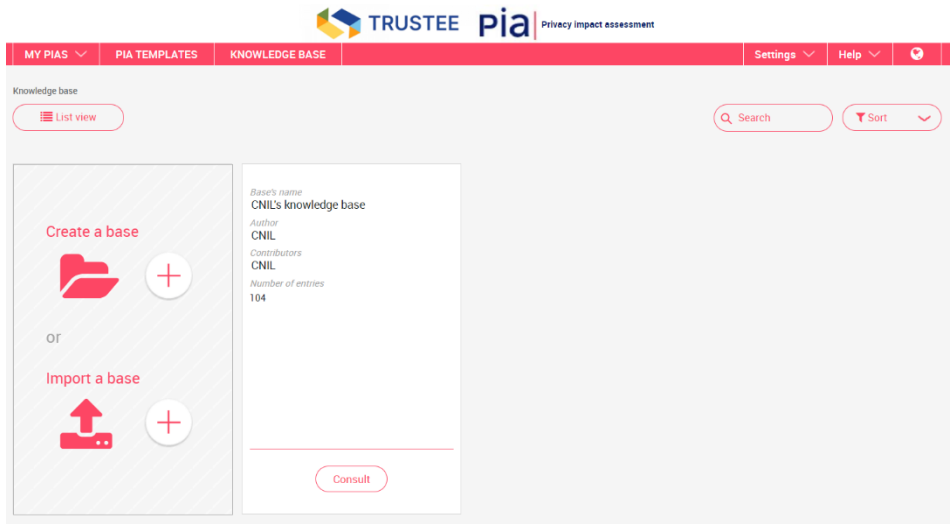


Figure 83: DPIA Knowledge Bases Screen

Figure 84 presents the forms for creating new DPIAs, DPIA templates, and knowledge bases, enabling users to provide essential information for each element. In the form for creating a new DPIA, users can input details such as a title for the DPIA, the names of the editor, reviewer, and validator, and specify the category to which the DPIA belongs. Additionally, users have the option to select an available template that serves as a starting point, streamlining the DPIA

creation process. When creating a new template, users define a title for the template and specify the associated sector's name. This allows for customization and tailoring of templates to align with specific industry or organizational requirements. For the creation of a new knowledge base, users are required to provide a base name, the name of the author, and include the names of contributors. This facilitates the creation and collaboration of comprehensive knowledge bases within the tool. By offering these well-defined forms, the tool ensures that users can input the necessary details to create new DPIAs, templates, and knowledge bases effectively.

The image displays three side-by-side form windows, each with a close button (X) in the top right corner and a 'Start' button at the bottom.

- Left Window (PIA):** The title bar says 'PIA'. The main content area has a section titled 'PIA's title' with a sub-label 'Editing' and a text input field 'Firstname, lastname'. Below this is a section titled 'Review' with a sub-label 'Firstname, lastname'. Further down is a section titled 'Validation' with a sub-label 'Firstname, lastname'. At the bottom of the main content area is a section titled 'Category' with a sub-label 'Category'. Below that is a section titled 'Template' with a sub-label 'Default template'. A horizontal line separates the main content area from the 'Start' button.
- Middle Window (Template's name):** The title bar says 'Template's name'. The main content area has a section titled 'Template's title' with a sub-label 'Sector' and a text input field 'Associated sector's name'. A horizontal line separates the main content area from the 'Start' button.
- Right Window (Base's name):** The title bar says 'Base's name'. The main content area has a section titled 'Base's name' with a sub-label 'Author' and a text input field 'Author'. Below this is a section titled 'Contributors' with a sub-label 'Contributors'. A horizontal line separates the main content area from the 'Start' button.

Figure 84: New DPIA, DPIA Template and DPIA Knowledge Base Forms

Users have the flexibility to edit a DPIA template by completing the forms in the screen depicted in Figure 85. Within these forms, users can pre-fill existing questions and even add new ones to further customize the template. By utilizing the forms, users can tailor the DPIA template to their specific needs. They can provide relevant information, address specific requirements, and adapt the template to align with their organization's policies or industry standards. The ability to pre-fill questions streamlines the process and ensures consistency, while the option to add new questions enables users to incorporate any additional considerations, they deem necessary.

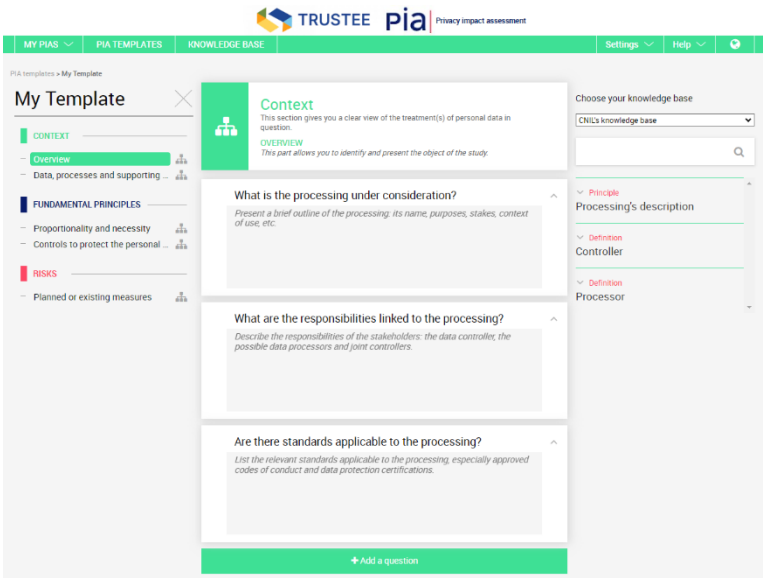


Figure 85: DPIA Templates Screen

Figure 86 shows the DPIA knowledge base screen, providing users with access to view the elements (knowledges) contained within a specific knowledge base. Users can also create new elements by pressing the "Create an element" button. The list of available elements presents key details, including their names, categories, and the date of the last modification. In Figure 87, the form for creating a new element is displayed, encompassing all the available options. Users are required to define the element's name, provide a description, assign a category, and select the sections with which the element should be linked. This linkage ensures that the element appears when editing a DPIA, offering relevant information and guidance. The provided form empowers users to efficiently create new knowledge elements within the knowledge base.



Figure 86: DPIA Knowledge Base Screen

×

CONTENTS

Choose a category

Element's name

Your description here...

LINKED TO

☐ ALL SECTIONS

☐ CONTEXT

☐ Overview

☐ Data, processes and supporting ass...

☐ FUNDAMENTAL PRINCIPLES

☐ Proportionality and necessity

☐ Controls to protect the personal rig...

☐ RISKS

☐ Planned or existing measures

☐ Illegitimate access to data

☐ Unwanted modification of data

☐ Data disappearance

☐ Risks overview

☐ VALIDATION

☐ Risk mapping

☐ Action plan

☐ DPO and data subjects' opinions

☐ Validate the PIA

☐ Rejected PIA

Create this element

Figure 87: DPIA Knowledge Base Elements Form

The DPIAs main screen, as depicted in Figure 88, enables users to edit a DPIA effectively. The screen is divided into three areas to support seamless navigation. On the left side, users can access the available sections and sub-sections (items) of the DPIA, allowing for easy navigation through the different components of the assessment. In the middle area, users can view and interact with the questions corresponding to the selected item, facilitating the input and modification of relevant information. The right side of the screen offers users the ability to select a knowledge base and view the available elements within that base. Users can further refine their search for specific elements by utilizing the search field. Additionally, when users click inside any editable element in the middle area, they can view the available knowledges for that specific item and the related contextual help (entry aids), assisting users in understanding and providing accurate information. At the bottom left of the screen, users have options to upload attachments related to the DPIA and create a new version of the current assessment, allowing for efficient document management and

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

version control. This well-structured DPIAs main screen enhances usability, streamlines the editing process, and provides users with the necessary resources and tools to effectively complete and manage their DPIAs.

Figure 88: DPIA Main Screen

Figure 89 presents the form fields within the risks section, requiring users to provide essential information. Users can input the impacts, threats, and sources associated with the selected risk by adding multiple elements using the text input field and pressing “tab” or “enter” to add additional items. In the field for identifying planned controls, users can choose from the available options defined in the sub-section “Planned or existing measures”. This selection ensures that users can associate the appropriate controls with the specific risk being addressed. For the last two fields related to the estimation of risk severity and likelihood, users can utilize a slide bar (gauge) to select the desired level. This visual representation enables users to choose the appropriate severity and likelihood levels while providing a clear understanding of the risk assessment. Users can also include their justification for the chosen levels. By incorporating these user-friendly form fields, the tool streamlines the process of capturing essential risk-related information, facilitates precise control identification, and enables users to provide a comprehensive risk assessment.

What could be the main impacts on the data subjects if the risk were to occur?

Enter the potential impacts

What are the main threats that could lead to the risk?

Enter the threats

What are the risk sources?

Enter the risk sources

Which of the identified planned controls contribute to addressing the risk?

Click here to select controls which address the risk.

How do you estimate the risk severity, especially according to potential impacts and planned controls?

(Undefined)NegligibleLimitedImportantMaximum

Justify here the estimated severity of the risk.

How do you estimate the likelihood of the risk, especially in respect of threats, sources of risk and planned controls?

(Undefined)NegligibleLimitedImportantMaximum

Justify here the estimated likelihood.

Figure 89: DPIA Risks Assessment

Once users have completed filling in the questions of a sub-section, they can initiate a review process by clicking the "ask for review" button, as depicted in Figure 90. This functionality allows users to request a thorough evaluation and feedback on the completed sub-section. By clicking the "ask for review" button, users signal that they have finished their input and are ready for the review phase to commence.

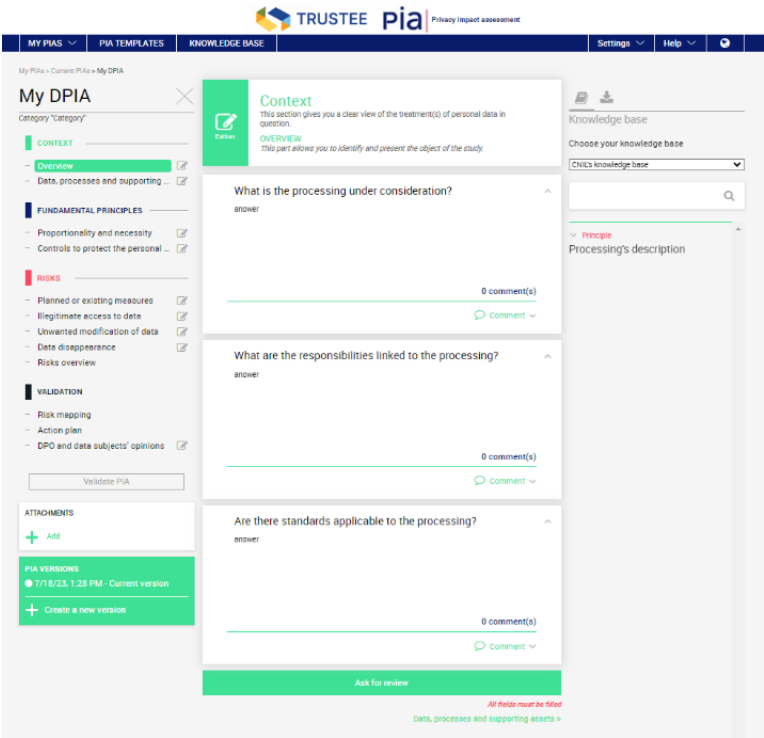


Figure 90: DPIA Review Request

The DPIA Interface includes various pop-up messages, as depicted in Figure 91, to provide users with important notifications and updates regarding the status of their DPIA. One such pop-up message informs users when the current section is ready to be reviewed. This notification serves as a reminder that the section has been completed and is now available for the review process to commence. It ensures that users are aware of the next step in the DPIA workflow and can proceed accordingly. Another pop-up message indicates when the current section is ready for the overall validation of the DPIA. This notification signifies that the section has been completed and reviewed. Additionally, a message is displayed when users have previously chosen to send a section to the edition phase for corrections. This serves as a reminder of any required revisions or modifications that need to be made before the section can progress further in the DPIA process. These pop-up messages enhance user awareness, keeping them informed of the status and actions required for their DPIA. By providing timely notifications, the DPIA Interface ensures that users can effectively manage and complete their assessments with accuracy and efficiency.

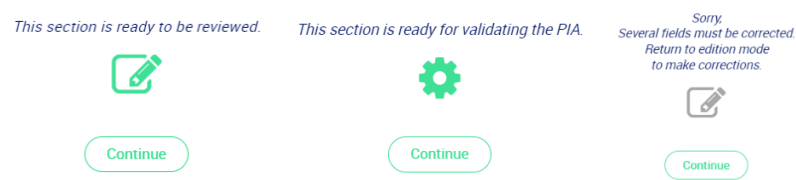


Figure 91: DPIA Messages

The review process enables reviewers to assess the submitted sub-section, provide comments, suggest improvements, and offer validation or correction actions. This collaborative approach ensures that the DPIA is thoroughly evaluated and enhances the quality and accuracy of the assessment.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

During the review phase, users have the ability to cancel the review request and return to the edition phase by clicking on the highlighted text, as depicted in Figure 92. In this way, they have the opportunity to refine and improve their work before finalizing the section.

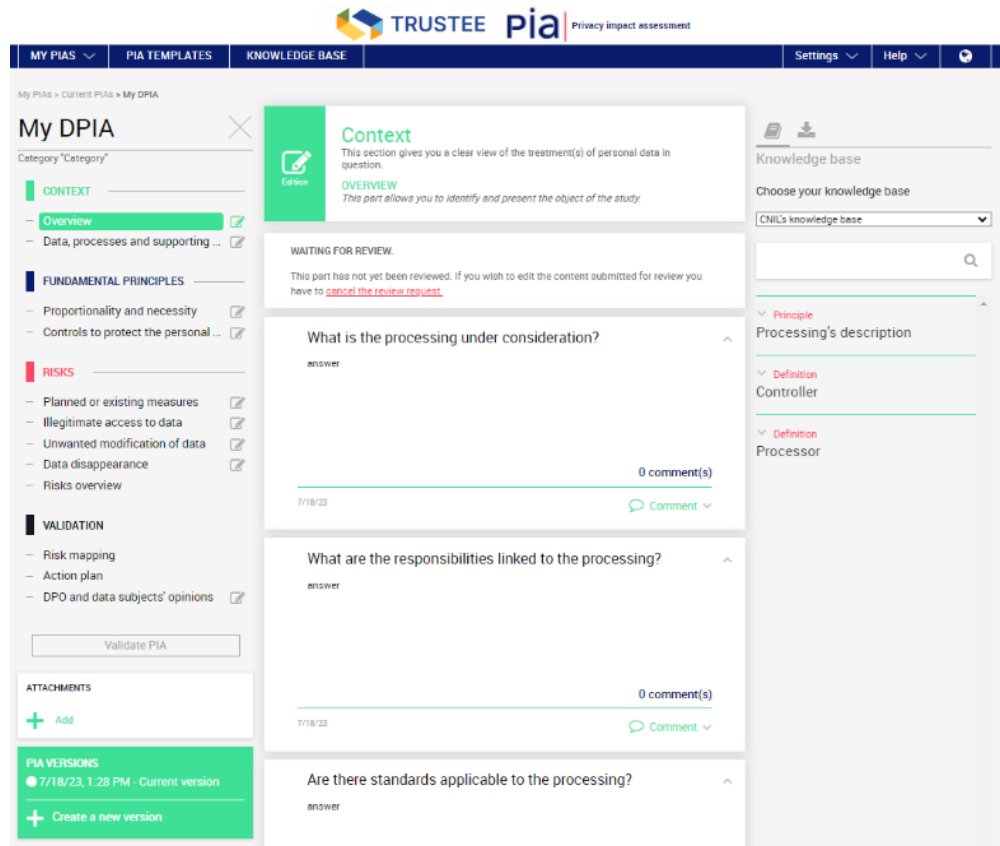


Figure 92: DPIA Review Phase

During the validation phase, users can cancel the validation request and return to the review phase by clicking on the highlighted text, as depicted in Figure 93. This functionality allows users to reconsider the validation request and make any necessary adjustments or revisions.

By including this feature of returning in the previous phase of the DPIA process, the tool promotes a continuous improvement approach, enabling users to actively manage and refine their DPIA throughout all the phases.

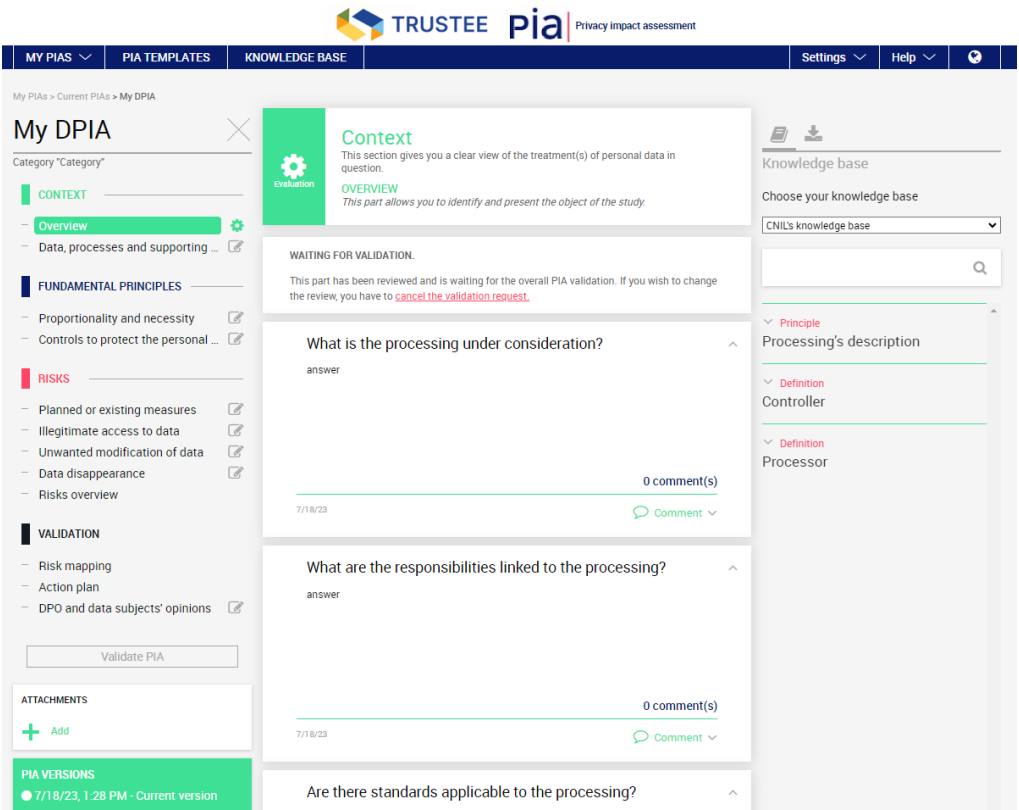


Figure 93: DPIA Validation Phase

Users reviewing different sections have various options available to them depending on the context which are displayed in Figure 94. They can choose to reject a section and send it back for corrections by clicking on the "To correct" toggle button. Alternatively, they can accept the section by either clicking "Acceptable" or "Improvable" and providing relevant comments or feedback. By selecting the "To correct" option, users indicate that the section requires revisions or adjustments. This initiates the process of returning the section to the editing phase for necessary corrections to be made. When users choose the "Acceptable" or "Improvable" options, they express their acceptance of the section with the option to provide comments or suggestions for improvement. This allows users to provide constructive feedback, propose enhancements, or highlight areas where further improvements can be made.

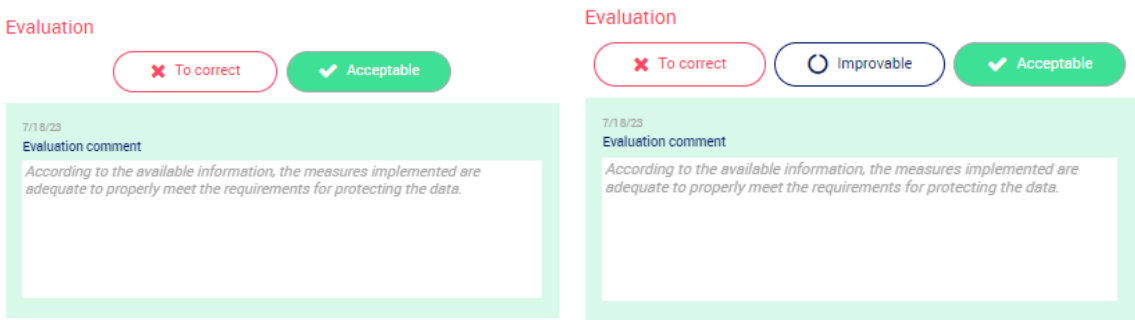


Figure 94: DPIA Evaluation Options

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The tool incorporates various visualizations, including “Risks overview”, “Risk mapping”, and “Action plan”. In Figure 95, a visualization demonstrates the mapping of impacts, threats, sources, and measures with different risks, including the assessment of severity and likelihood for each risk. This visualization provides a comprehensive view of the risk landscape, helping users understand the relationships between various risk factors and the associated severity and likelihood. Another visualization in Figure 96 showcases the positioning of risks based on their likelihood and severity, both before and after the implementation of complementary controls. This comparison enables users to evaluate the effectiveness of the applied controls in mitigating risks and identify any remaining areas of concern. The “Action plan” visualization, as depicted in Figure 97, offers an overview of the controls, indicating whether the measures are deemed acceptable or improvable. Additionally, it presents a list of recorded action plans, outlining the necessary steps or actions required to address any identified gaps or improvements. These visualizations provide users with valuable insights into the risk landscape, the effectiveness of controls, and the necessary action plans. By utilizing these visual representations, users can make informed decisions, prioritize areas for improvement, and ensure the ongoing compliance and mitigation of risks within the DPIA process.

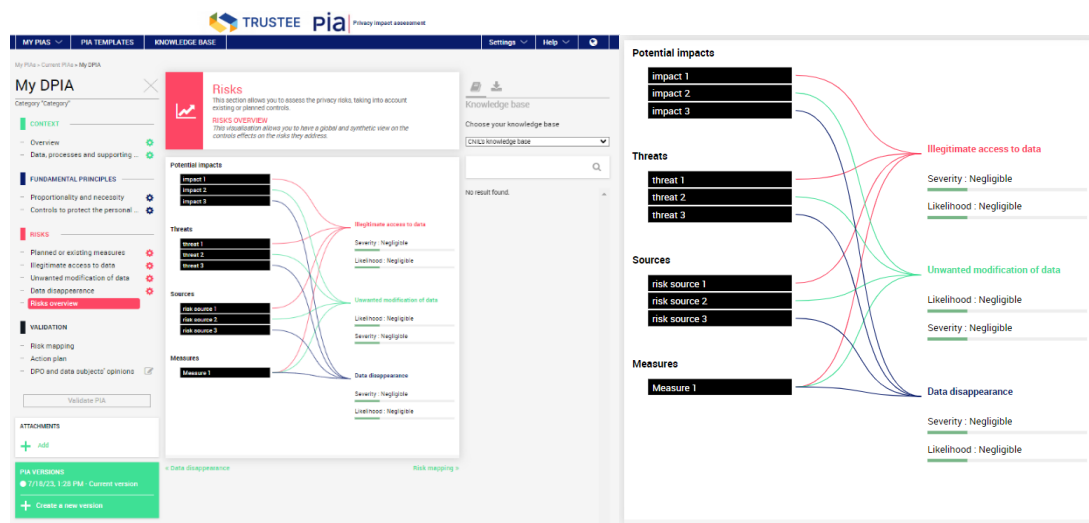


Figure 95: DPIA Risks Overview Visualisation

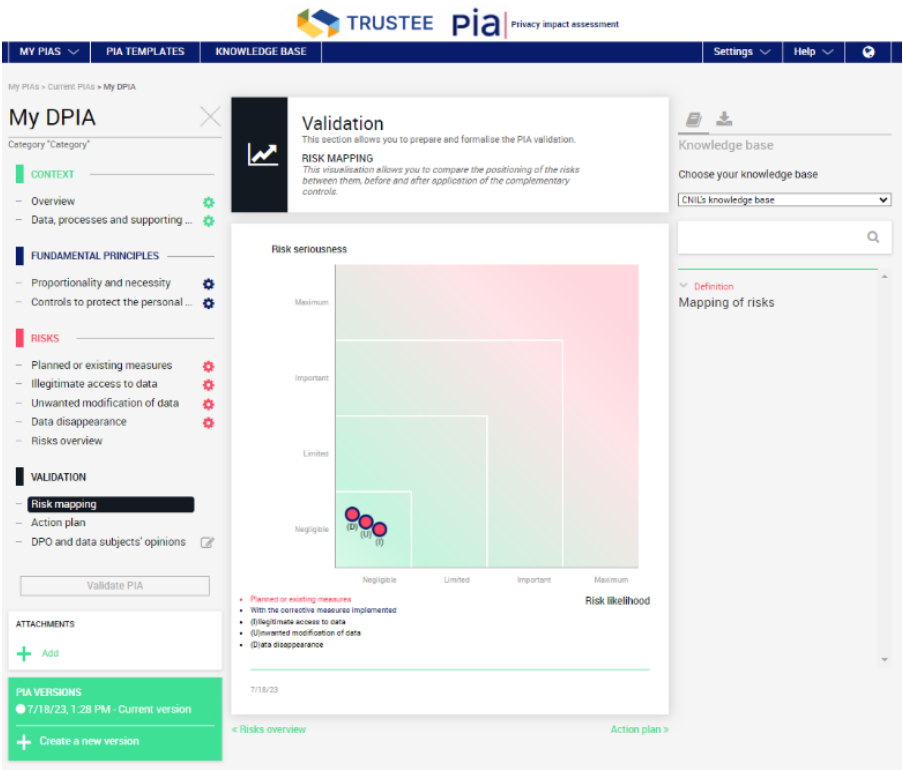


Figure 96: DPIA Risk Mapping Visualisation

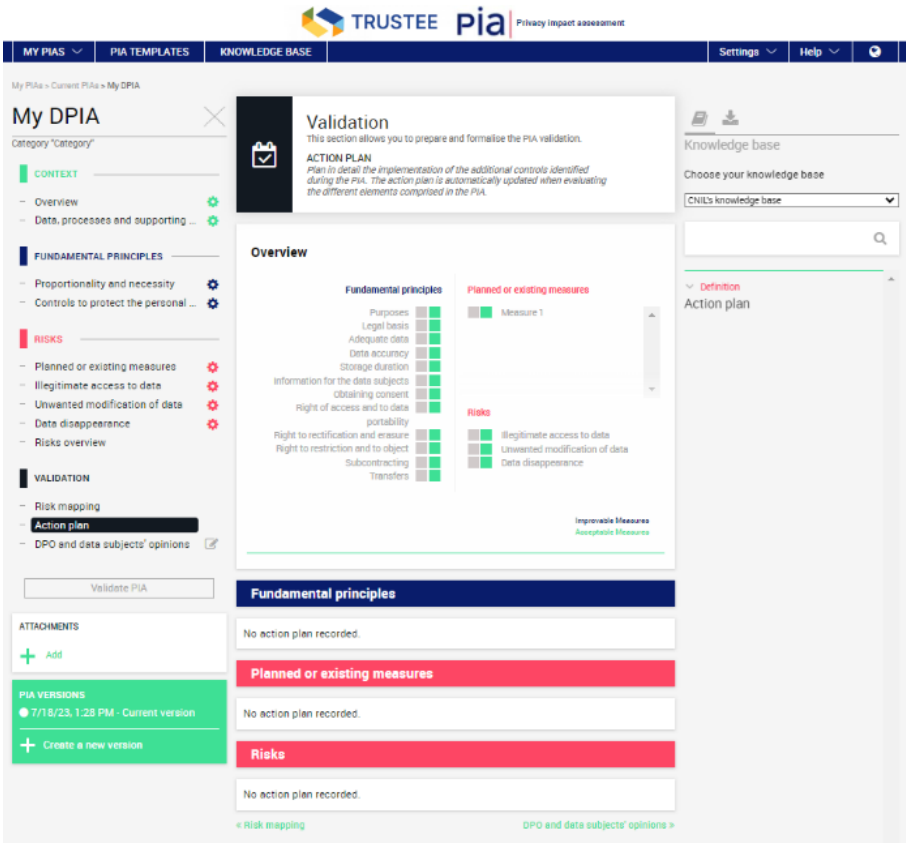


Figure 97: DPIA Action Plan Visualisation

The final step of a DPIA is the validation, which is facilitated through the screen depicted in Figure 98. In this screen, users are required to provide the names and opinions of the Data Protection Officer (DPO) and Data Subjects. They can then proceed by checking the checkboxes displayed in figure X and finally selecting the appropriate button based on whether they reject or approve the current DPIA. Users also have the option to upload a signed document for validation. By filling in the DPO's and Data Subjects' names and opinions, users ensure that the relevant stakeholders' perspectives are captured in the validation process. The checkboxes presented in Figure 99 serve as a checklist to confirm that all necessary steps have been taken and considerations have been addressed. Through this validation screen, the tool streamlines the final step of the DPIA process, ensuring that all necessary parties' opinions are considered and that the DPIA receives the required approvals.

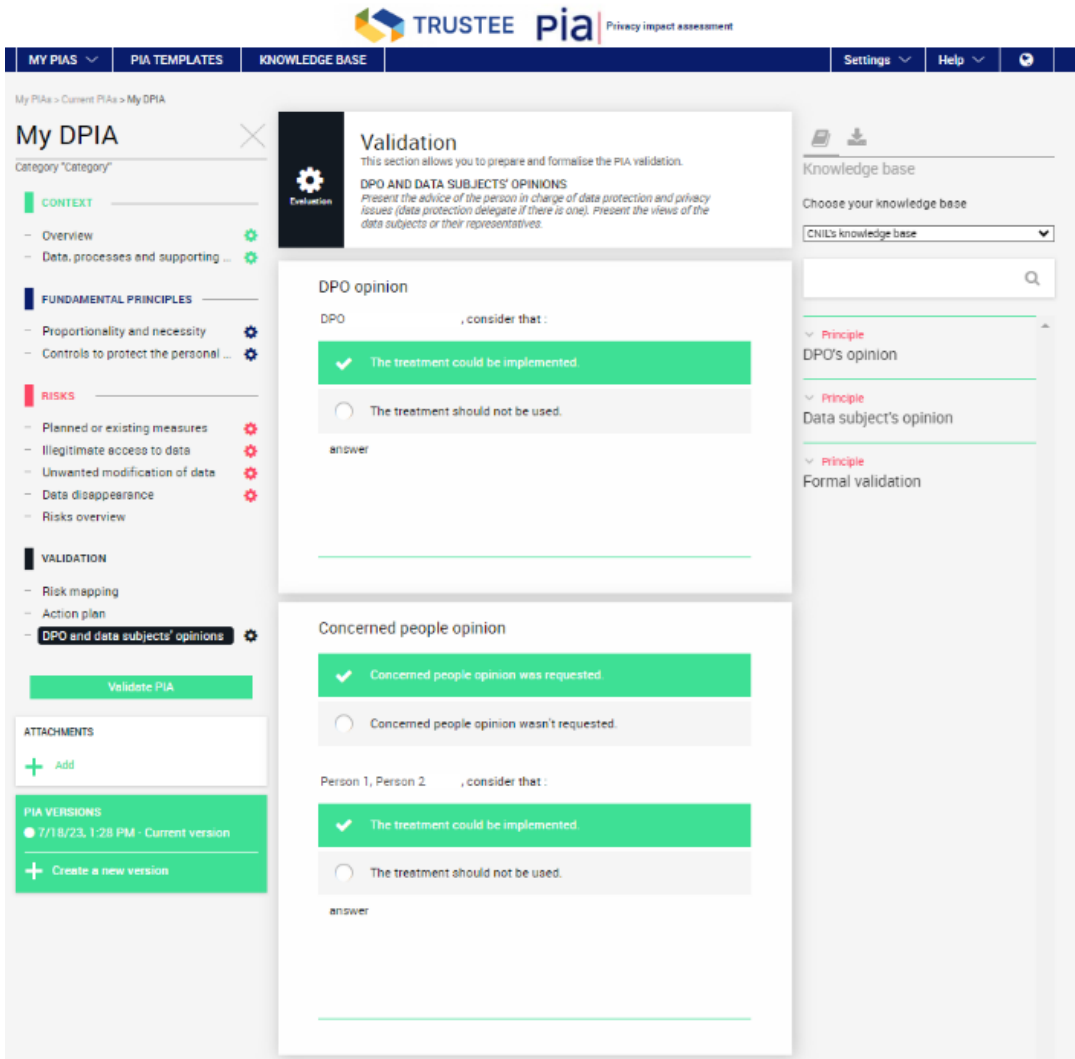


Figure 98: DPO and Data Subjects' Opinions Screen

(7/18/23)

Person accountable for validating the PIA: **Validator**.

After reading full PIA relating to the data processing linked to **My DPIA**

✓ I attest that the context of the processing described in the PIA is consistent with the reality.

✓ I attest that I am aware of the risks depending on the existing or planned measures.

✓ I ratify the corrective actions indicated in the action plan.

✓ I commit to implement as quickly as possible the indicated corrective actions.

Reject PIA
Approve PIA (direct)
Approve PIA (signed)

All boxes must be checked

Figure 99: DPIA Final Rejection or Approval

Table 35 provides an overview of the available sections and sub-sections (items) within the DPIA, along with concise descriptions and the associated questions. The table allows users to quickly navigate and understand the structure of the DPIA. Each section and sub-section are accompanied by a short description, providing context and outlining its purpose within the assessment. The included questions give users an idea of the specific information that needs to be addressed within each section.

Table 35: Sections, sub-sections and included questions

Section #01: Context		
This section gives you a clear view of the treatment(s) of personal data in question.		
1. Overview	This part allows you to identify and present the object of the study.	1. What is the processing under consideration? 2. What are the responsibilities linked to the processing? 3. Are there standards applicable to the processing?
2. Data, processes and supporting assets	This part allows you to define and describe the scope of the processing in detail.	1. What are the data processed? 2. How does the life cycle of data and processes work? 3. What are the data supporting assets?
Section #02: Fundamental principles		
This section allows you to build the compliance framework for privacy principles.		
1. Proportionality and necessity	This part allows you to demonstrate that you are implementing the necessary means to enable the persons	1. Are the processing purposes specified, explicit and legitimate? 2. What are the legal basis making the processing lawful? 3. Are the data collected adequate, relevant and limited to what is necessary in relation

	concerned to exercise their rights.	to the purposes for which they are processed ('data minimisation')? 4. Are the data accurate and kept up to date? 5. What are the storage duration of the data?
6. Controls to protect the personal rights of data subjects	This part allows you to demonstrate that you are implementing the necessary means to enable the persons concerned to exercise their rights.	1. How are the data subjects informed on the processing? 2. If applicable, how is the consent of data subjects obtained? 3. How can data subjects exercise their rights of access and to data portability? 4. How can data subjects exercise their rights to rectification and erasure? 5. How can data subjects exercise their rights to restriction and to object? 6. Are the obligations of the processors clearly identified and governed by a contract? 7. In the case of data transfer outside the European Union, are the data adequately protected?
Section #03: Risks		
This section allows you to assess the privacy risks, taking into account existing or planned controls.		
1. Planned or existing measures	This section allows you to identify controls (existing or planned) that contribute to data security.	1. Add an empty measure (otherwise, use the knowledge base)
2. Illegitimate access to data	Analyze the causes and consequences of illegitimate access to data, and estimate its severity and likelihood.	1. What could be the main impacts on the data subjects if the risk were to occur? 2. What are the main threats that could lead to the risk? 3. What are the risk sources? 4. Which of the identified planned controls contribute to addressing the risk? 5. How do you estimate the risk severity, especially according to potential impacts and planned controls? 6. How do you estimate the likelihood of the risk, especially in respect of threats, sources of risk and planned controls?
3. Unwanted modification of data	Analyze the causes and consequences of an undesired change in data, and estimate its seriousness and likelihood.	1. What could be the main impacts on the data subjects if the risk were to occur? 2. What are the main threats that could lead to the risk? 3. What are the risk sources? 4. Which of the identified planned controls contribute to addressing the risk? 5. How do you estimate the risk severity, especially according to potential impacts and planned controls? 6. How do you estimate the likelihood of the risk, especially in respect of threats, sources of risk and planned controls?

4. Data disappearance	Analyze the causes and consequences of data loss, and estimate its seriousness and likelihood.	1. What could be the main impacts on the data subjects if the risk were to occur? 2. What are the main threats that could lead to the risk? 3. What are the risk sources? 4. Which of the identified planned controls contribute to addressing the risk? 5. How do you estimate the risk severity, especially according to potential impacts and planned controls? 6. How do you estimate the likelihood of the risk, especially in respect of threats, sources of risk and planned controls?
5. Risks overview	This visualisation allows you to have a global and synthetic view on the controls effects on the risks they address.	There are no questions in this sub-section.
Section #04: Validation		
This section allows you to prepare and formalise the PIA validation.		
1. Risk mapping	This visualisation allows you to compare the positioning of the risks between them, before and after application of the complementary controls.	There are no questions in this sub-section.
2. Action plan	Plan in detail the implementation of the additional controls identified during the PIA. The action plan is automatically updated when evaluating the different elements comprised in the PIA.	There are no questions in this sub-section.
3. DPO and data subjects' opinions	Present the advice of the person in charge of data protection and privacy issues (data protection delegate if there is one). Present the views of the data subjects or their representatives.	There are no questions in this sub-section.

4.5.2.1. API

As mentioned in the architecture of the DPIA ABB, the DPIA Server is the back-end of the DPIA ABB and is implemented as a RESTful API. It provides several endpoints that execute the actions requested by the DPIA Interface. It can also be used by other components or ABBs to exchange data related to a DPIA with the database. Section 0 shows the available endpoints of the API including a) a description, b) the endpoint, c) the authentication (if any), d) the method, e) the input parameters, f) an example of the request body, and g) an example of the response body. In the cases that include the parameter “body” the parameter content type is “multipart/form-data”. See ANNEX 4: DPIA ENDPOINTS for more details.

4.5.3. TECHNICAL DETAILS

Table 36 presents various technical details related to the DPIA ABB, including information about its development, installation and deployment. These details provide an overview of the tool's technical aspects, ensuring its effective implementation and utilization within TRUSTEE.

Table 36: DPIA Technical Details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	Linux (Debian)
Database and databases tools	PostgreSQL
GUI	YES (web)
Programming language	Ruby (Rails), javascript/typescript (Angular CLI)
Development Environment	
INSTALLATION AND DEPLOYMENT	
Software Requirements	Linux (preferable), Ruby (3.1), Rails (7.0), PostgreSQL (12.0), Nodejs (14.17.4), Angular CLI
Hardware Requirements	CPU (i5), RAM (4GB), Storage (20GB)
Containerization	docker
Communications	

4.6. VALIDATION

Python's unit test library is considered to create comprehensive unit tests for the purpose of assessing the functionality and intercommunication of the ABBs. By utilizing this testing framework, we aim to evaluate the individual units of code within the ABBs to ensure they perform as intended. These unit tests will provide us with a robust means of verifying the correctness of each ABB's behaviour in isolation and also assess how they interact and collaborate with one another within the larger system. Through this systematic testing approach, we seek to enhance the reliability and stability of our software, ultimately leading to a more seamless and efficient performance of our ABBs in real-world scenarios, as well as to facilitate testing the system throughout the development process.

4.7. NEXT STEPS

Moving forward, the implementation of the DPIA within TRUSTEE will entail several significant steps which are currently on going as part of the CI/CD methodology employed by the project:

- Developing and Integrating DPIA Application Modules: As the first step, we need to develop the application modules required for DPIA and integrate them with the existing TRUSTEE framework. This process includes building interfaces for data input, processing, and output related to the DPIA.
- Integration with Other Application Building Blocks: The DPIA tool is only one component of the broader TRUSTEE framework. The next step is ensuring its seamless interaction with other application building blocks. For example, integrating the DPIA tool with cybersecurity risk assessment frameworks will be crucial for achieving a balance between security and privacy.

- iii. **Testing and Debugging:** After the development and integration phases, it is essential to conduct rigorous testing of the entire system to ensure it functions as expected and to fix any bugs or issues that might arise.
- iv. **Validation Across Different TRUSTEE Pilots:** Once the DPIA tool has been integrated into the broader TRUSTEE framework, it needs to be validated across various TRUSTEE pilot projects. This means deploying the tool in real-world settings and assessing its performance in diverse environments.
- v. **Verification:** Concurrently with validation, the tool will undergo verification procedures. This includes ensuring that the tool complies with GDPR regulations and that it meets all the specifications outlined in the technical, user-centric, ethical/social, and legal requirements.
- vi. **Feedback and Iterative Improvement:** Upon validation and verification, feedback from users and stakeholders across different pilots will be collected. This feedback will be instrumental in identifying any areas of improvement, which will then be addressed in iterative cycles of improvement.

4.8. CONCLUSIONS

In conclusion, the integration of DPIA within TRUSTEE is a vital step towards enhancing the data protection capabilities of EU stakeholders. This chapter has aimed to shed light on the intricate workings of the DPIA, highlighting its importance and the process it follows. We have navigated through the legal and regulatory framework that underpin the necessity for DPIA, demonstrating the pivotal role of GDPR in guiding data protection practices. We have also delved into the data processing activities being assessed under DPIA, emphasizing the need for a comprehensive understanding of the nature, operations, and purpose of data processing. Subsequently, we discussed the various requirements for a well-rounded DPIA, considering technical, user-centric, ethical, social, and legal aspects. This was followed by an examination of the architectural layout of our DPIA tool, underscoring the significance of a well-structured, efficient mechanism to carry out the assessment. Furthermore, we addressed the implementation phase, illuminating the practical application of DPIA within the TRUSTEE framework. Lastly, the validation plan has been presented, underscoring our commitment to ensuring the effectiveness and reliability of our DPIA tool.

5. PART III: SELF-SOVEREIGN IDENTITY HOMOMORPHIC CAPABLE FRAMEWORK

5.1. INTRODUCTION

This chapter gathers all the main details of the SSI-HE ABB. The objective of the SSI HE is to develop a framework that combining the Self-Sovereign Identity (SSI) technology and HE allows to perform search in an encrypted domain ensuring data privacy and data confidentiality. This section presents a comprehensive overview of the SSI-HE component focusing on its core components, requirements, and implementation details.

5.1.1. NEED AND OBJECTIVES

Nowadays, data is a fundamental component of the digital economy. In fact, in recent years, the volume of generated data has experienced an exponential increase. Consequently, it is essential to characterize it in order to better identify which data can be useful or not for a specific purpose and, by this way, maximize profitability.

Although there are several secure solutions on the literature that already allow data characterization, such as the use of Self-Sovereign Identity (SSI) [39], sometimes, data characterization needs to be kept confidential, as metadata can also disclose sensitive details. SSI already addresses confidentiality issues for numeric values, allowing compliance proves without revealing the specific values through the use of Zero Knowledge Proofs (ZKPs) [40]. However, datasets metadata usually refers to string attributes.

There are also some solutions that allow to protect non-numeric values as long as they are part of a pre-existing list (through Zero Knowledge Set Membership [41]). However, nowadays, it is not possible to keep data confidential in general, covering both numeric and non-numeric values (strings) and allowing random values (which do not have to be part of a predefined list).

In this sense, TRUSTEE aims to extend the existing SSI solutions for data characterization including confidential random numeric and non-numeric values and allowing their validation keeping them confidential. For this purpose, the Homomorphic Encryption (HE) technology is considered, resulting in the SSI-HE ABB.

SSI-HE is the TRUSTEE component dedicated to the confidential characterization of datasets. On the one side, this component enables the generation and issuance of verifiable credentials with confidential (encrypted) attributes that characterize a dataset. On the other side, this component allows the attributes validation keeping them confidential.

For instance, imagine an example in which just data from Spain is needed and the location attribute for different data is defined as follows: “Main Street, 10, 5C, Madrid, Spain”. If the attribute is shared in clear, unnecessary information is revealed such as the city, the street, the number and the floor. However, the provision of the attribute homomorphically encrypted enhances confidentiality by allowing the search of “Spain” word inside the location attribute without revealing the rest of the location.

5.1.2. INNOVATION

Although SSI and HE are technologies that have been extensively analyzed in the literature, homomorphically encrypted attributes have never been considered inside SSI verifiable credentials. TRUSTEE will combine both technologies, extending the traditional ZKP and ZKP memberships that are currently part of existing SSI solutions. By

this way, all type of attributes (not only numeric or prefixed ones) could be kept confidential while being able to be proved if needed.

5.2. BACKGROUND

This section includes a high-level explanation of the two main technologies considered for the SSI-HE component and their main features: homomorphic encryption and Self-Sovereign Identity. More details can be found in D2.1 [1], where both technologies have been deeply analyzed.

5.2.1. HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a kind of cryptography technique that allows mathematical operations to be performed directly on encrypted data [42]. In other words, it enables manipulation of the encrypted data without revealing its original content. In a homomorphic cryptography system, data is encrypted using a specific encryption algorithm that preserves certain mathematical properties. This technology allows certain operations, such as addition or multiplication, to be performed on the encrypted data, generating an outcome that remains encrypted. Subsequently, the encrypted outcome can be decrypted to obtain the operation's result on the original data. There are different types of homomorphic cryptography [43], such as partial homomorphism, which allows certain but not all operations to be performed, and full homomorphism, which allows any operation to be performed.

Homomorphic cryptography has many applications in areas such as cloud data processing, private data analysis and distributed computing security. However, it is crucial to note that homomorphic cryptography is still a developing field and presents challenges in terms of efficiency and scalability [44] [45]. In exchange for enhanced security, homomorphic algorithms tend to be slower, and demand greater computational resources compared to operations performed on clear data. As research and development continues in this area, homomorphic cryptography is expected to improve, leading to broader future applications.

5.2.1.1. OPENFHE

OpenFHE²⁷ is an open-source library with BSD 2-Clause License²⁸ that has been developed by *NumFOCUS: A Nonprofit Supporting Open Code for Better Science*²⁹. This project to develop an open homomorphic cryptography library started under the name PALISADE, and evolved into OpenFHE in 2022, when the old library was combined with some of the best capabilities coming from other homomorphic open-source developers in the HELib³⁰ and HEAAN³¹ projects. At the time of writing (July 2023), the latest version of the library is v1.0.4.

²⁷ <https://www.openfhe.org/>

²⁸ <https://opensource.org/license/bsd-2-clause/>

²⁹ <https://numfocus.org/>

³⁰ <https://github.com/homenc/HElib>

³¹ <https://heaan.it/>

OpenFHE details have been gathered in D2.1 [1]. However, as a summary, it includes functionalities for the main known fully homomorphic cryptography schemes, i.e., BFV/BGV, CKKS and TFHE, allowing to work with data of almost any type. For BGV and CKKS, it provides the possibility to work with asymmetric encryption, as well as other useful tools such as Bootstrapping [46], Key-Switching [47] or, the most relevant one for TRUSTEE, Threshold FHE [48].

5.2.1.2. THRESHOLD FHE

Threshold Fully Homomorphic Encryption (Threshold FHE) is one of the functionalities that OpenFHE provides for some of the homomorphic schemes [48] [49].

Threshold FHE relies on Key-switching functionality to generate a public key that binds multiple private keys, so that all ciphertext encrypted from that key must be decrypted using all private keys that are bound to that cryptographic context. This is achieved by decrypting fragments of the ciphertext: each key gets a fragment, and the clear result is not obtained until all fragments are merged. The goal of Threshold FHE is to ensure that no single entity can decrypt the data or obtain sensitive information about it.

Threshold FHE is an active field of research in cryptography and has diverse applications in protecting the privacy and security of confidential data, especially in distributed environments where multiple parties must collaborate securely without disclosing confidential information. Figure 100 shows the process needed to operate using Threshold FHE between Alice and Bob, including the generation of the required cryptographic context and all the linked keys necessary to perform the cryptographic operations of encryption and decryption.

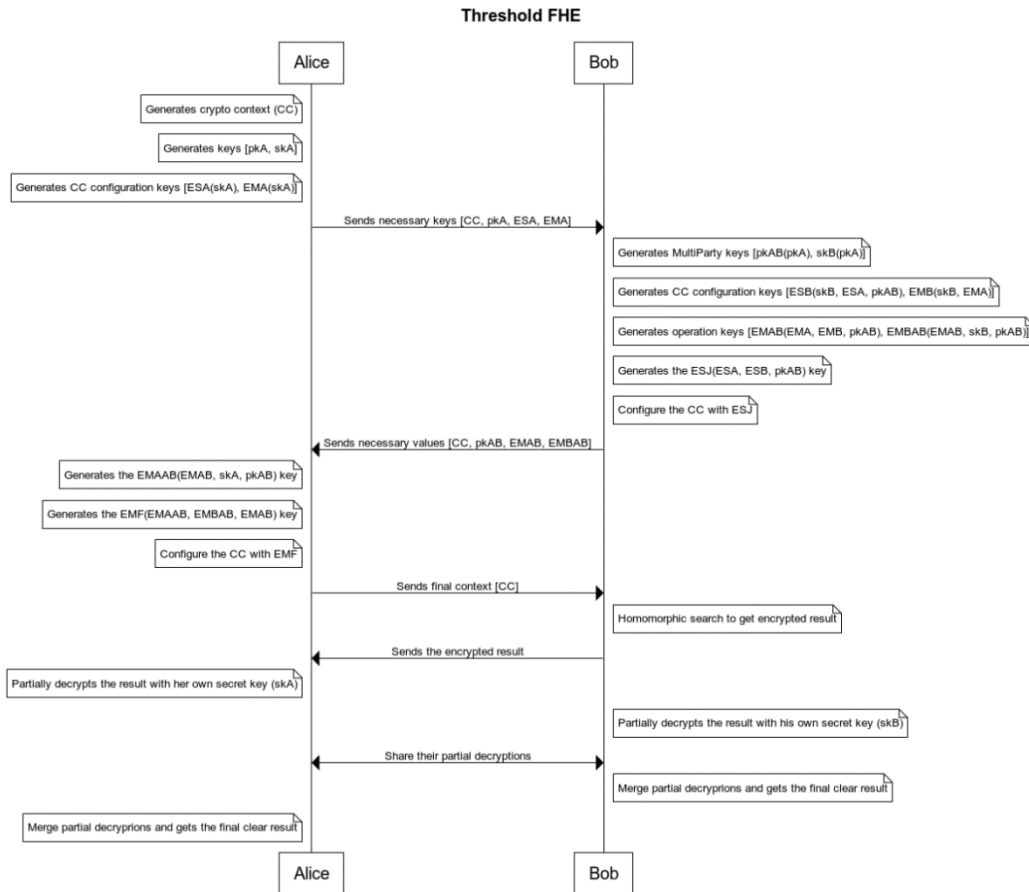


Figure 100: Threshold FHE operation

The first step is similar to a normal homomorphic encryption flow without the use of Threshold FHE. Alice generates the Crypto Context (CC) using the required configuration and then creates a first pair of keys, public (pk) and secret (sk), from this context. Bob must know the configuration considered by Alice to be able to regenerate the same CC, and if he does not know it, Alice needs to send it to him.

Alice will then create some intermediate keys EvalMultKey (EMA) and EvalSumKeys (ESA) which need to be shared with Bob along with the public key and context.

Once Bob receives all the required information from Alice, he starts the generation of the combined own intermediate keys. For this purpose, he first generates the multiparty keys based on the public key received from Alice. As a result, Bob's secret key and a new public key for Alice and Bob (pkAB) are generated. This public key will be the one to be considered by Alice and Bob from this moment.

As Alice does, Bob generates his own intermediate keys EvalSumKeysB (ESB) and EvalMultKeyB (EMB), using his own secret key and the new generated public key. These values are also dependent on Alice's ESA and EMA keys, respectively.

At this moment, both Alice and Bob have generated intermediate keys that are needed for the final combined keys generation.

- In the case of the Sum operation, it can be automatically generated by Bob, as it only depends on ESA, ESB and the new public key. Bob already knows all this information. This final combined EvalSumKeysJoin (ESJ) is obtained and included in the CC to be able to perform sum operations on the homomorphic domain. Once Bob generates ESJ, he updates the CC including it and he shares it again with Alice.
- In the case of the Mult operation, the final combined key generation still needs two keys: firstly, Bob obtains EvalMultAB (EMAB) combining the two intermediate Mult keys from Alice and Bob and the new public key. Additionally, Bob needs to calculate EvalMulBAB (EMBAB), considering EMAB he has just calculated, his own secret key and the final publickey (pkAB). All this information, except for the Bob's secret key, is shared back with Alice. Alice needs to do a similar process than Bob, obtaining EvalMulAAB (EMAAB) based on EMAB, the final public key and her own secret key. The final combined key for the Mult operation EvalMultFinal (EMF). can be then obtained based on EMAB, EMBAB and EMAAB. This is also introduced in the CC to be considered in cryptographic operations involving multiplications. The final context is shared again with Bob. By this way, both Alice and Bob have updated the CC context with combined keys for the Sum and Mult operations.

The updated context is needed for the encryption, evaluation and decryption processes. Although the encryption and evaluation processes follow anormal homomorphic operation flow, just considering the updated CC, the decryption process changes as it must be executed in three steps as Alice and Bob have different, but linked, private keys. For this purpose, Alice needs to partially decrypt the encrypted result with her own private key while Bob does something equivalent with his own private key. Both of them share their own partially decrypted result with the other. By this way, both of them have the two partially decrypted results that can be merged to obtain the clear result.

5.2.2. SELF-SOVEREIGN IDENTITY

SSI is a concept that provides full and exclusive control over own digital identity [50]. It has been traditionally applied to people, but its utility has been extended to other actors such as IoT devices [51] or datasets [39]. Rather than relying on third parties, such as governments or companies, to validate and manage a digital identity, SSI proposes that each actor can securely own and control his own digital identity. SSI outperforms traditional digital identity management systems [52] in terms of security through the use of digital signatures (usually based on asymmetric cryptography) and verifiable data registries that have been extensively implemented using decentralised blockchain-based networks [53] [54].

The idea behind SSI is to empower actors by giving them greater privacy and control over their digital identity. With SSI, they can decide what information to share, with whom and for what purpose, thus avoiding excessive data collection and the possibility of their personal information being subject to security breaches or misuse by third parties.

As a summary, Figure 101 shows a traditional SSI scenario which involves four main components:

- Issuer. He creates and signs verifiable credentials with identity attributes related to the actor.
- Holder. He locally stores and controls the credentials about himself.
- Verifier. He needs to identify (verify/validate) one or more actor's attributes based on verifiable credentials issued by trusted issuers. This verification is based on validating the holder's provided proof, in where requested claims by the verifier are attested.
- Verifiable Data Registry (VDR). It is a source of truth for public information (Decentralized Identifiers – DID – with public key information for the signatures validation, and schemas of the issued credentials).

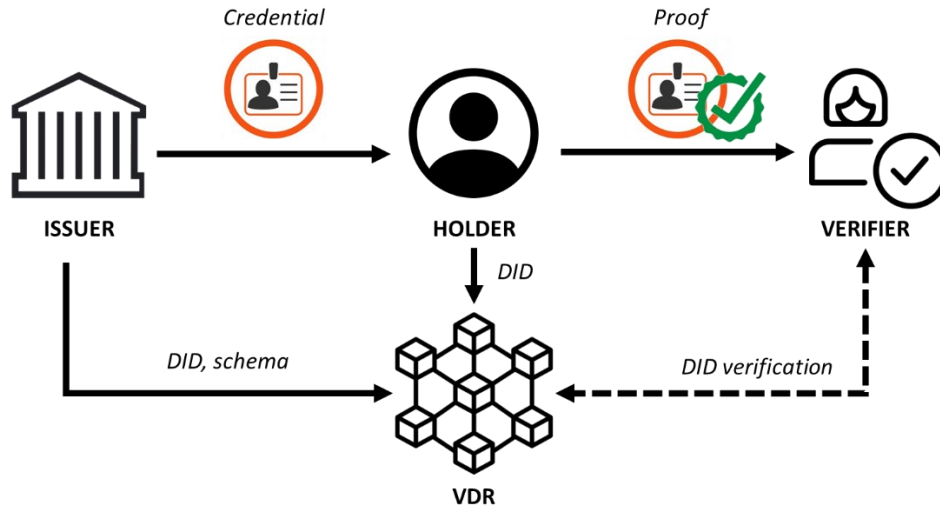


Figure 101: SSI main components

5.3. REQUIREMENTS

In this chapter, the compliance of the SSI-HE ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding with the aforementioned requirements addressed during the development and by the implementation of the SSI-HE and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements have been considered. As the implementation of the SSI-HE progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

User-centred requirements were identified under T2.3. These requirements were reported in D2.1 [1], while they were also utilized in order to extract technical, functional and non-functional requirements, that were also outlined in D2.1. Table 37 below presents the interplay between user-centred requirements and technical requirements, that is the functional and non-functional requirements that address each user-centred requirement. The user-centred requirement ID is used to reference the respective requirement reported in D2.1 [1] while the requirement's description is also provided alongside whether the SSI-HE is currently fully or partially compliant with the user-centred requirement, i.e., indicating whether technical requirements have been identified in order to address the respective user-centred requirement. The ID of the technical requirement, as reported in D2.1 [1], that is addressing each user-centred requirement is also provided in the last column of the table. Additionally, it is possible than more than one technical requirement addresses a single user-centred requirement.

Table 37: SSI-HE - User-centred requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
USR-001	TRUSTEE must enable collaboration among domains and enable access and use of data for secondary use from multiple domains	Partially	Req-SSI-HE-FUNC-1 Req-SSI-HE-NFUNC-1

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

USR-002	TRUSTEE must use standardized methods and clear instructions for correct data entry and formatting	Partially	Req-SSI-HE-FUNC-1 Req-SSI-HE-FUNC-2 Req-SSI-HE-NFUNC-1
USR-013	TRUSTEE should make data provider's data readily available and without the need for them to reformat them	Yes	Req-SSI-HE-FUNC-1
USR-016	TRUSTEE should enable interoperability and integration of data from different data sources	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-020	The user should be able to find out about the validity, quality, and fairness of the data	Yes	Req-SSI-HE-FUNC-1 Req-SSI-HE-FUNC-3
USR-026	TRUSTEE should support data that contain meta-data, text based included.	Yes	Req-SSI-HE-FUNC-1
USR-027	TRUSTEE should include good quality metadata	Yes	Req-SSI-HE-FUNC-1
USR-032	TRUSTEE should enable query and align datasets with other data from different domains	Partially	Req-SSI-HE-FUNC-4
USR-036	TRUSTEE should allow efficient data querying	Yes	Req-SSI-HE-FUNC-4
USR-041	TRUSTEE should allow the process of data without IPR and privacy issues	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-042	TRUSTEE should be compliant with all security requirements and legislations regarding data manipulation and querying	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-046	TRUSTEE should eliminate the correlation of data with personal data	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-048	TRUSTEE should support the ability to automatically add data and include extra metadata fields to these data if needed	Yes	Req-SSI-HE-FUNC-1
USR-057	TRUSTEE should be an efficient and highly secure platform	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-064	TRUSTEE should enable users to provide their AI/ML algorithms for others to use them	Yes	Req-SSI-HE-FUNC-1
USR-084	TRUSTEE could enable fast pre-processing of datasets	Partially	Req-SSI-HE-NFUNC-1
USR-104	TRUSTEE could be a secure platform with waivers from local IRB and privacy policies	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4
USR-107	Through TRUSTEE data providers could be able to avoid pseudonymization and be able to share data	Yes	Req-SSI-HE-FUNC-3 Req-SSI-HE-FUNC- 4

Table 38 presents the preliminary results of the investigation regarding the compliance of the DPIA with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether DPIA complies with the requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 38: SSI-HE - Functional and non-functional technical requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
Req-SSI-HE-FUNC-1	This tool includes the characteristics definition (credential issuance), check (credential storage and visualization) and verification (credential validation).	Yes	Three tools have been developed for the three operations related to data characterization: issuer, checker and verifier.
Req-SSI-HE-FUNC-3	Extend the data characterization tool functionality allowing private metadata/characteristics to be validated in a confidential way	Yes	Homomorphically encrypted attributes are considered.
Req-SSI-HE-FUNC- 4	Allow encrypted searches over confidential datasets metadata/attributes for specific queries	Yes	Search over homomorphically encrypted data are executed on the verifier.
Req-SSI-HE-NFUNC- 1	All type of metadata/attributes should be supported	Partially	Numeric and string attributes have been considered. Different kind of strings need to be considered yet.

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

5.4. ARCHITECTURE

This section gathers the architecture designed for the SSI-HE component. It extends the one presented in D2.1 [1].

5.4.1. HIGH LEVEL

The high-level architecture of the SSI-HE component, shown in Figure 102, has been updated (and extended) from that presented in D2.1. As a summary, the SSI-HE component is formed by a common SSI Blockchain network and as many SSI issuers (certification authorities), and SSI holders (data providers) as needed inside TRUSTEE. TRUSTEE as a platform will always be the SSI verifier in this scenario.

Each of the SSI participants will be composed of an SSI Agent backend to perform all the SSI-related operations, a HE orchestrator backend to orchestrate all the operations related to FHE (encryption, decryption, homomorphic searches, etc.) and a frontend controller for the user to interact with the backends.

As a new feature, an InterPlanetary File System (IPFS) network has been included in the SSI-HE component to allow dealing with large encrypted attributes. Homomorphically encrypted attributes can sometimes be larger than the maximum size allowed by SSI credentials. To avoid limitations, the encrypted attributes will not be directly included on the SSI credential. On the contrary, they will be uploaded to the IPFS network (there are no size limitations) and the credential will include the IPFS endpoint where the encrypted attribute is stored.

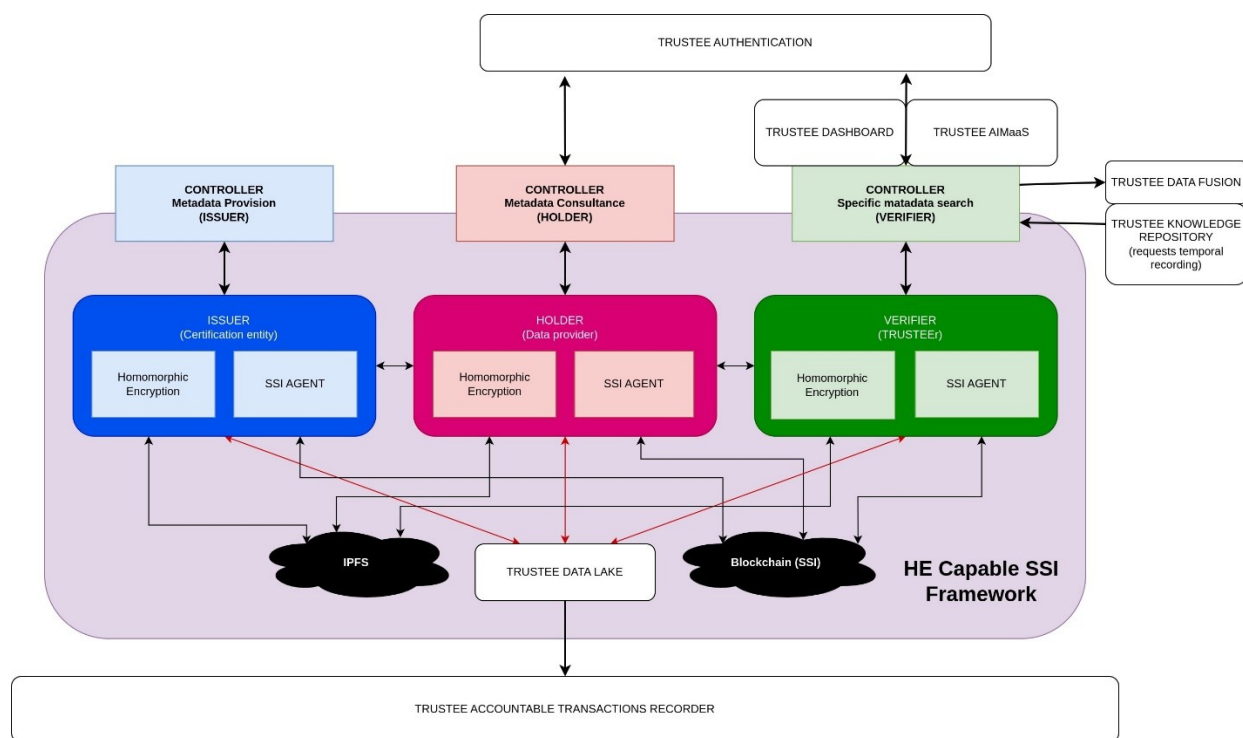


Figure 102: SSI-HE high level architecture

Taking everything into consideration, the SSI-HE component is divided into five main subcomponents:

- **Issuer:** this subcomponent provides the encrypted metadata, uploads it to IPFS and generates the verifiable credentials including the IPFS endpoints. It should refer to the data owner itself (for example, in a healthcare use case, it could be the patient the data refers to) or other trusted certification entity (for example, a government, etc.). In TRUSTEE, TECNALIA will simulate this actor for validation purposes.
- **Holder:** this subcomponent holds and manages the verifiable credentials and creates verifiable proofs when needed. It should refer to the TRUSTEE data providers (following with the healthcare use case, it could be the hospital itself).
- **Verifier:** this subcomponent asks for some specific attributes of the metadata, receives verifiable proofs of the required attributes, downloads the encrypted attributes from IPFS and homomorphically executes a search over the confidential attributes. It refers to the TRUSTEE platform itself allowing a data consumer who needs datasets with specific characteristics (metadata) to filter by valid providers. The specific characteristics of the datasets are what he searches over the encrypted attributes.
- **Blockchain (SSI):** this is the SSI Blockchain network needed for the correct and secure operation of SSI technology. Public keys and schemas will be securely recorded in order guarantee verifiable credentials and verifiable proofs validity.

- IPFS: this refers to the IPFS network needed for storing large homomorphically encrypted attributes avoiding size limitations.

5.4.2. LOW LEVEL

Figure 103 describes the low level architecture of the SSI-HE component, in which the internal components of each of the SSI subcomponents are shown.

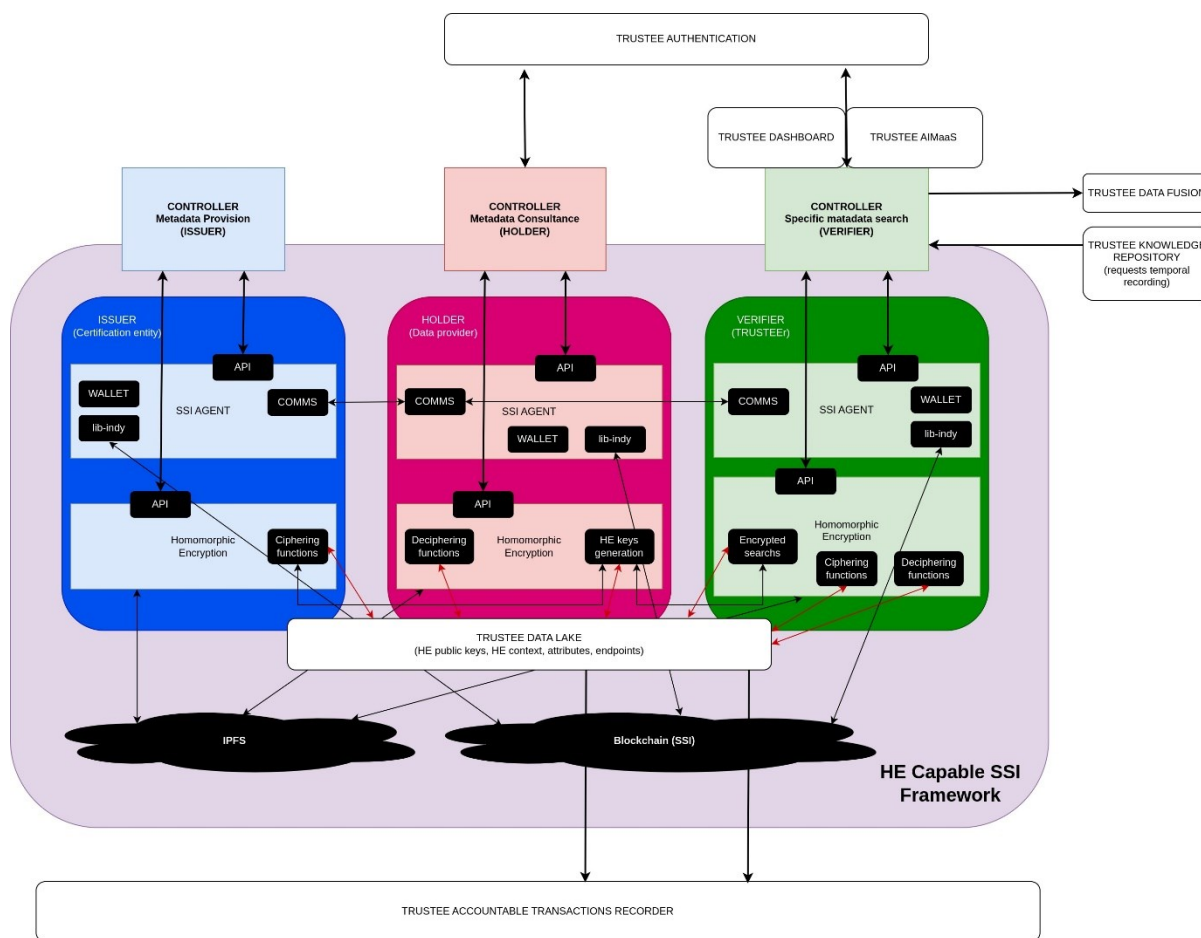


Figure 103: SSI-HE low level architecture

As mentioned above, the three types of SSI participants will be composed of two main elements: the SSI agent and a module dedicated to homomorphic cryptographic operations. Both modules will expose an API where they will receive instructions which will be orchestrated through the controller.

Looking at the functionalities of each subcomponent, the SSI agent has been designed in such a way that all the roles share the same internal elements:

- API: It is the interface that is exposed and that allows the communication of the agent with the controller.
- Wallet: It is the tool that allows working with digital identities in SSI, mainly used to store verifiable credentials and private crypto material in a secure way.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- Lib-indy: It is the library that provides all the functionality needed by SSI technology.
- Comms: HTTP based communications between different SSI agents are considered.

The homomorphic encryption module, however, contains different internal elements depending on the specific role.

- API: It is the interface that is exposed and to allow the communication of the HE module with the controller. This is common to all roles (issuer, holder and verifier).
- Ciphering functions: Homomorphic text encryption functions. This is needed in the issuer for encrypting the attributes to be included in the verifiable credentials; and in the verifier to encrypt the required attribute value for the search.
- Deciphering functions: Homomorphic text decryption functions. This is needed in the holder for internal validation; and in the verifier for the decryption of the result of the search.
- HE key generation: Key generation functions with a Threshold FHE cryptographic context (two private keys and the public key are generated).
- Encrypted search: Execution function of the homomorphic search operation of an encrypted attribute value over encrypted attributes from verifiable credentials.

Each controller frontend will be designed according to the role of each participant. The functionalities to be provided by each one are:

- Issuer's controller: Create an SSI scheme, create an SSI credential definition, issue an SSI credential, homomorphically encrypt an attribute, store information in IPFS.
- Holder's controller: Save an SSI credential, create SSI proof responses, generate HE crypto context, generate Threshold FHE keys, make partial decryption and merge partial decryptions, retrieve information from IPFS.
- Verifier's controller: Create an SSI proof request, verify a SSI proof response, retrieve information from IPFS, encrypt the value to be search, perform homomorphic search evaluation, make a partial decryption and merge partial decryptions.

Finally, the SSI-HE component also interacts with other TRUSTEE components:

- DA Although, the issuer and holder controllers will include their own graphical interface, the verifier (TRUSTEE platform) will be integrated in the DA. It should allow the graphical input of attributes to search (for example, the value Spain for location attribute) and the graphical output of the obtained data providers fulfilling the requested attributes.
- AIMaaS: It refers to a similar graphical interface as the DA., but for AI purposes (training or inference). It should allow the graphical input of attributes to search (for example, the value Spain for location attribute) and the graphical output of the obtained data providers fulfilling the requested attributes.
- AM: Only authenticated users can use the SSI-HE component (for data providers – holder -and data consumers – verifier -). Certification authorities -issuers - do not require to be part of TRUSTEE.
- ATR: The important events of the SSI-HE operation (those related to the holder and verifier actors) will be logged.
- DL: For accessing the list of attributes available for searching functionality, the HE public keys and HE context and the endpoints of the valid data providers. Some of the information needs to be securely stored (authentication should be required) while other, such as the HE public keys and HE context should be publicly available to all potential issuers that do not need to be part of TRUSTEE platform.
- KR: For temporal recording and obtaining of previous requests results (in case they are needed in a short time). Only the verifier needs access to the KR.

- HEDF: The SSI-HE provides the result of his operation to the HEDF component (the data providers endpoints fulfilling the data consumers requirements). Only the verifier needs access to the HEDF.

All the external interfaces definitions between SSI-HE and other TRUSTEE components have been gathered in D2.1 [1].

5.4.2.1. HOMOMORPHIC SEARCH

The algorithm to perform the search operation in the homomorphic plane will be based on Fermat's Little Theorem (FLT) [55]. Text data must be encoded in ASCII before encryption, since homomorphic cryptography does not consider homomorphisms with text values, only arithmetic operations with numeric values. Each of the strings will be encrypted in a single ciphertext, so that the comparison will be executed directly between them.

5.4.2.2. ATTRIBUTE VERIFICATION PROCESS

In the SSI-HE operation, the verifier needs to verify if a specific value fits with one of the encrypted attributes included in a verifiable credential. During the credential verification process, the verifier requests a proof of the attributes he needs to know and, following the traditional SSI verification process, the holder sends a verifiable proof with the IPFS endpoint of the requested attribute. The verifier then collects the encrypted attribute from the IPFS network and applies the homomorphic search algorithm to check if the specific value is contained in the encrypted attribute.

The result obtained from this homomorphic search operation is also encrypted. In a traditional scenario, the holder is the only one who can decrypt the result since he is the only one who has the private key needed to obtain the clear result. However, the result needs to be known by the verifier, but the verifier does not trust the holder. To avoid this problem, Threshold FHE has been considered. The process to be followed is shown in Figure 104.

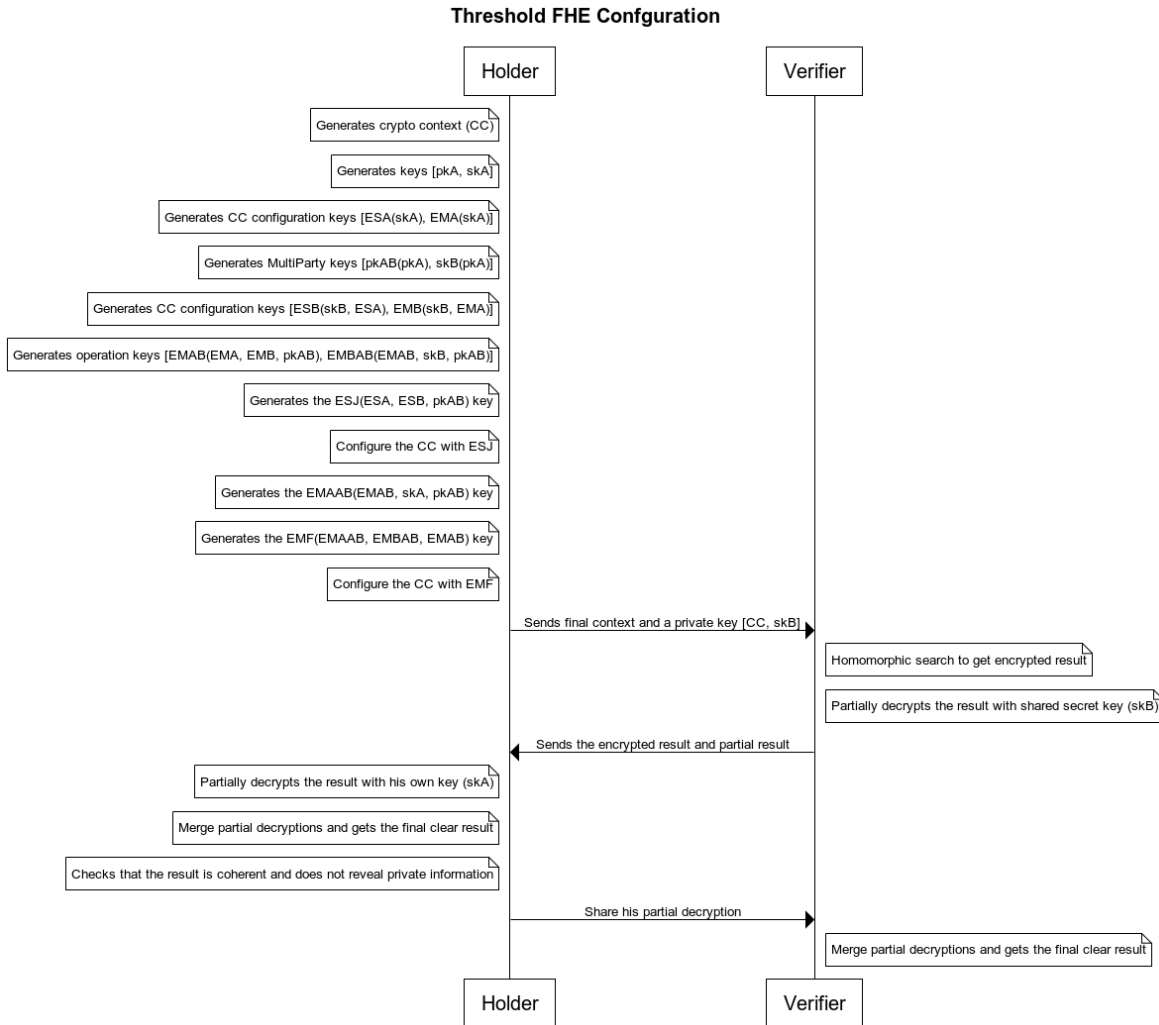


Figure 104: Threshold FHE in the SSI-HE component

In this case, each holder (data provider) will create the CC, the public key and the two secret keys needed for the Threshold FHE operation (as explained in Background section). One of the secret keys will be for himself (holder) (skA) and the other one (skB) will be shared with TRUSTEE Platform (verifier). The secure decryption process requires that the verifier knows the updated CC (after the Threshold FHE generation) and his secret key (skB).

Once this information is shared, the verifier uses it to make a first partial decryption of the result. He sends the partially decrypted result and the original encrypted result (without any decryption process being applied) to the holder. Once the holder receives this information, he uses his secret key to make the second partial decryption process. He can merge both partially decrypted results (the one received from the verifier and the one he has just obtained) and get the clear result. In case of being a coherent result, in other words, it is a true/false value (as expected in a search operation), he sends his second partially decrypted result to the verifier, who can merge it with his own first partially decrypted result and obtain the clear result, in other words, the result of the search operation.

This process should be repeated for each data provider.

5.4.2.3. HE FRAMEWORK

OpenFHE library has been chosen for the SSI-HE component due to the following reasons:

- **Homomorphic scheme for integer values:** The idea is to encode any input text in ASCII before being encrypted, so integer values will be needed. For this reason, a FHE library that implements the BGV homomorphic scheme, such as OpenFHE, is recommended. This specific scheme is known to be efficient and safe for arithmetic operations on integer values, which makes it a suitable choice for this scenario.
- **Threshold FHE for secure verification:** To ensure secure verification of the final result, it is necessary to perform Threshold FHE. This feature allows splitting the encryption keys and homomorphic computations among several participants (in this case, holder and verifier), which increases security and avoids having to trust on a third-party. OpenFHE offers this functionality, which provides greater confidence in the integrity of the results obtained during the verification of SSI credentials.
- **Open-source licence:** OpenFHE is an open-source library, which means that its source code is available for study, modification and distribution. This facilitates transparency in the research environment and allows the library to be adapted to the specific needs of the proposed architecture in combination with SSI.
- **Community and support:** OpenFHE is a library with an active community of developers and users, as well as continuous support, which is beneficial for solving technical problems, receiving updates and improvements, and sharing knowledge with other researchers or professionals who use this library.

5.4.2.4. COMPLETE OPERATION FLOW

The SSI-HE operation is composed of two main phases:

- Metadata generation
- Metadata validation
- Result sharing

METADATA GENERATION

This phase refers to the generation and storage of verifiable credentials with IPFS endpoints where homomorphically encrypted attributes of a specific data provider dataset are stored. This phase involves the issuer and holder actors. Additionally, the SSI Blockchain network and the IPFS are also needed. The public information (public key, CC) is shared between holder and issuer through the DL. Finally, the AM (for users authentication) and ATR (for logging reasons) are also required for extended functionality. This process should be followed by all the data providers as a secure/confidential way to characterize their datasets.

Figure 105 shows the sequence flow for the metadata generation phase.

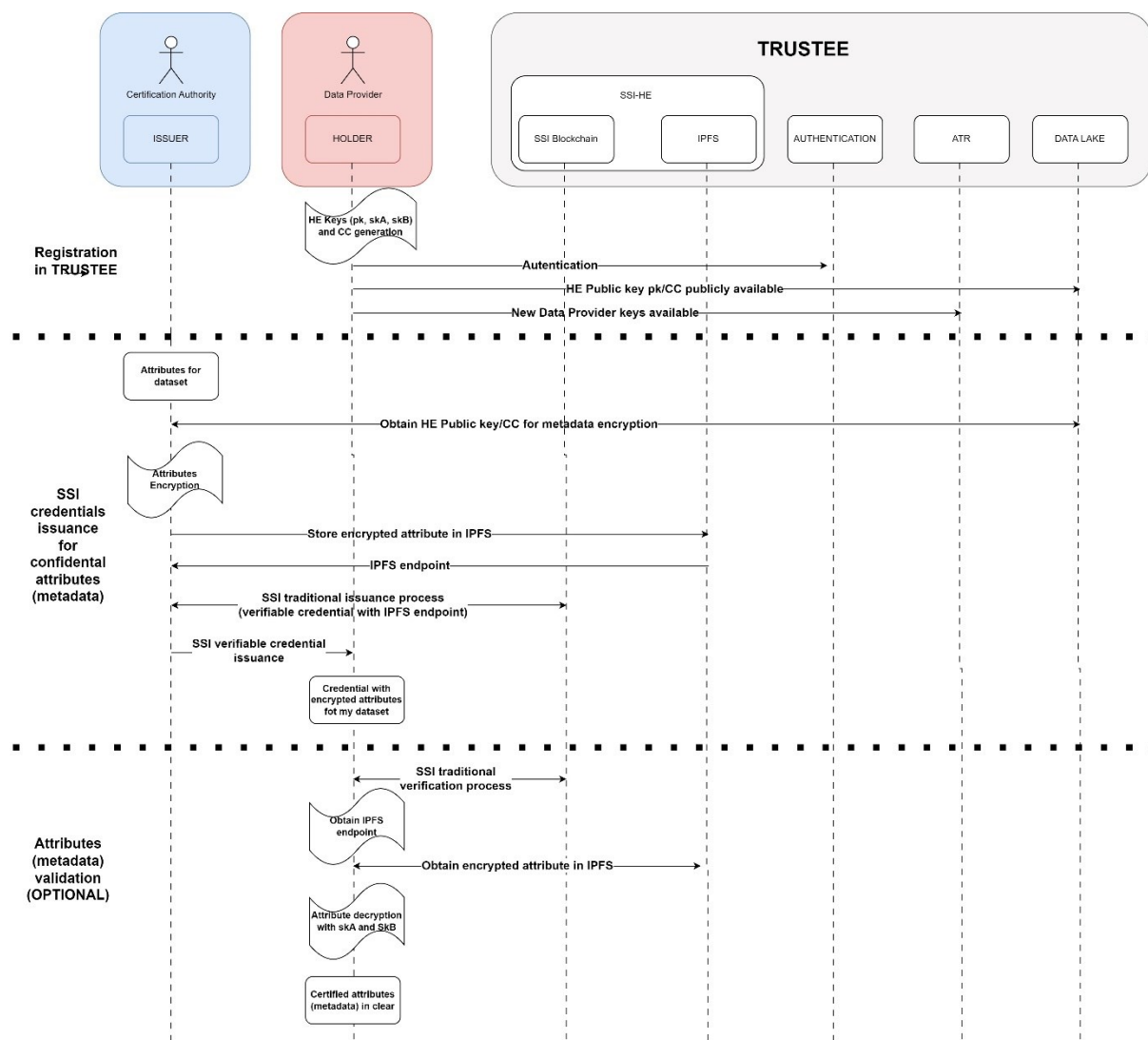


Figure 105: SSI-HE metadata generation phase

This process starts when a new data provider is registered in TRUSTEE. TRUSTEE requires the data provider to have an HE public key and HE CC that can be used for the subsequent metadata validation process. This information should have been created by the data provider before registering in TRUSTEE. Taking this into consideration, the process is as follows:

- The data provider generates the necessary HE keys (and HE CC) for their dataset's attributes encryption. The keys are locally generated following the Threshold FHE scheme, so one public key (pk) and two secret keys (skA, skB) are generated (and the HE CC).
- The data provider authenticates in TRUSTEE to be able to store information in the DL.
- The public key (pk) and the HE CC are stored in a public part of the DL.
- This action is logged in the ATR.

At any time, a certification authority should be able to issue verifiable credentials about a data provider's dataset attribute (metadata). For this purpose:

- v. The certification authority needs to obtain the data provider's public key (pk) and HE CC from the public part of the DL.
- vi. With this information, the certification authority encrypts the data provider datasets attributes (metadata).
- vii. This encrypted (confidential) attributes are stored in IPFS as their size is usually too large for SSI credentials.
- viii. The IPFS endpoint where the encrypted attribute has been stored is obtained.
- ix. The SSI traditional operational is followed by the certification authority for the generation of the verifiable credential with IPFS endpoints.
- x. This SSI verifiable credential is then issued (and shared) to the data provider.
- xi. This SSI verifiable credential is stored by the data provider.

Optionally, the data provider could verify the attributes that have been certified in the received SSI verifiable credential. For this purpose:

- xii. The SSI traditional operational is followed by the data provider for the validation of the verifiable credential with IPFS endpoints.
- xiii. The IPFS endpoint is obtained from the SSI credential.
- xiv. The encrypted attribute is obtained from the IPFS endpoint.
- xv. The data provider can decrypt the attributes included in the SSI credential as he already has the two secret keys needed for the Threshold FHE decryption process.
- xvi. After the attributes' decryption, the data provider can see the attributes the certification authority has issued about his datasets.

METADATA VALIDATION

This phase refers to the generation and validation of verifiable proofs with IPFS endpoints where homomorphically encrypted attributes of a specific data provider's dataset are stored, in order to check if a specific value of an attribute fits with the attributes certified by the data providers. This phase involves the holder and verifier actors. Additionally, the SSI Blockchain network and the IPFS are also needed. The information is shared between holder and verifier through the DL (public key, CC, partial private key). Additionally, the DA or AIMaaS are considered as the input point (to define the attribute to be searched). Finally, the AM (for users authentication) and ATR (for logging reasons) are also required for extended functionality.

Figure 106 shows the sequence flow for the metadata validation phase.

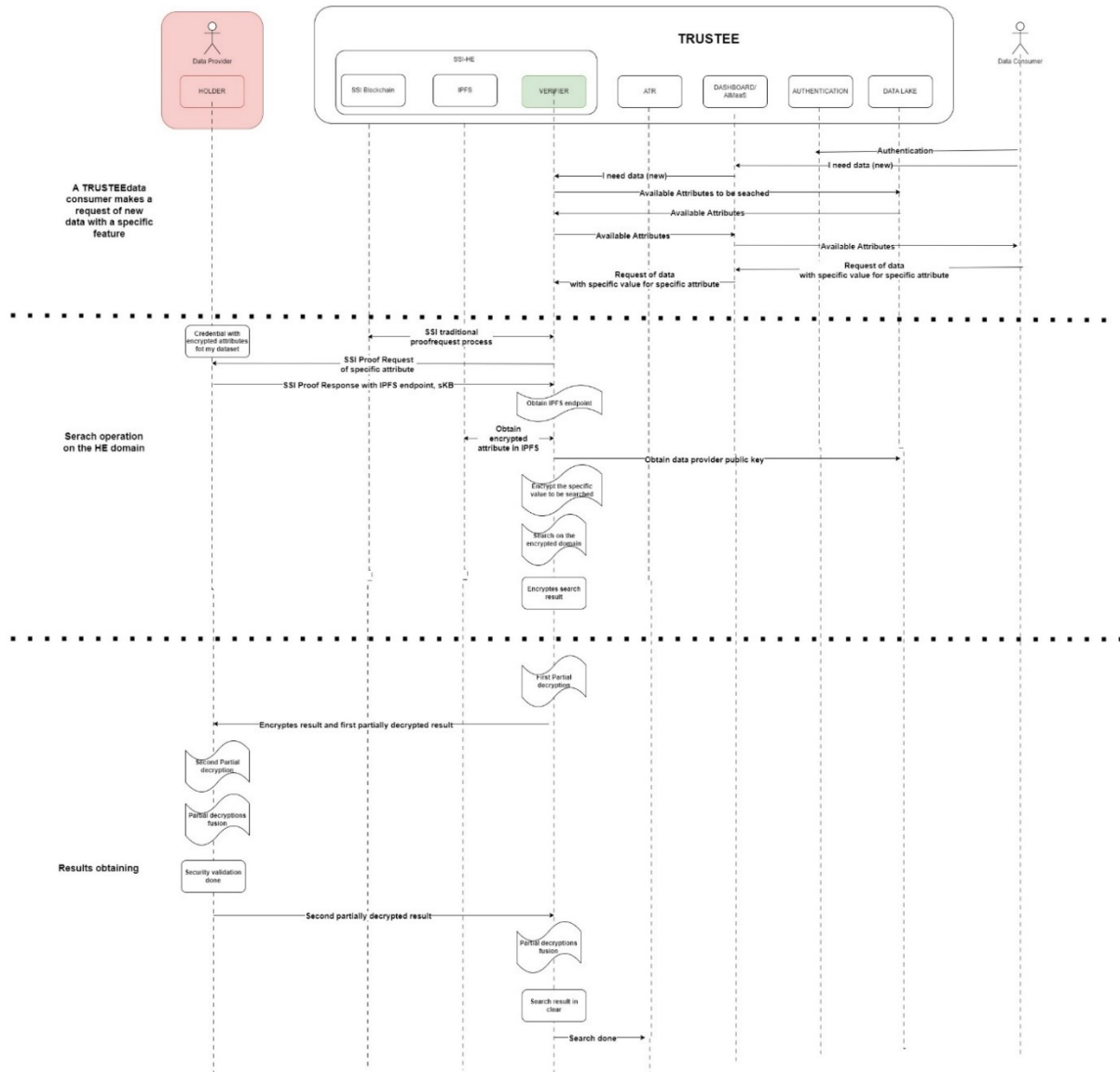


Figure 106: SSI-HE metadata validation phase

This process starts when a TRUSTEE data consumer makes a new request of datasets/models with a specific characteristic (imagine for example, data with location Spain). For this purpose:

- The data consumer makes a new request of new data needed through the DA (for data) or AIMaaS (for training purposes).
- The TRUSTEE verifier obtains the available attributes that can be searched from the DL in order to show them to the data consumer.
- The data consumer specifies the specific attribute to search and the desired value for this attribute.

With this information, the TRUSTEE Verifier can start talking with all registered data providers in order to identify if they fulfil the data consumer requirement for a specific attribute. For this purpose:

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- iv. Following a traditional SSI proof request process, the TRUSTEE verifier makes a SSI proof request to a data provider.
- v. The data provider sends a proof response based on the previously received SSI verifiable credentials (from the metadata generation phase). Additionally, it sends the secret key skB (generated in the previous metadata generation phase), TRUSTEE will afterwards need for the Threshold FHE decryption process.
- vi. The TRUSTEE verifier receives the SSI proof response and obtains the IPFS endpoint where the encrypted attribute is.
- vii. The encrypted attribute is obtained from the IPFS endpoint.
- viii. At the same time, the TRUSTEE verifier obtains the data providers public key from the DL.
- ix. With the public key, the TRUSTEE verifier can encrypt the specific value the data consumer has requested in his request.
- x. The TRUSTEE verifier will execute the search algorithm of the encrypted value over the encrypted attribute pulled from IPFS.
- xi. The result of the search is obtained. It is also encrypted.

Finally, the result of the search needs to be also decrypted following the Threshold FHE operation to provide enhanced security. For this purpose:

- xii. The TRUSTEE verifier executes the first partially decryption process of the result with the partial secret key skB received in step 5.
- xiii. The TRUSTEE verifier sends the encrypted result (from step 11) and the first partially decrypted result (from step 12) to the data provider.
- xiv. The data provider executes the second partially decryption process of the encrypted result with the other partial secret key (skA) he has from the previous metadata generation phase.
- xv. The data provider combines both partially decrypted results (the one received from TRUSTEE verifier in step 13 and the one just calculated by him in step 14) in order to verify if the result received from the data consumer makes sense with the expected result (true/false result of a search operation).
- xvi. If the security validation is successful, the data provider sends his second partially decrypted result to the TRUSTEE verifier.
- xvii. This partially decrypted result from the data provider is combined with the one previously obtained by the TRUSTEE verifier in step 12, finalizing the Threshold FHE decryption process. By this way the data consumer obtains the result in clear (but not having accessed the attributes from the data owner datasets). This is the final result of the searching process.
- xviii. This action is logged in the ATR.

RESULT SHARING

This phase refers to the result management, sharing it with Data Consumer through the DA, with the HEDF for being able to start the execution process over identified datasets, and with the KR in case both the data provider and data consumer have agreed so for greener computation and better performance. Figure 107 shows the sequence flow for the metadata validation phase.

- i. If the result is positive, that means that the data provider meets the requirement from the data consumer. In this case, TRUSTEE verifier obtains the associated data endpoint for this specific data provider from the DL.
- ii. This specific endpoint is then shared with the HEDF to start the fusion process.
- iii. The endpoint is also shared with the data consumer through the DA.

- iv. This specific endpoint is stored in the KR in case the same request is done. For completing this step, Data provider and Data consumer should consent the recording of the result, and both are responsible to indicate how long the result will be kept in the KR.

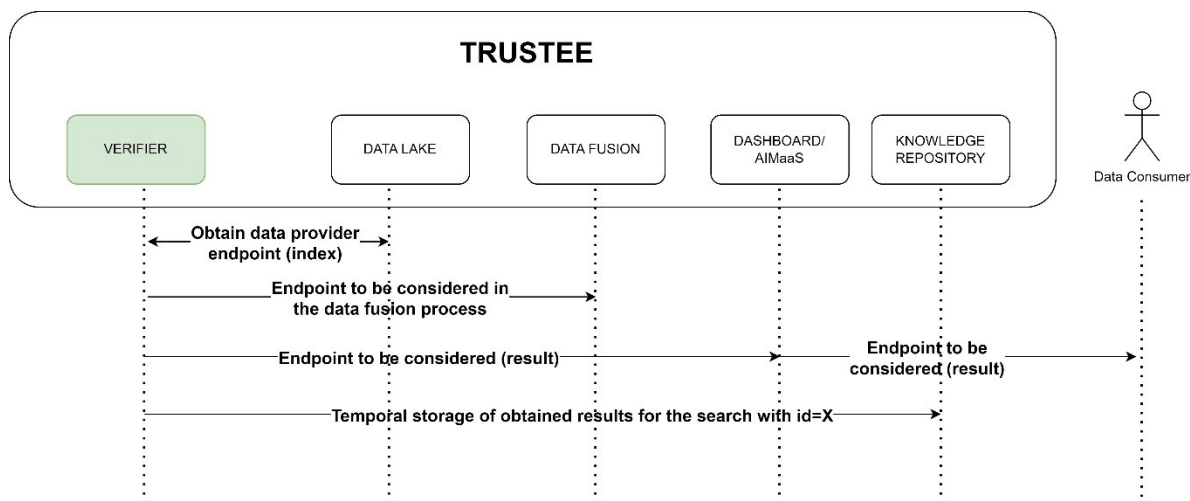


Figure 107: SSI-HE result sharing phase

Figure 108 shows the sequence flow for the result sharing phase when the request done from the data consumer already exists in the KR because it has been previously done.

This process starts with the data consumer indicating that the is in a previously recorded request. In this case:

- i. The TRUSTEE verifier obtains the available requests from the KR.
- ii. The available requests are shown to the data consumer through the DA or AIMaaS.
- iii. The data consumer selects one of the available requests.
- iv. The TRUSTEE Verifier obtains the data provider endpoints associated to that specific request from the KR.
- v. The endpoints are then shared with the HEDF to start the fusion process.
- vi. The validity of the specific request is restarted again.

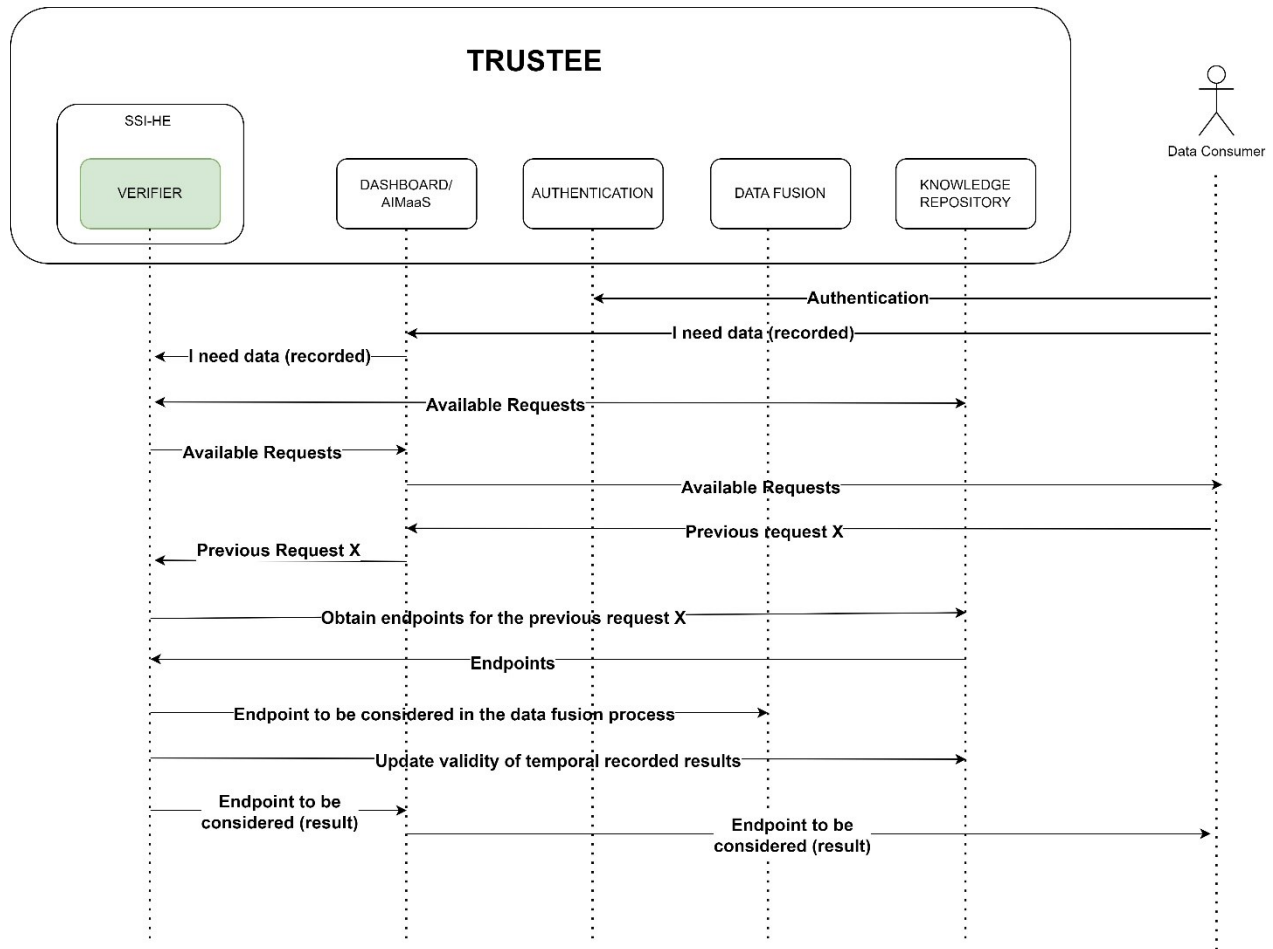


Figure 108: SSI-HE result sharing phase with existing results

5.4.2.5. MAIN FUCNTIONALITIES

Taking everything into consideration, the main functionalities of the three components are:

ISSUER

The main functionalities for the issuer actor are:

- Define SSI schemas and credential definitions related to the available attributes for the different use cases in TRUSTEE (or others).
- Obtain the HE public key and CC of the data provider from the DL.
- Homomorphically encrypt an attribute with a public key.
- Store the encrypted attribute in IPFS (as a file) and obtain the associated IPFS endpoint.
- Create SSI credentials with IPFS endpoints as attributes following suitable SSI schemas and credential definitions.

HOLDER

The main functionalities for the holder actor are:

- Generate HE public and secret keys for a Threshold FHE, as well as the CC.
- Authenticate in the TRUSTEE platform.
- Store the public information (public key and CC) in the DL.
- Inform the TRUSTEE Accountable Transactions Manager about the new keys registered in the DL.
- Store SSI credentials received from the issuer.
- Obtain IPFS endpoints from SSI credentials.
- Retrieve the HE encrypted attribute files from IPFS using the IPFS endpoint.
- Partial decryption with both secret keys of the Threshold FHE context.
- Merge partial decryption results to verify what the issuer has certified (internal verification and final result security verification).
- Make suitable SSI proof responses based on stored SSI credential with IPFS endpoints.

VERIFIER

The main functionalities for the verifier actor are:

- Receive new attributes validation requests through the DA or the AIMaaS components.
- Obtain the list of available attributes to be searched from the DL.
- Make an SSI proof request of a specific attribute to the data provider (holder).
- Receive an SSI proof response from the data provider.
- Receive a Threshold FHE secret key from the data provider.
- Obtain the IPFS endpoints from the SSI proof response.
- Retrieve the HE encrypted attribute files from IPFS using the IPFS endpoint from the proof response.
- Obtain the public key and CC of the data provider from the DL.
- Encrypt the specific value to be searched with the HE public key.
- Execute the homomorphic searching operation over the encrypted attributes and obtain an encrypted result.
- Partial decryption with one of the secret HE keys of the Threshold FHE context.
- Receive the second partially decrypted result from the data provider.
- Merge partial decryption results to get a final clear result.
- Inform the ATR about the attributes validation process end.
- If the result is positive, obtain the associated endpoints from the DL.
- Pass these endpoints to the data consumer and the HEDF component.
- The result of the search can be also recorded on the KR for fast response for the same request.
- Obtain the list of available previous requests from the KR.
- Obtain the list of endpoints associated to a previous request from the KR.

5.5. IMPLEMENTATION

The focus during this first implementation phase has been on the design and theoretical analysis of the way to properly join SSI and HE technologies that could be useful for TRUSTEE. Additionally, a first isolated and standalone prototype of the SSI-HE component for the three subcomponents (with no communications with other TRUSTEE components at the moment) has been implemented, focusing mainly on a basic complete process (including metadata generation and metadata validation phases).

Additionally, the three subcomponents have not been distributed yet; all of them have been locally deployed sharing different volumes in order to easily share the needed information (HE public key, HE context, data and results).

5.5.1. FUNCTIONAL DESCRIPTION

5.5.1.1. ARCHITECTURE

Taking everything into consideration, Figure 109 shows the functional components of the first implementation of the SSI-HE component.

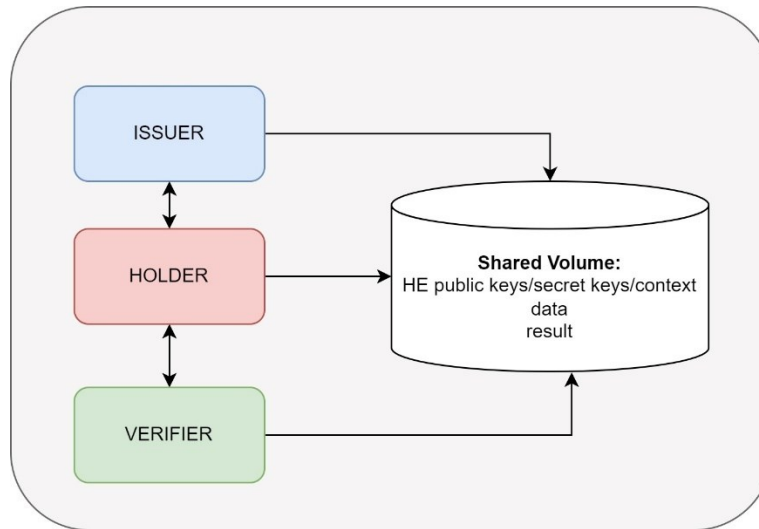


Figure 109: SSI-HE first implementation functional components

In detail, the functionalities of the first implementation in comparison with the complete functionalities of the layer, defined in section Architecture, are gathered in Table 39.

Table 39: SSI-HE component first implementation functionalities

Functionality	Coverage Status	Current implementation
ISSUER		
Define SSI schemas and credential definitions related to the available attributes for the different use cases in TRUSTEE (or others).	Partially	The possibility to create different SSI schemas and credential definitions has been implemented. However, just one schema is considered by the moment. Discussions with the TRUSTEE use cases are needed.
Homomorphically encrypt an attribute with a public key.	Yes	The issuer is capable of homomorphically encrypting an attribute with the public key generated by the holder.
Store the encrypted attribute in IPFS (as a file) and obtain the associated IPFS endpoint	Yes	The issuer is capable of storing an encrypted attribute in IPFS and obtain the associated IPFS endpoint to be included in the SSI credential.
Create SSI credentials with IPFS endpoints as attributes following suitable SSI schemas and credential definitions.	Yes	The issuer is capable of creating and sharing the suitable verifiable credential.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

HOLDER		
Generate HE public and secret keys for a Threshold FHE, as well as the CC.	Yes	OpenFHE Threshold FHE keys generation and CC generation has been implemented.
Store the public information (public key and CC) in the DL.	No	Shared volumes are considered. No interaction with the DL has been implemented yet. To be addressed in future versions.
Inform the TRUSTEE Accountable Transactions Manager about the new keys registered in the DL	No	No interaction with the ATR component has been considered yet. To be addressed in future versions.
Store SSI credentials received from the issuer	Yes	Received SSI credentials are stored in the local wallet of the SSI agent of the holder.
Obtain IPFS endpoints from SSI credentials.	Yes	The attributes from the SSI credential are obtained.
Retrieve the HE encrypted attribute files from IPFS using the IPFS endpoint.	Yes	The holder is capable of obtaining the encrypted attribute using the IPFS endpoint obtained from the SSI credential.
Partial decryption with both secret keys of the Threshold FHE context.	Yes	The holder knows the two secret keys (as he has generated them) and is capable of making the first and second partially decryption processes.
Merge partial decryption results to verify what the issuer has certified (internal verification and final result security verification).	Partially	The internal verification has been implemented but the final result security verification process has not been implemented yet.
Make suitable SSI proof responses based on stored SSI credential with IPFS endpoints.	Yes	The holder is capable of making proof responses including the requested attributes (as IPFS endpoints).
VERIFIER		
Make an SSI proof request of a specific attribute to the data provider (holder).	Yes	The verifier is capable of making proof requests of a specific attribute.
Receive an SSI proof response from the data provider.	Yes	The verifier is capable of receiving proof responses from the data provider (holder).
Obtain the IPFS endpoints from the SSI proof response.	Yes	The attributes from the SSI credential are obtained.
Retrieve the HE encrypted attribute files from IPFS using the IPFS endpoint from the proof response.	Yes	The verifier is capable of obtaining the encrypted attribute using the IPFS endpoint obtained from the SSI credential.
Encrypt the specific value to be searched with the HE public key.	Yes	The verifier is capable of homomorphically encrypting a value with the public key generated by the holder.
Execute the homomorphic searching operation over the encrypted attributes and obtain an encrypted result.	Partially	A limited searching functionality has been implemented using OpenFHE. This functionality has still to be extended.

Partial decryption with one of the secret HE keys of the Threshold FHE context.	Yes	The verifier is capable of making the first partially decryption process.
Merge partial decryption results to get a final clear result.	Yes	The verifier is capable of merging the two partially decrypted results, obtaining the final decrypted result (result in clear).

5.5.2. OPERATIONAL DESCRIPTION

Figure 110 describes the operation flow in a sequence of the first implementation of the SSI-HE component, considering that all the components have been deployed in the same machine and information exchange is done through shared volumes. Additionally, interaction with other TRUSTEE components has been investigated; however, it is still a work in progress and further details will be provided in upcoming versions of this document, namely D4.2 and D4.3. For this reason, all the information to be shared between components is done through the use of shared volumes.

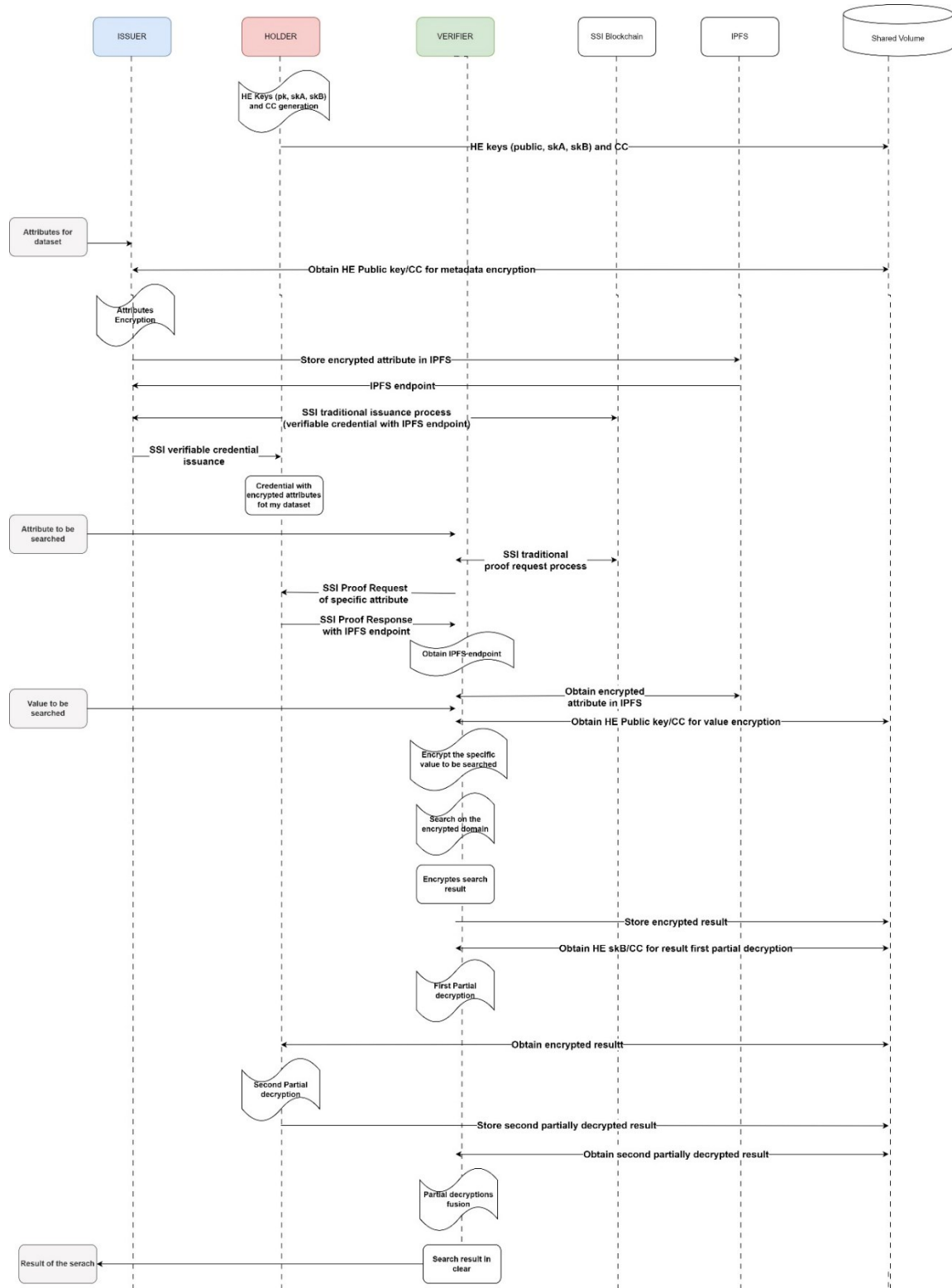


Figure 110: SSI-HE first implementation operation flow

5.5.3. TECHNICAL DESCRIPTION

5.5.3.1. IMPLEMENTATION DETAILS

Figure 111 shows the technical scheme of what has been implemented for the SSI-HE component up until the development of this first version of this live document.

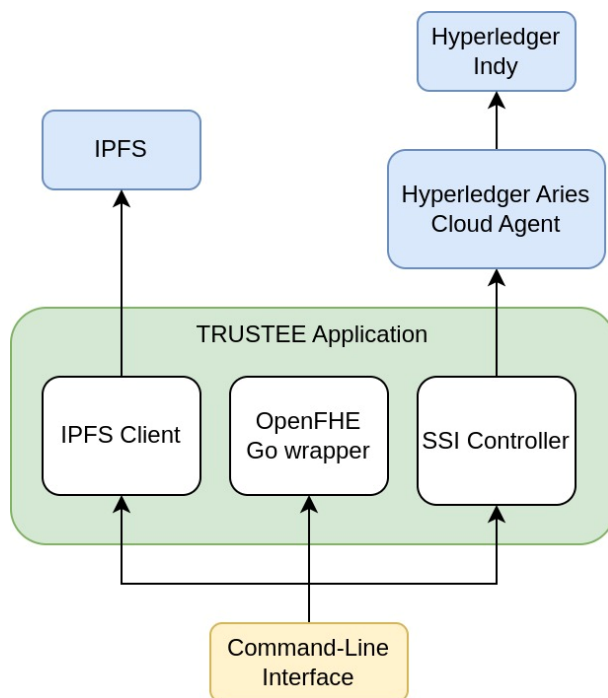


Figure 111: SSI-HE first implementation details

As mentioned, the first implementation has been done to validate the main functionality of the SSI-HE component. Most of the subcomponents require integration with other TRUSTEE components which are still under development, that is why this preliminary implementation has been done exclusively working locally.

In any case, the components in Figure 111 are explained below to better understand how the tool works:

- IPFS. It is a distributed and decentralised storage system. It is responsible for storing and retrieving files using content-based addressing technology.
- Hyperledger Indy. It is an SSI blockchain network, it is used in autonomous and sovereign identity systems. It functions as a distributed ledger that stores and manages identity information.
- Hyperledger Aries Cloud Agent. It is a server used mainly to provide a set of tools and libraries for the development of digital identity solutions and data exchange systems in an SSI network, e.g., issuing, holding or verifying verifiable credentials.
- TRUSTEE Application, which is composed by:
 - SSI Controller. It is a component developed in Go that allows users to interact with the functionalities provided by Hyperledger Aries as it exposes an API that can be consumed from the outside, for example, from a command line interface.

- OpenFHE Go Wrapper. It is a component that provides all the necessary functionalities to perform the cryptographic operations of the OpenFHE library such as generating homomorphic encryption keys or encrypting and decrypting ciphertexts.
- IPFS client is an agent used to mediate with the IPFS network.
- Command-Line Interface (CLI). It is an interface that allows users to interact with a system or application via commands. It provides users to perform operations related to the consumption of the APIs exposed by the SSI controller, OpenFHE wrapper and the IPFS client to execute specific functions on the system and facilitating the administration and use of all the components.

5.5.3.2. INPUT/OUTPUT DATA MODELS

Some details about the expected input and output data:

- Input data: it is text provided through the command line with a limitation of 65 536 amount of characters. In this case, it refers to both the searchable attribute (for the issuer) and the value to be searched (for the verifier). These data are pre-processed and ASCII-encoded so that they are understandable by the homomorphic search algorithm.
- Output data: It is an integer that represents the number of characters that are different between the input data. It refers to the final result of the search; if the value is 0, the comparison executed in the search algorithm is true.

5.5.3.3. API

Table 40 gathers the main API endpoints of the first implementation of the SSI, HE and IPFS subcomponents in the SSI-HE component.

Table 40: SSI-HE component first implementation API details

Function	Input	Output
OpenFHE Wrapper (HE)		
GenerateKeys	-	SharedPublicKey, holderPrivateKey, verifierPrivateKey
Cipher	InputData	Ciphertext, metadata
searchThreshold	searchableTextMetadataFile,	
DeCipher	searchableTextCipherTextFile,	
deCipherFusionVectorVectorInt64	textToSearchMetadataFile,	
IPFS Client		
AddFile	fileName	IPFS_CID
RetrieveFile	IPFS_CID	fileName
SSI Controller		
RegisterSchema	Name, version, attributes	SchemaID
GetSchema	SchemaID	schema

CreateCredentialDefinition	schemaID	credentialDefinitionID
GetCredentialDefinition	credentialDefinitionID	credentialDefinition
IssueCredentialV2	SchemaID, IssuerDID, credentialDefinitionID, publicDID, AttributeValues	CredentialExchangeID
GetCredentialExchangeV2	CredentialExchangeID	CredentialExchangeRecord
PresentationV2SendRequest	schemaID, credentialDefinitionID	PresentationExchangeID
GetPresentationV2ExchangeByID	PresentationExchangeID	presentationExchange

5.5.4. TECHNICAL DETAILS

Table 41 gathers some technical details of the first implementation phase.

Table 41: SSI-HE component first implementation technical details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	Linux (Ubuntu)
Database and databases tools	IPFS, Blockchain
GUI	Not yet
Programming language	Go, C++
Development Environment	Visual Studio Code
INSTALLATION AND DEPLOYMENT	
Software Requirements	Docker
Hardware Requirements	-
Containerization	Docker
Communications	-

5.6. VALIDATION

Several functional tests have been performed for this first implementation phase functionalities. Figure 112 resumes the main functionalities implemented in this first phase.

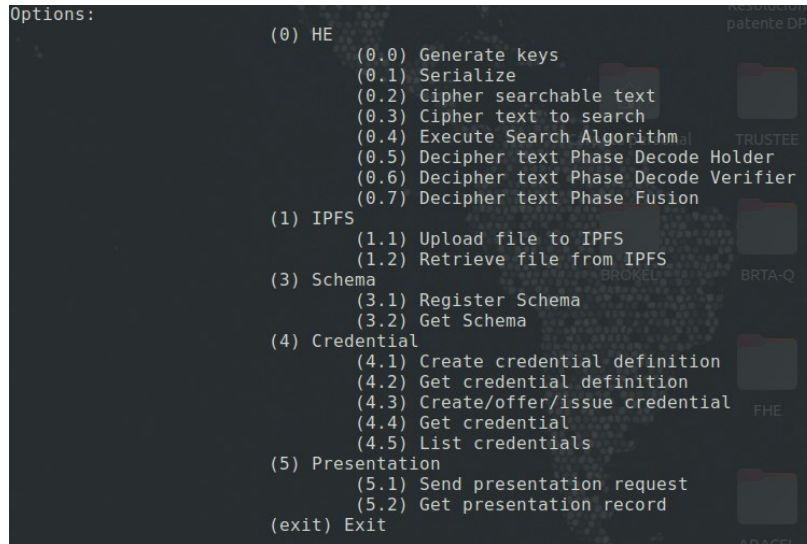


Figure 112: SSI-HE first implementation available functionalities

From this moment, the main functionalities for the three involved actors in the SSI-HE operation has been executed as a first functional validation.

5.6.1. MAIN FUNCTIONALITIES FOR THE ISSUER

- Functional Issuer_Test 1: Encrypt attributes with an HE public key from the data provider (see Figure 113).

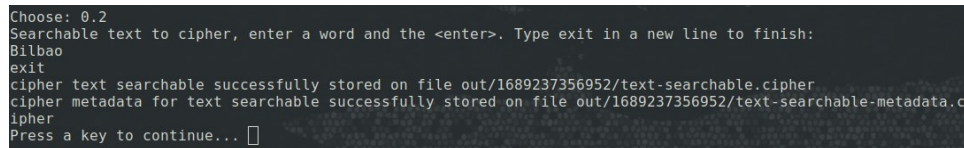


Figure 113: SSI-HE first implementation functional Issuer_Test 1

- Functional Issuer_Test 2: Upload HE encrypted attributes to IPFS and obtain the IPFS endpoint (see Figure 114).

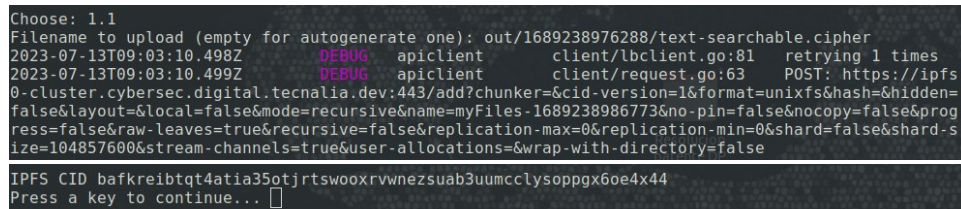


Figure 114: SSI-HE first implementation functional Issuer_Test 2

- Functional Issuer_Test 3: Create and issue SSI credentials with IPFS endpoints as attributes (see Figure 115).


```
Choose: 4.3
Schema ID: C9zuKQERhMPwAh3wrkSaJy:2:identity-schema:1.1
Schema Issuer DID: C9zuKQERhMPwAh3wrkSaJy
Credential Definition ID: C9zuKQERhMPwAh3wrkSaJy:3:CL:2519:tag
Issuer public DID: C9zuKQERhMPwAh3wrkSaJy
Comment: identity
Attribute "age" value: 30
Attribute "location" value: bafkreibtqt4atia35otjrtswooxrvwnezsuaab3uumcclysoyppgx6oe4x44
Credential Exchange ID: f2d6a243-249d-4123-a769-898651386737
Press a key to continue... [ ]
```

Figure 115: SSI-HE first implementation functional Issuer_Test 3

5.6.2. MAIN FUNCTIONALITIES FOR THE HOLDER

- Functional Holder_Test 1: Create HE public and secret (threshold) keys (see Figure 116).

```
Choose: 0.0
Keys generated:
  shared public key written to out/1689237356952/shared-key-public.json
  issuer private key written to out/1689237356952/issuer-key-private.json
  verifier private key written to out/1689237356952/verifier-key-private.json
Press a key to continue... [ ]
```

Figure 116: SSI-HE first implementation functional Holder_Test 1

- Functional Holder_Test 2: Store SSI credentials (see Figure 117).

```
Choose: 4.5
Schema ID of the credentials to filter: C9zuKQERhMPwAh3wrkSaJy:2:identity-schema:1.1
WARNING: Only the first 10 credentials will be displayed
Number of credentials - 1s
  Credential 1 of 1
  - credentialDefinitionID: C9zuKQERhMPwAh3wrkSaJy:3:CL:2519:tag
  - attribute: age value: 30
  - attribute: location value: bafkreibtqt4atia35otjrtswooxrvwnezsuaab3uumcclysoyppgx6oe4x44
Press a key to continue... [ ]
```

Figure 117: SSI-HE first implementation functional Holder_Test 2

- Functional Holder_Test 3: Pull encrypted attributes from IPFS from its endpoint (see Figure 118).

```
Choose: 1.2
CID to retrieve: bafkreibtqt4atia35otjrtswooxrvwnezsuaab3uumcclysoyppgx6oe4x44
DEBUG [2023-07-13T09:04:49.561]ipfs.go RetrieveFile:494 Downloaded a file 1689239089136.txt
module=ipfs
IPFS CID downloaded to 1689239089136.txt
Press a key to continue... [ ]
```

Figure 118: SSI-HE first implementation functional Holder_Test 3

- Functional Holder_Test 4: Decrypt (with two HE secret keys) the encrypted attributes (see Figure 119).

```
Choose: 0.6
Cipher metadata file to decipher: out/1689237356952/result-metadata-search-tfhe.cipher
Cipher data file to decipher: out/1689237356952/result-search-tfhe.cipher
threshold phase1 successfully done
data deciphered phase1 completed. Result stored out/1689237356952/resultB.cipher
Press a key to continue... [ ]
```

Figure 119: SSI-HE first implementation functional Holder_Test 4

- Functional Holder_Test 5: Verify what the issuer has certified (see Figure 120).


```
Choose: 0.7
Cipher metadata file to decipher: out/1689237356952/result-metadata-search-tfhe.cipher
ResultA data file: out/1689237356952/resultA.cipher
ResultB data file: out/1689237356952/resultB.cipher
threshold phase Fusion successfully done
data deciphered [[1]]
Press a key to continue... █
```

Figure 120: SSI-HE first implementation functional Holder_Test 5

- Functional Holder_Test 6: Create verifiable proofs with the IPFS endpoints of the encrypted attributes (see Figure 121).

```
Choose: 5.2
Presentation exchange ID: 03df46d4-5d50-42cf-80c5-d1fbf761e5fd
Presentation exchange ID status : true
attribute: age value: 30
attribute: location value: bafkreibtqt4atia35otjrtswooxrvwnezsuab3uumcclysoppgx6oe
4x44
Press a key to continue... █
```

Figure 121: SSI-HE first implementation functional Holder_Test 6

- Functional Holder_Test 7: Decryption of the result of the searching operation (partial and combined) (see Figure 122).

```
Choose: 0.7
Cipher metadata file to decipher: out/1689237356952/result-metadata-search-tfhe.cipher
ResultA data file: out/1689237356952/resultA.cipher
ResultB data file: out/1689237356952/resultB.cipher
threshold phase Fusion successfully done
data deciphered [[1]]
Press a key to continue... █
```

Figure 122: SSI-HE first implementation functional Holder_Test 7

5.6.3. MAIN FUNCTIONALITIES FOR THE VERIFIER

- Functional Verifier_Test 1: Encrypt the value for the attribute with the HE public key (see Figure 123).

```
Choose: 0.3
Text to search to cipher, enter a word and the <enter>. Type exit in a new line to finish:
Bilba
exit
search text cipher successfully stored on file out/1689238976288/text-search.cipher
search text cipher metadata successfully stored on file out/1689238976288/text-search-metadata.cipher
Press a key to continue... █
```

Figure 123: SSI-HE first implementation functional Verifier_Test 1

- Functional Verifier_Test 2: Make an SSI proof request to the data provider for a specific attribute (see Figure 124).

```
Choose: 5.1
SchemaID: C9zuKQERhMPwAh3wrkSaJy:2:identity-schema:1.1
CredentialDefinitionID: C9zuKQERhMPwAh3wrkSaJy:3:CL:2519:tag
Comment: identity proof request
Presentation exchange ID: 03df46d4-5d50-42cf-80c5-d1fbf761e5fd
Press a key to continue... █
```

Figure 124: SSI-HE first implementation functional Verifier_Test 2

- Functional Verifier_Test 3: Receive the SSI proof response from the data provider and obtain the IPFS endpoints from the SSI proof (see Figure 125).

```
Choose: 5.2
Presentation exchange ID: 03df46d4-5d50-42cf-80c5-d1fbf761e5fd
Presentation exchange ID status : true
      attribute: age value: 30
      attribute: location value: bafkreibtqt4atia35otjrtswooxrvwnezsuaab3uumcclysoppgx6oe4x44
4x44
Press a key to continue... [ ]
```

Figure 125: SSI-HE first implementation functional Verifier_Test 3

- Functional Verifier_Test 4: Obtain the encrypted attribute from IPFS (see Figure 126).

```
Choose: 1.2
CID to retrieve: bafkreibtqt4atia35otjrtswooxrvwnezsuaab3uumcclysoppgx6oe4x44
DEBUG [2023-07-13T09:04:49.561]ipfs.go RetrieveFile:494 Downloaded a file 1689239089136.txt
      module=ipfs

IPFS CID downloaded to 1689239089136.txt
Press a key to continue... [ ]
```

Figure 126: SSI-HE first implementation functional Verifier_Test 4

- Functional Verifier_Test 5: Execute the searching operation over the encrypted attributes. As a result, an encrypted result is obtained (see Figure 127).

```
Choose:0.4
searchable text cipher metadata file: out/1689237356952/text-searchable-metadata.cipher
searchable text cipher data file: out/1689237356952/text-searchable.cipher
text to search cipher metadata file: out/1689237356952/text-search-metadata.cipher
searchable text cipher data file: out/1689237356952/text-search.cipher
algorithm search-tfhe execution result stored at out/1689237356952/result-search-tfhe.cipher
algorithm search-tfhe execution result metadata stored at out/1689237356952/result-metadata-search-tfhe.cipher
her
Press a key to continue... [ ]
```

Figure 127: SSI-HE first implementation functional Verifier_Test 5

- Functional Verifier_Test 6: Partial decryption with one of the secret HE keys (see Figure 128).

```
Choose: 0.6
Cipher metadata file to decipher: out/1689237356952/result-metadata-search-tfhe.cipher
Cipher data file to decipher: out/1689237356952/result-search-tfhe.cipher
threshold phase1 successfully done
data deciphered phase1 completed. Result stored out/1689237356952/resultB.cipher
Press a key to continue... [ ]
```

Figure 128: SSI-HE first implementation functional Verifier_Test 6

- Functional Verifier_Test 7: Combination of the two partially decrypted results (see Figure 129).

```
Choose: 0.7
Cipher metadata file to decipher: out/1689237356952/result-metadata-search-tfhe.cipher
ResultA data file: out/1689237356952/resultA.cipher
ResultB data file: out/1689237356952/resultB.cipher
threshold phase Fusion successfully done
data deciphered [[1]]
Press a key to continue... [ ]
```

Figure 129: SSI-HE first implementation functional Verifier_Test 7

5.7. NEXT STEPS

As explained, the focus during this first implementation phase has been on the design of the SSI-HE component. Additionally, a first isolated and standalone prototype has been implemented. Interaction with other TRUSTEE components has been investigated; however, it is still a work in progress and further details will be provided in upcoming versions of this live document, namely D4.2 and D4.3.

The next versions of the SSI-HE component will include the following actions:

- TRUSTEE use cases particularization. Different SSI schemas and SSI credential definition will be created adapted to the different use cases particularities.
- Distribution of the three subcomponents (issuer, holder and verifier) in different instances to be deployed in different machines in order to provide more realism to the solution. The idea is to deploy the issuer instance in TECNALIA, in order to simulate a certification authority. The holder instance should be deployed at the different use cases premises. Finally, the verifier instance should be part of the TRUSTEE platform.
- Interactions with other TRUSTEE components. The required interactions with other TRUSTEE components will be implemented. More details about the input/output data models of the SSI-HE component will be defined for easier integration.
- GUI design and implementation. The issuer and holder instances will include their own GUI while the verifier will be integrated with the DA and AIMaaS ABBs. Recommendation guidelines from T4.2 will be followed.
- Extend validation tests once the SSI-HE component is integrated with the rest of the TRUSTEE platform.

5.8. CONCLUSIONS

This chapter has presented the documentation associated with the SSI-HE. The SSI-HE component allows TRUSTEE data consumers to identify suitable TRUSTEE data providers datasets for the specific requirements they have in terms of dataset attributes (features) while TRUSTEE data providers can keep these dataset attributes private or confidential.

This chapter has described in detail the associated requirements coverage, the component architecture, the main functionalities, and the operational flow inside the whole TRUSTEE platform. Furthermore, the first implementation has been described in terms of covered functionalities and technical details. Finally, some functional validation tests have been presented to prove the correct operation of the current implementation.

This component is currently deployed on TECNALIA for first validation purposes. In next implementation phases, the component will be distributed, as will be documented in following versions of this deliverable, namely D4.2 and D4.3.

6. PART IV: DATA LAKE

6.1. INTRODUCTION

This chapter provides an overview of the DL component within the TRUSTEE ecosystem. Primarily, the DL endeavours to establish a unified understanding of data and services across all TRUSTEE components. This strategic approach not only fosters semantic interoperability but also affords adaptable data management practices. This alignment is in line with prevalent standards such as the European Open Science Cloud (EOSC) Semantic Interoperability v1.0 framework and the findable, accessible, interoperable, and reusable (FAIR) principles, ensuring coherence in operations.

6.1.1. NEED AND OBJECTIVES

Semantic interoperability refers to the ability of different information systems to exchange and understand data in a meaningful way. In the context of the TRUSTEE project, achieving semantic interoperability is crucial for several reasons:

- **Machine Computable Logic:** To enable advanced data processing, reasoning, and decision-making, it is necessary to have a common understanding of the data and its semantics across the various components. By achieving semantic interoperability, we can unlock the potential for machine computable logic, allowing automated inferencing and knowledge discovery.
- **Data Federation:** The TRUSTEE project involves multiple technological components that need to collaborate and exchange data. Semantic interoperability ensures that data can be federated across these systems, enabling seamless integration and analysis of data from diverse sources.
- **Harmonization and Consistency:** The TRUSTEE reference architecture comprises different components, each with its own data models. To ensure harmonization and consistency across these components, it is essential to establish guidelines that promote a common understanding of the data and services offered. Semantic interoperability facilitates the alignment of data models, allowing for efficient data integration and minimizing inconsistencies.
- **Data lake development:** A data lake is necessary to address the following challenges and requirements:
 - **Data Integration:** The TRUSTEE project involves multiple components that generate and consume data. To enable seamless integration and analysis of this data, a central data lake is required. It provides a unified platform for aggregating and harmonizing data from various sources, promoting efficient data sharing and collaboration.
 - **Scalability and Flexibility:** As the project progresses, the volume and variety of data are expected to increase. A data lake offers the scalability and flexibility needed to accommodate growing data volumes and diverse data types. It allows for the ingestion of both structured and unstructured data, ensuring the project's readiness to handle future data expansion.
 - **Data Discovery and Exploration:** The data lake provides an environment for data discovery and exploration. Researchers and stakeholders can easily access and search for relevant data within the lake, facilitating knowledge discovery and supporting data-driven decision-making.

For the DL design and implementation, the following objectives have been defined.

- **Analyzing Data Models:** The task involves a comprehensive analysis of the data models supported by the various components within the TRUSTEE reference architecture. This analysis will identify the similarities, differences, and potential gaps in the data models.

- **Establishing Guidelines:** Based on the analysis, the task aims to develop guidelines that ensure a common understanding of the data and services offered by each component. These guidelines will serve as a foundation for achieving semantic interoperability and enabling machine computable logic, inferencing, knowledge discovery, and data federation.
- **Leveraging FAIR Principles:** The work will align with the principles of the European Research Data Commons, where data are FAIR. By adopting these principles, the task will ensure that the guidelines for semantic interoperability adhere to recognized standards and best practices.
- **Compliance with EOSC Semantic Interoperability v1.0:** The task will strive to achieve compliance with the EOSC Semantic interoperability v1.0. This version defines the specifications and requirements for semantic interoperability within the EOSC framework, and adherence to these guidelines will enhance the compatibility and interoperability of the TRUSTEE project with other EOSC initiatives.
- **Designing and Implementing a Robust Data Lake:** This objective entails designing and implementing a robust data lake architecture that aligns with the TRUSTEE reference architecture and supports the requirements of the project. The data lake should be capable of efficiently ingesting, storing, managing, and processing large volumes of structured and unstructured data.

6.1.2. INNOVATION

The task of analyzing data models and establishing guidelines for semantic interoperability within the TRUSTEE project represents a significant opportunity for innovation. By addressing the needs and objectives outlined in the previous sub-section, TRUSTEE aims to drive innovation in the following key areas:

Advancing Machine Computable Logic

Semantic interoperability enables a common understanding of data semantics, allowing for advanced data processing, reasoning, and decision-making. By achieving semantic interoperability, the TRUSTEE project opens up new possibilities for machine computable logic. Automated inferencing and knowledge discovery can be leveraged to gain deeper insights from the integrated data and enable intelligent automation in data analysis workflows. This innovation empowers researchers and stakeholders to make data-driven decisions more efficiently and effectively.

Enabling Data Federation

Semantic interoperability plays a crucial role in facilitating data federation across the various technological components involved in the TRUSTEE project. By establishing a common understanding of data and services, the task ensures that data from diverse sources can be seamlessly integrated and analyzed. This innovation enables collaboration and data sharing among different systems, enhancing the ability to perform comprehensive analyses and derive insights from integrated data sources.

Promoting Harmonization and Consistency:

The TRUSTEE reference architecture encompasses multiple components, each with its own data models. The establishment of guidelines for semantic interoperability ensures harmonization and consistency across these components. This innovation streamlines data integration efforts, minimizing inconsistencies and enhancing the overall compatibility of the project's information systems. Researchers and stakeholders benefit from a cohesive and unified understanding of data and services, enabling smoother collaboration and more accurate analysis outcomes.

Leveraging FAIR Principles:

By aligning with the principles of the European Research Data Commons, such as the FAIR principles, the task promotes innovation in data management and sharing. Ensuring that the guidelines for semantic interoperability adhere to recognized standards and best practices fosters an environment where data becomes more discoverable, accessible, and reusable. This innovation maximizes the value of the data generated within the TRUSTEE project, allowing for its broader utilization and potential for cross-domain collaborations.

Complying with EOSC Semantic Interoperability v1.0:

Striving to achieve compliance with the EOSC Semantic interoperability v1.0 represents a significant innovation in the context of the TRUSTEE project. Adhering to these specifications and requirements enhances compatibility and interoperability with other initiatives within the European Open Science Cloud framework. This innovation not only ensures seamless integration with existing EOSC infrastructure but also contributes to the overall vision of building an interconnected and harmonized research environment at a European level.

Developing a Robust Data Lake:

The objective of designing and implementing a robust data lake aligns with the TRUSTEE reference architecture and addresses the challenges associated with data integration, scalability, and flexibility. By creating a centralized repository capable of ingesting, storing, managing, and processing large volumes of structured and unstructured data, this innovation lays the foundation for efficient data sharing, collaboration, and data-driven decision-making. The data lake provides an environment for data discovery and exploration, fostering innovation in knowledge discovery and supporting interdisciplinary research efforts.

In this sense, TRUSTEE drives innovation in multiple areas, including machine computable logic, data federation, harmonization and consistency, adherence to FAIR principles, compliance with EOSC standards, and the development of a robust data lake. By pushing the boundaries of semantic interoperability and data management, this task enables novel approaches to data analysis, collaboration, and knowledge discovery, contributing to the advancement of research and innovation within the TRUSTEE project and beyond.

6.2. BACKGROUND

This sub-section provides background information on the main concepts relevant to the DL ABB.

6.2.1. SEMANTIC INTEROPERABILITY

Data semantic interoperability refers to the ability of different systems and applications to exchange and interpret data with shared meaning and understanding. It involves establishing common vocabularies, ontologies, and data models to enable effective communication and integration of heterogeneous data sources. Research studies have highlighted the importance of data semantic interoperability in various domains.

Study [56] emphasized the role of semantic technologies in achieving data semantic interoperability. The researchers proposed a semantic interoperability framework that utilized ontologies to represent and align domain knowledge across different systems. The framework facilitated the integration and exchange of data by ensuring that the meaning of the data was preserved during the communication process. The authors demonstrated the effectiveness of their approach in the healthcare domain, where interoperability is critical for seamless data exchange between different healthcare systems.

Study [57] explored the challenges and opportunities of data semantic interoperability in the context of the IoT. The authors discussed the heterogeneity of IoT devices and the need for a common understanding of data to enable efficient collaboration and integration. They proposed a semantic interoperability architecture that utilized semantic models and ontologies to bridge the gap between different IoT systems and facilitate meaningful data exchange. The study highlighted the potential of data semantic interoperability to unlock the full potential of IoT applications in various domains.

Furthermore, [58] focused on the significance of data semantics in achieving interoperability in distributed data systems. The authors discussed the challenges posed by data heterogeneity and the lack of shared understanding across different data sources. They proposed the use of semantic mapping techniques to establish a common semantic layer that enables data integration and exchange. The study emphasized the role of metadata and semantic annotations in facilitating data semantic interoperability and enabling seamless data integration across distributed systems.

Ongoing research emphasises that by establishing common vocabularies, ontologies, and semantic models, data semantic interoperability enables effective communication, integration, and interpretation of data, ultimately enhancing the value and usability of heterogeneous data sources.

6.2.1.1. EOSC SEMANTIC INTEROPERABILITY FRAMEWORK V1.0

The EOSC [59] is a pioneering initiative aiming to provide seamless access to scientific data and services across Europe. At the core of this ambitious project lies the concept of semantic interoperability, which ensures that data from diverse sources can be shared, integrated, and understood coherently. EOSC's semantic interoperability framework v1.0 is built on several key principles that promote data accessibility, collaboration, and reuse:

FAIR Principles: The FAIR principles form the foundation of EOSC's semantic interoperability. By adhering to these principles, EOSC ensures that data resources are easily discoverable, accessible, and reusable by both humans and machines. Making data FAIR facilitates seamless integration and meaningful exchange of information among researchers and disciplines.

Semantic Web Technologies: EOSC leverages Semantic Web technologies like Resource Description Framework (RDF), Ontology Web Language (OWL), and Protocol and RDF Query Language (SPARQL) to represent and link data in a machine-readable format. RDF enables the expression of data and its relationships in a structured manner, enhancing data understanding and interpretation by computers. OWL allows the creation of ontologies that define domain-specific knowledge, enabling standardized data representation and facilitating data integration.

Ontologies and Vocabularies: EOSC promotes the use of ontologies, which provide formal representations of knowledge in specific domains. Ontologies define entities, attributes, and relationships, establishing a shared vocabulary for data interpretation and exchange. The adoption of standardized vocabularies enhances data consistency and supports seamless integration of data from diverse sources, overcoming issues related to data heterogeneity.

Metadata Standards: Metadata standards play a pivotal role in describing and annotating datasets and services in EOSC. By adhering to common metadata standards compliant with the FAIR principles, data providers ensure that their resources are properly documented and enriched with contextual information. This metadata-centric approach promotes data discoverability and aids in the integration of data from different sources.

EOSC's semantic interoperability framework v1.0 offers various benefits and potential impacts [60]:

Enhanced Data Integration and Reuse: Semantic interoperability enables researchers to integrate data from different disciplines, promoting interdisciplinary collaboration and unlocking new research opportunities. By providing a

common language for data representation, EOSC fosters data reuse and facilitates the generation of new insights from existing data.

Accelerated Scientific Discoveries: The seamless integration and access to diverse datasets through EOSC's semantic interoperability framework can accelerate scientific discoveries. Researchers can efficiently combine and analyze data from various sources, promoting data-driven research and enabling exploration of complex scientific questions.

Improved Data Discoverability: The adoption of semantic web technologies and FAIR principles enhances the discoverability of data in EOSC. Researchers can efficiently search and access relevant datasets, ensuring that valuable data resources are not underutilized. This increased discoverability encourages data sharing and collaboration, ultimately advancing open science objectives.

Despite its potential, EOSC's semantic interoperability framework v1.0 faces challenges and limitations:

Heterogeneity of Data and Ontologies: Integrating data from diverse sources with varying structures and vocabularies remains a complex challenge. Ontology mapping and alignment techniques are necessary to bridge these differences, but they require considerable effort and expertise.

Data Quality and Provenance: Semantic interoperability relies on accurate and reliable data, including high quality ontologies and metadata. Ensuring data quality and provenance is essential to avoid misleading interpretations and foster trust in the shared data resources.

Community Adoption and Engagement: The success of the EOSC semantic interoperability framework depends on widespread adoption and active participation from the research community. Encouraging researchers to embrace semantic web technologies, FAIR principles, and standardized vocabularies requires awareness-raising, training, and incentives.

The EOSC semantic interoperability framework v1.0 plays a pivotal role in enabling open and seamless access to scientific data and services across Europe. By embracing the FAIR principles and leveraging Semantic Web technologies, EOSC promotes data integration, reuse, and discoverability, accelerating scientific discoveries and advancing open science objectives. While challenges persist, ongoing initiatives and collaborations hold the promise of enhancing semantic interoperability, fostering a vibrant research ecosystem that benefits the scientific community and society at large.

6.2.1.2. FAIR PRINCIPLES: ADVANCING OPEN SCIENCE THROUGH DATA ACCESSIBILITY AND INTEROPERABILITY

In the era of big data and advanced technology, ensuring open access to scientific data is crucial for accelerating research and innovation. The FAIR principles (Figure 130) offer a groundbreaking framework that promotes the accessibility and interoperability of data across diverse scientific domains [61].

- i. Findable. The first principle of FAIR emphasizes the importance of making data findable by both humans and machines. To achieve this, data should be assigned a unique and persistent identifier, commonly in the form of a Digital Object Identifier (DOI) or a URL. Additionally, rich metadata describing the data's content, context, and provenance must accompany the dataset, facilitating its discovery through online repositories, search engines, or data catalogues. By enhancing data discoverability, researchers can efficiently locate relevant data resources, fostering collaboration and knowledge exchange.

- ii. Accessible. The accessibility principle centers on ensuring that data is readily available to all potential users. This principle advocates for removing technical, legal, and financial barriers that hinder data access. Data should be openly accessible under well-defined conditions, allowing researchers to access and download datasets without facing unnecessary restrictions. Data providers must consider privacy and ethical concerns, offering different levels of access control while ensuring data privacy and security. Through improved accessibility, the scientific community can leverage existing data for new insights and validate research findings more effectively.
- iii. Interoperable. The third principle, interoperability, addresses the challenge of integrating and combining data from diverse sources. To achieve interoperability, data should be represented using standardized and widely accepted data models and formats. The adoption of common vocabularies and ontologies enables a shared understanding of data meaning across disciplines, facilitating seamless data integration and exchange. Moreover, the use of machine-readable formats, such as the RDF and structured data files, enhances the capability of automated data processing and analysis. By embracing interoperability, researchers can harness the full potential of integrated data, unlocking novel research opportunities.
- iv. Reusable. The final FAIR principle focuses on making data reusable for different purposes and by diverse user communities. Reusability is achieved through the provision of clear and comprehensive licenses that specify the terms and conditions for data reuse. Data should be accompanied by documentation that outlines the methods used to generate and process the data, enabling others to replicate or build upon the research. By encouraging data reuse, researchers can validate findings, support data-driven decision-making, and maximize the societal impact of research.

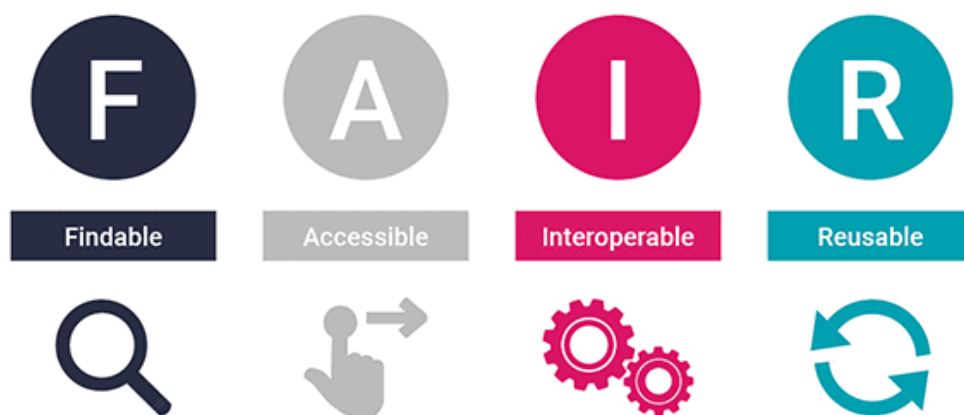


Figure 130: FAIR Principles³²

The FAIR principles have transformative implications for open science and research practice [62]:

Accelerating Scientific Discovery. By adhering to the FAIR principles, researchers can efficiently find and access a wealth of data resources across diverse domains. This accelerated access to data fosters collaboration, enabling researchers to explore complex research questions and discover new insights. It streamlines the research process, allowing researchers to focus on analysis and interpretation rather than data collection.

³² <https://biosistemika.com/blog/data-integrity-and-fair-principles-faq/>

Promoting Data Sharing and Collaboration. The FAIR principles promote a culture of data sharing and collaboration in the scientific community. By making data findable, accessible, and interoperable, researchers are encouraged to share their data openly. This culture of openness and collaboration advances scientific knowledge and accelerates breakthroughs, fostering a cooperative research environment.

Enhancing Reproducibility and Transparency. Reproducibility is a cornerstone of scientific research. The FAIR principles support reproducibility by ensuring that data and research outputs are well-documented and reusable. By providing comprehensive metadata and data provenance, researchers can validate and verify research findings, enhancing the credibility and trustworthiness of scientific research.

Facilitating Data-Driven Decision Making. The FAIR principles enable policymakers, organizations, and businesses to leverage open data for data-driven decision-making processes. Access to high quality and interoperable data empowers stakeholders to make informed choices, implement evidence-based policies, and address societal challenges more effectively.

The FAIR principles represent a paradigm shift in the way scientific data is managed, shared, and utilized. By promoting data findability, accessibility, interoperability, and reusability, these principles facilitate collaboration, enhance scientific discovery, and accelerate progress in research and innovation. Embracing the FAIR principles will continue to play a pivotal role in fostering an open, transparent, and collaborative research ecosystem, advancing open science and contributing to the betterment of society as a whole.

6.2.2. DATA LAKES

Data lakes are designed to accommodate large volumes of raw, unprocessed data in their native format. Unlike traditional data warehouses that require data to be pre-modeled and structured before ingestion, data lakes adopt a "schema-on-read" approach. This architecture allows organizations to ingest data from various sources, such as social media, IoT devices, logs, and databases, without the need for immediate data transformation [63].

Data lakes have revolutionized the way organizations store, manage, and analyze vast amounts of data from various sources. Unlike traditional data warehouses, data lakes accommodate raw, unstructured, and structured data in its native format. This article delves into the technical aspects of data lakes, exploring how they work, their key features, and the mechanisms they employ to store data efficiently [64].

Data lakes are designed to be highly scalable and flexible, allowing organizations to ingest data from diverse sources without the need for immediate data transformation. The architecture of a data lake revolves around three primary components [65]:

- i. **Data Ingestion Layer:** This layer is responsible for capturing and ingesting data from various sources, such as databases, log files, social media, IoT devices, and more. Data is stored in its raw form, preserving its original structure.
- ii. **Storage Layer:** The storage layer serves as the central repository of data in the data lake. It utilizes scalable distributed file systems like Hadoop Distributed File System (HDFS) or cloud-based object storage systems like Amazon S3 or Azure Blob Storage. Data is stored in its native format, eliminating the need for schema modifications during ingestion.
- iii. **Processing Layer:** The processing layer enables data transformation, analytics, and data preparation. Various processing frameworks, such as Apache Spark or Apache Hadoop MapReduce, are used to process and analyze the data residing in the storage layer.

Some of the most important technical Features of Data Lakes are included below, which provide essential information on how Data Lakes work and the reasons they are selected to store large amounts of data [66].

- i. **Schema-on-Read:** One of the key features of data lakes is the "schema-on-read" approach. Unlike traditional databases or data warehouses that enforce a schema during data ingestion, data lakes allow users to apply a schema when the data is read or queried. This approach provides greater flexibility, as it enables users to interpret and analyze the data in different ways without modifying the original data.
- ii. **Near-Real-Time Data Ingestion:** Data lakes can handle near-real-time data ingestion, making them suitable for streaming data sources. With the integration of streaming frameworks like Apache Kafka or Amazon Kinesis, data lakes can capture and process data as it arrives, enabling organizations to gain real-time insights and respond to events promptly.
- iii. **Horizontal Scalability:** Data lakes are built to scale horizontally, meaning they can handle increasing amounts of data by adding more nodes or storage resources to the system. This scalability ensures that data lakes can accommodate petabytes or even exabytes of data, making them suitable for enterprises dealing with massive data volumes.
- iv. **Cost-Effectiveness:** Data lakes leverage cost-effective storage solutions, especially when deployed in the cloud. By utilizing commodity hardware or cloud-based object storage, data lakes reduce infrastructure costs, making them an attractive option for businesses of all sizes.
- v. **Advanced Data Analytics:** Data lakes provide a unified and comprehensive view of all data types, enabling advanced data analytics, including machine learning, artificial intelligence, and predictive modelling. This feature empowers organizations to extract valuable insights from their data, leading to data-driven decision-making.

Data lakes employ various storage mechanisms to efficiently store data and ensure high performance. Some common data storage mechanisms include [67]:

- i. **Object Storage:** Cloud-based data lakes often utilize object storage systems, such as Amazon S3, Google Cloud Storage, or Azure Blob Storage. Object storage stores data in discrete units called objects, each with its own unique identifier. This mechanism allows for easy retrieval, scalability, and cost-effectiveness, as organizations only pay for the storage capacity they use.
- ii. **Distributed File Systems:** On-premises data lakes often use distributed file systems like HDFS. These file systems divide large datasets into smaller blocks and distribute them across multiple nodes in a cluster. This approach provides fault tolerance, data redundancy, and high availability.
- iii. **Columnar Storage:** Columnar storage formats, such as Apache Parquet and Apache Optimized Row Columnar (ORC), are commonly used in data lakes. These formats organize data by columns instead of rows, enabling faster data retrieval and compression. This feature significantly improves query performance and reduces storage space requirements.
- iv. **Data Partitioning:** Data lakes often employ data partitioning techniques to organize data based on certain attributes, such as date, region, or customer ID. Partitioning enhances query performance by allowing the system to read only the relevant data needed for a specific analysis, reducing the data scanning overhead.
- v. **Metadata Management:** Data lakes rely heavily on metadata management to keep track of the data stored within the lake. Metadata includes information about data types, data sources, data owners, data lineage, and more. Efficient metadata management facilitates data discovery, data governance, and data lineage tracking.

6.3. REQUIREMENTS

In this chapter, the compliance of the DL ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding with the aforementioned requirements addressed during the development and by the

implementation of the DL and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements have been considered. As the implementation of the DL progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

User-centred requirements were identified through the UX Research that was conducted under T2.3. These requirements were reported in D2.1 [1], while they were also utilized in order to extract technical, functional and non-functional requirements, that were also outlined in D2.1. Table 42 below presents the interplay between user-centred requirements and technical requirements, that is the functional and non-functional requirements that address each user-centred requirement. The user-centred requirement ID is used to reference the respective requirement reported in D2.1 [1] while the requirement's description is also provided alongside whether the DL is currently fully or partially compliant with the user-centred requirement, i.e., indicating whether technical requirements have been identified in order to address the respective user-centred requirement. The ID of the technical requirement, as reported in D2.1 [1], that is addressing each user-centred requirement is also provided in the last column of the table. Additionally, it is possible than more than one technical requirement addresses a single user-centred requirement.

Table 42: DL - User-centred requirements

Req id (WP2)	Req. Description	Compliance	How it is addressed
USR-001	TRUSTEE must enable collaboration among domains and enable access and use of data for secondary use from multiple domains	Partially	Req-DL-FUNC-1 Req-DL-FUNC-2 Req-DL-FUNC-3
USR-002	TRUSTEE must use standardized methods and clear instructions for correct data entry and formatting	Yes	Req-DL-FUNC-1
USR-016	TRUSTEE should enable interoperability and integration of data from different data sources	Yes	Req-DL-FUNC-1
USR-018	TRUSTEE should be compatible with existing data exchange technology infrastructures, such as IDS and Gaia-X	Partially	Req-DL-FUNC-1 DL will communicate with GAIA-X and IDS through their connectors
USR-022	TRUSTEE should enable access to fully-formed, timestamped, and synchronized datasets	Yes	Req-DL-FUNC-1 Metadata will exist for each dataset with time-related information
USR-023	TRUSTEE should enable access to numerical and descriptive data	Yes	Req-DL-FUNC-1
USR-026	TRUSTEE should support data that contain meta-data, text based included.	Yes	Req-DL-FUNC-1
USR-029	TRUSTEE should offer collaboration across multi-disciplinary data domains	Partially	Req-DL-FUNC-1 Req-DL-FUNC-2 Req-DL-FUNC-3

USR-032	TRUSTEE should enable query and align datasets with other data from different domains	Yes	Req-DL-FUNC-1 Querying the RDF-zed datasets' concepts allows their alignment across domains
----------------	---	-----	--

Table 43 presents the preliminary results of the investigation regarding the compliance of the DL with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether DL complies with the requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 43: DL – Functional and non-functional technical requirements

Req id (New)	Req. Description	Compliance	How is addressed
Req-DL-FUNC-1	The TRUSTEE platform should be able to quickly and efficiently search across multiple data sources for the user selected criteria.	Yes	The usage of standards prototypes such as existing RDF semantic ontologies for the annotation of datasets allows the seamless and quick retrieval of the appropriate data concepts.
Req-DL-FUNC-2	All data processing within TRUSTEE should be done according to relevant privacy regulations. No personally identifiable data should ever be revealed.	Yes	No raw data will be stored within DL
Req-DL-FUNC-3	TRUSTEE should comply with relevant IPR and data security regulations. Data providers should be able to control how their data is processed and for which users	Partially	No raw data will be stored within DL

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

6.4. ARCHITECTURE

6.4.1. HIGH LEVEL

Taking into consideration the needs of data consumers using the TRUSTEE solution, the architecture of the ecosystem includes a data lake that stores diverse types and formats of data from various data providers based on the consumers' requirements. Additionally, each data provider has its own decentralized data lake located on their premises to store their specific data.

Figure 131 shows the architecture of the DL, including the ABB's sub-components as well as the communications required with other ABBs of the TRUSTEE ecosystem.

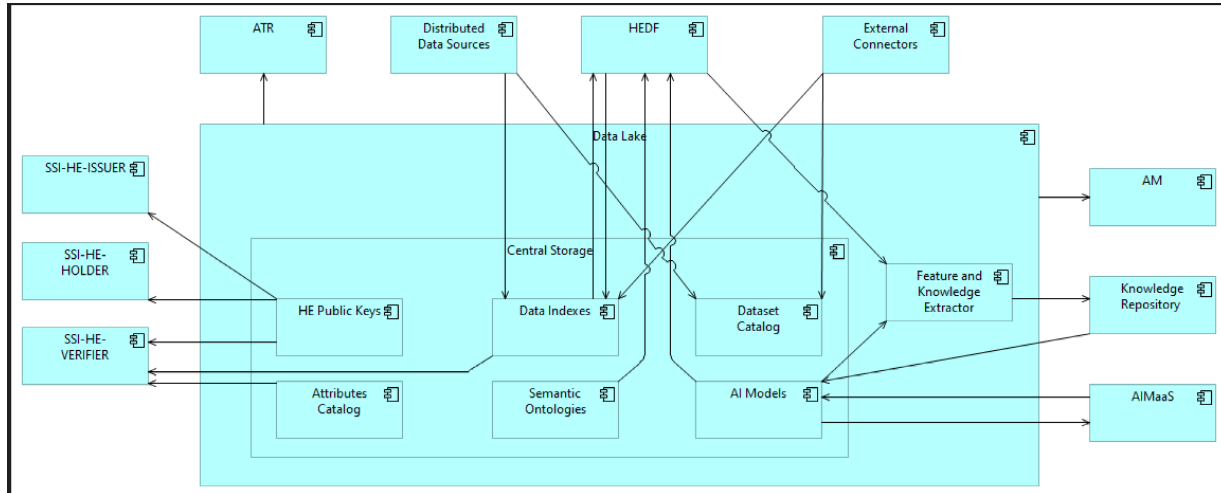


Figure 131: DL ABB breakdown diagram

As shown in the architecture diagram, DL interacts with multiple components and external interfaces within the TRUSTEE system. Firstly, the HEDF component is responsible for creating indexes of available datasets from each data provider. The DL retrieves these indexes for storage, to allow efficient access to the relevant data.

Another important communication includes the SSI-HE components, which interact with the DL to:

- store and provide data providers' public keys for secure metadata exchange and management.
- store and provide homomorphic public keys for metadata exchange, ensuring secure communication.
- provide data provider endpoints and indexes.

Furthermore, the DL collaborates with the ATR by sending transaction records for recording purposes. This ensures proper tracking and accountability of data-related activities. External connectors also enable the DL to index and incorporate data information from external sources. This enhances the scope and diversity of data available for consumers.

To facilitate authentication processes, the DL communicates with the AM to exchange necessary user authentication information. Additionally, the DL integrates with the AIMaaS component. It provides the AIMaaS ABB with the KR index for existing model runs and receives new AI models and logs for storage.

Finally, the KR plays a significant role in the architecture. The DL searches the KR for logs from previous AI model runs based on AI model and dataset indexes. If a previous run exists, the DL retrieves the KR index and shares it with AIMaaS. Conversely, the KR sends the DL the index of the last entry performed by AIMaaS, along with the model and dataset indexes. This information allows the DL to record a new entry in the log, capturing the details of the current run.

6.4.1. LOW LEVEL

The following sub-section provides an overview of the lower-level architecture for DL. This includes representation and description of each sub-component and their inter-connections.

Semantic repository - GraphDB

The Semantic repository sub-component consists of a GraphDB RDF store, which is responsible for storing all semantic annotations for the TRUSTEE domains. An instance of the GraphDB store has been installed and includes all semantic annotations available within TRUSTEE.

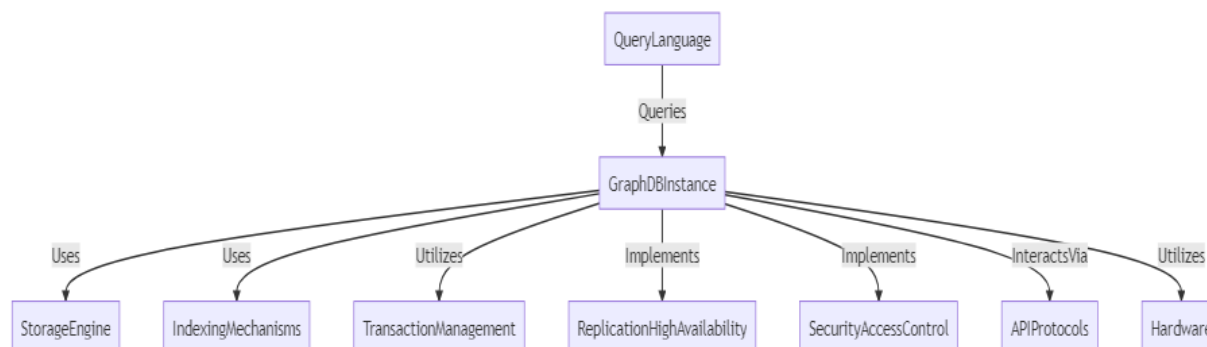


Figure 132: GraphDB architecture

Figure 132 shows the different mechanisms utilized within the GraphDB engine for the proper storage, access, processing and retrieval of the semantic concepts. Each dataset consists of RDF triplets, which are stored within the Storage Engine and the indexing mechanisms in place within GraphDB allow the easy retrieval of each concept. With the SPARQL querying language, we can access all available data and use API protocols to a) get dynamic fields as query parameters from external interfaces and b) post query results to external interfaces.

Rest of the DL

The rest of the DL architecture (Figure 133) includes sub-components that will store the HE public keys, data indexes and data catalogues as well as AI models in order to ensure transparent and flexible retrieval of any elements required by the other TRUSTEE platform ABBs.

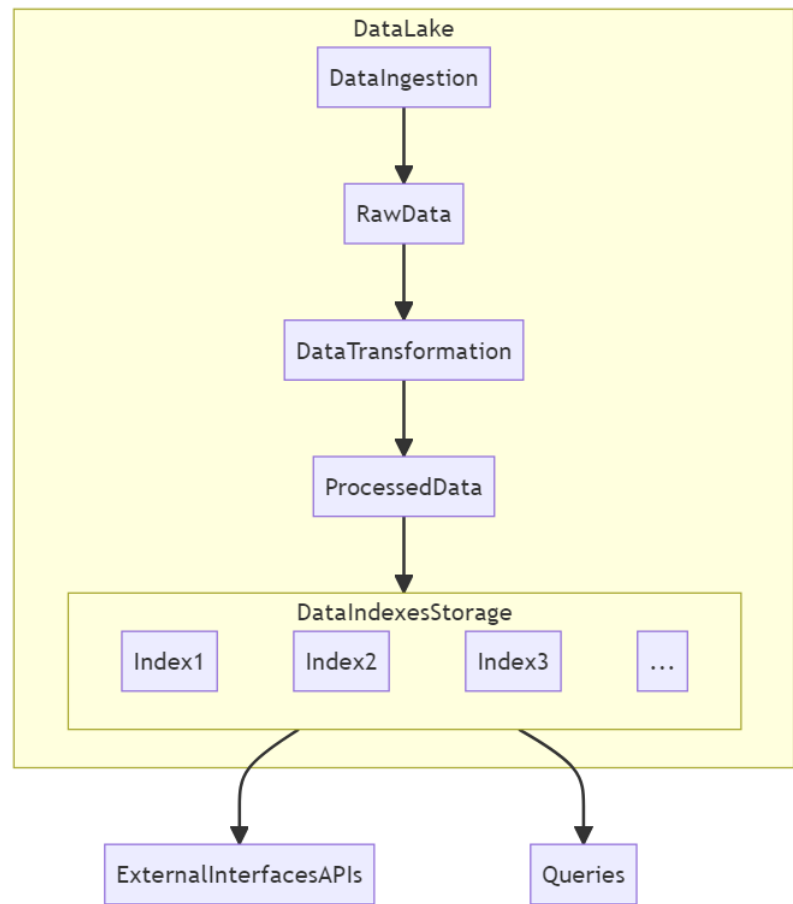


Figure 133: Data Indexes low level architecture

Another sub-component within the DL is the Data catalogs storage unit, which will be responsible for the communication with the connectors of the external data sources such as GAIA-x and IDS, and store the available catalogs (Figure 134).

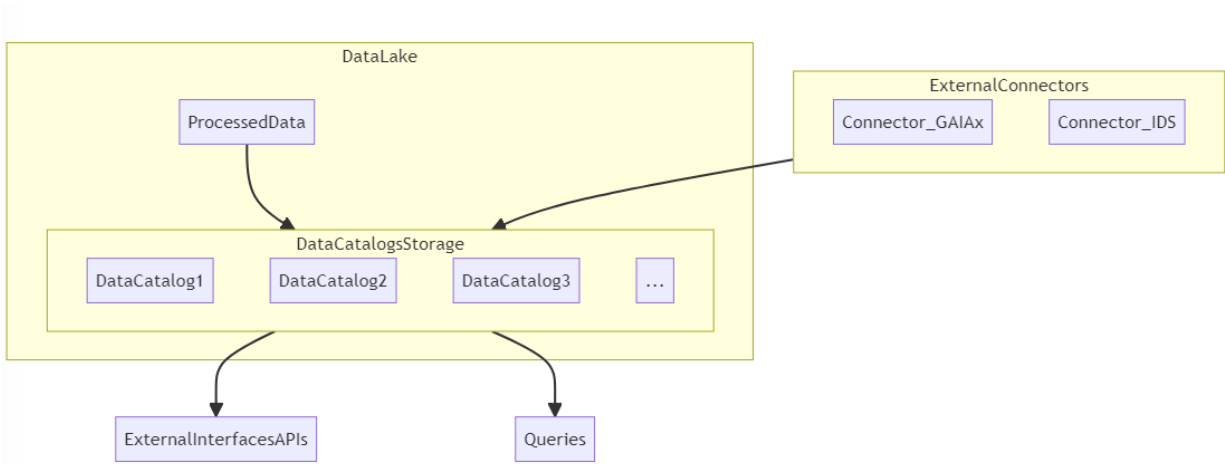


Figure 134: Data catalogues low level architecture

The DL will also store the attributes that will be available for retrieval of information about which data providers and their respective datasets are appropriate for operations based on the user's selections in the TRUSTEE platform. Each attribute can consist of different values e.g. date, location, domain and other characteristics that uniquely identify a dataset (Figure 135).

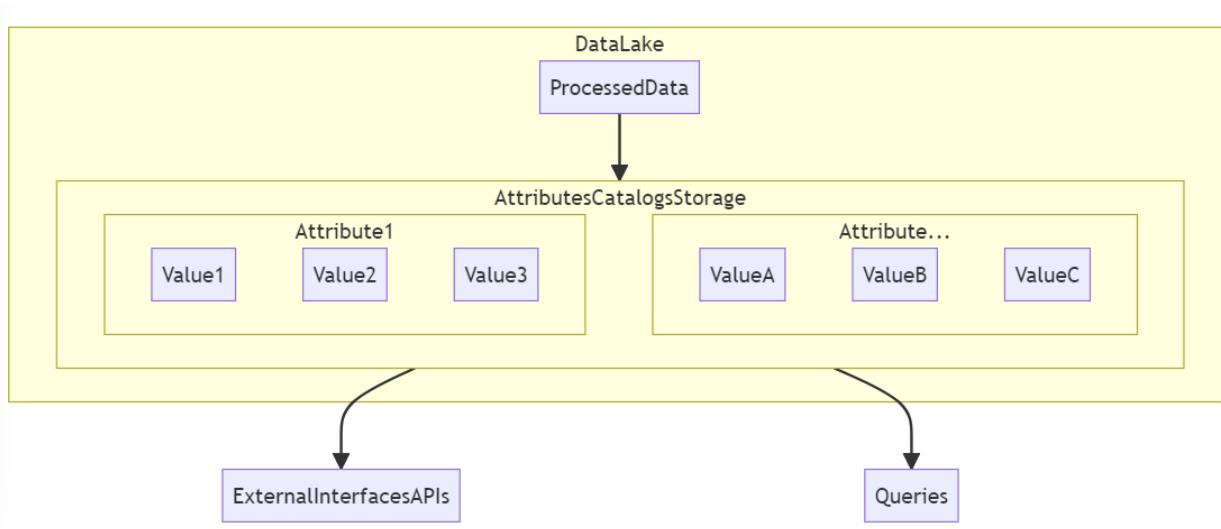


Figure 135: Attributes catalogue low level architecture

Another sub-component involves the storage of AI models that will be available for usage by the TRUSTEE users through the AIMaaS ABB. The DL will store the AI models, enabling data consumers to easily search and select a preferred model based on its description and the adopted HE schema that was used for training (for pre-trained models). Meanwhile, model providers can utilize this space to store all essential files needed for deploying and running their models as services (Figure 136).

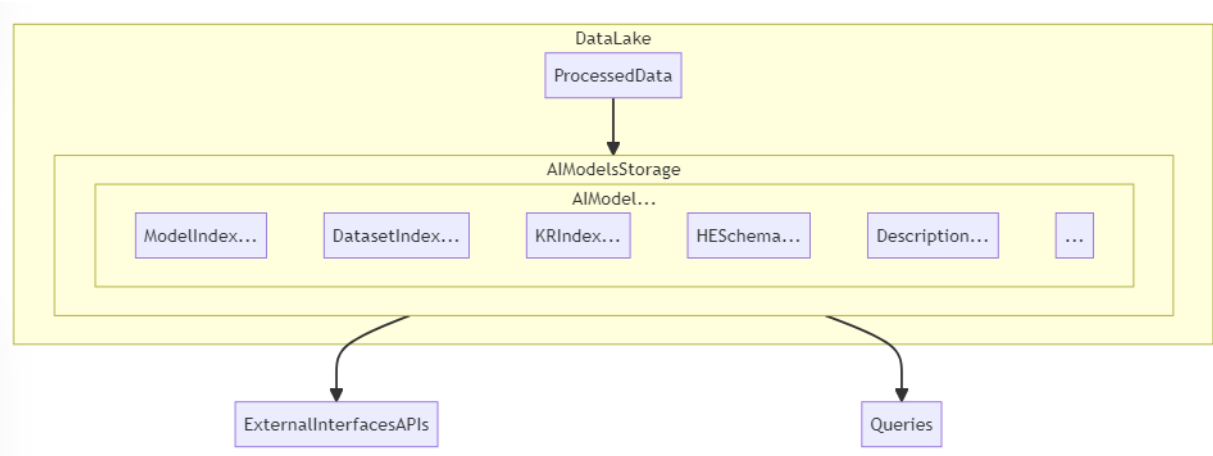


Figure 136: AI models storage low level architecture

The last sub-component within the DL includes the storage of the HE public keys that are generated for the realization of HE operations across the different datasets (Figure 137). The DL HE Public Keys storage component will store information about each public key so that it is easily retrievable by the ABBs that need this information (e.g. SSI-HE).

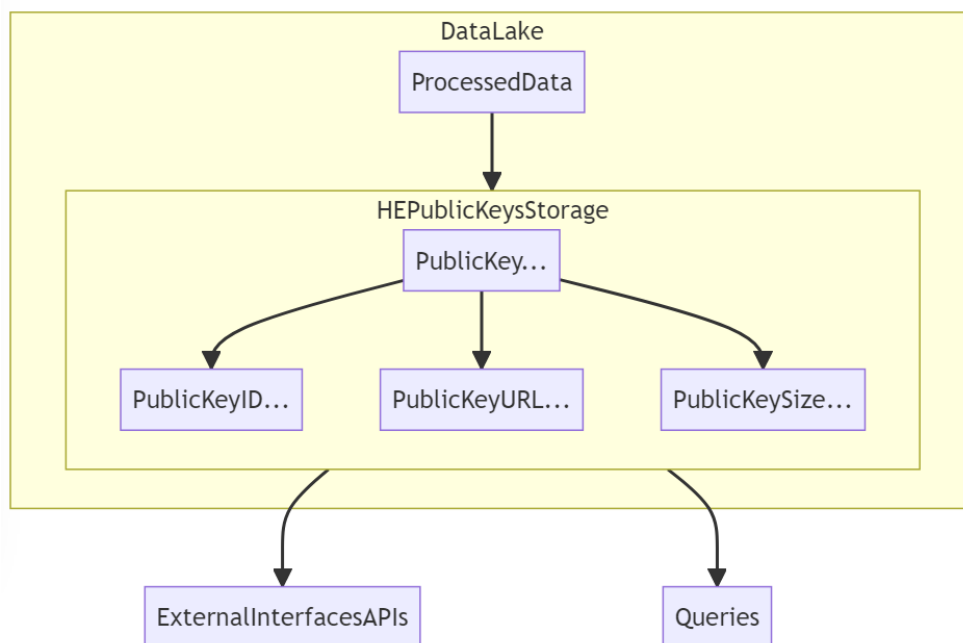


Figure 137: HE Public Keys storage low level architecture

6.5. IMPLEMENTATION

6.5.1. FUNCTIONAL DESCRIPTION

The DL will be responsible for the storage of different types of information that is crucial to the operations of TRUSTEE. Such information includes the:

- HE Public keys that will be generated and managed by the SSI-HE ABB.
- Data indexes and endpoints of all the data available from each data provider.
- Data catalogues with information available by the external connectors and data sources (GAIA-X, IDS, EGI etc.).
- Attributes catalogues with encrypted confidential attributes to categorize and perform search over the data providers' endpoints.
- AI/ML models & algorithms.
- Logs with information on models' executions.
- Semantic ontologies that will describe the datasets' schemas.

The DL will communicate with the different ABBs that require the above data for their own purposes within TRUSTEE, and through APIs communication protocols, will transfer information across the TRUSTEE platform.

6.5.2. OPERATIONAL DESCRIPTION

The DL will cover operations for all TRUSTEE users. Figure 138 shows the sequence diagram of the overall operations that will be carried out by the DL ABB.

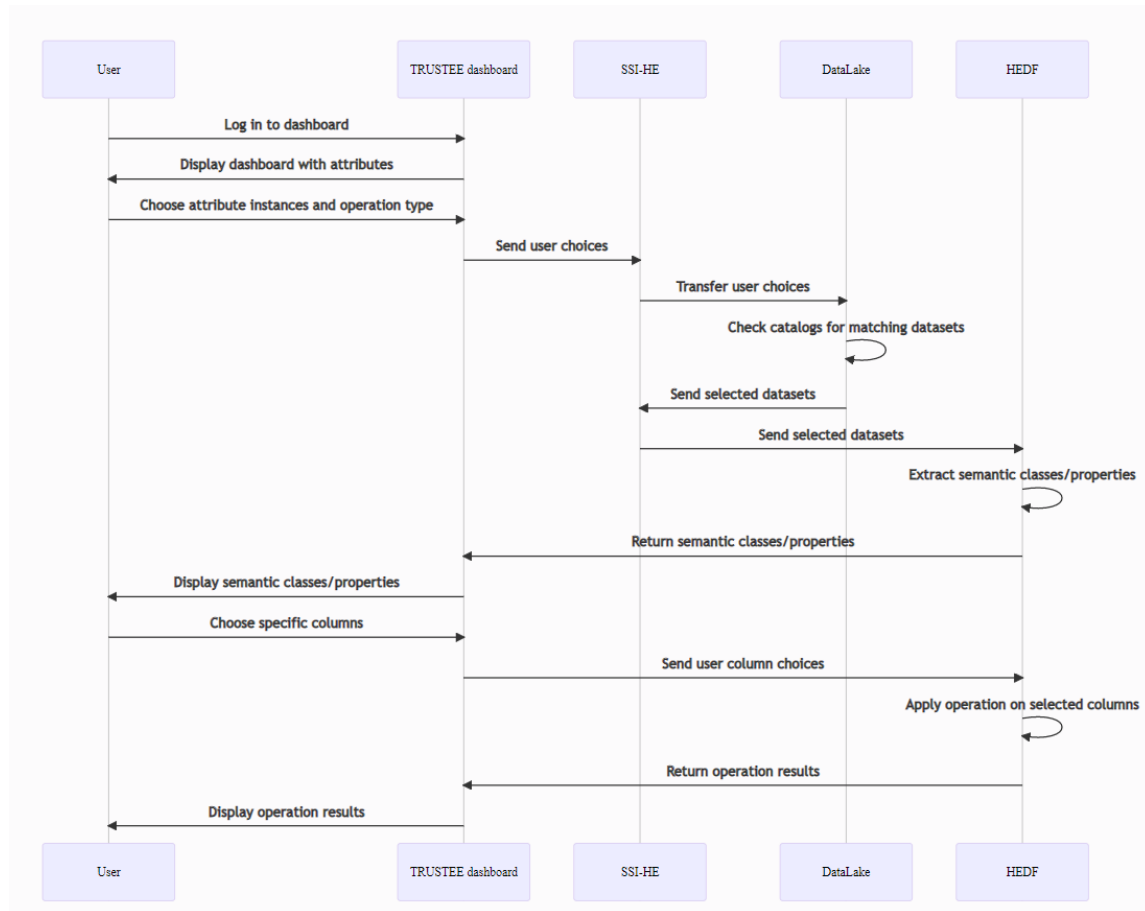


Figure 138: Sequence diagram for operations related to DL

The sequence of interactions begins with the user accessing the system by logging into the DA. Upon successful login, the DA displays the dashboard interface to the user, presenting various attributes and options to choose from. The user then selects specific attribute instances and indicates the type of operation they wish to perform on the data. These selections are transmitted from the DA to the SSI-HE ABB.

Subsequently, the SSI-HE relays the user's attribute choices and operation type to the DL for further processing. The DL receives these selections and engages in a process of checking its data catalogues to identify datasets that contain the requested attributes. Upon identifying the relevant datasets, the DL responds by sending back the selected datasets to the SSI-HE.

The SSI-HE, acting as an intermediary, forwards the chosen datasets to the HEDF. Within the HEDF, the selected datasets undergo processing to extract the semantic classes and properties associated with the data. This information is then communicated back to the DA, where it is presented to the user in an intuitive manner.

As users interact with the DA, they have the opportunity to choose specific columns from the displayed semantic classes and properties. These column selections are relayed from the DA back to the HEDF for further action. Within

the HEDF, the chosen columns are processed, and the requested operation is applied to the selected data columns from the datasets.

Having completed the operation, the HEDF generates the results and returns them to the DA. The DA receives these operation results and effectively displays them to the user. Throughout this series of interactions, the DL functions as a critical component, facilitating the seamless flow of data and enabling operations based on the user's preferences and choices.

The DL plays an important part in the initial demo scenarios that will be investigated for the 1st Pilot stage regarding the Data Consumer persona to be gathered in D5.1 [68]. In the relevant flows, the Data Consumer determines the features of datasets he is interested in (such as domain, location, year, etc.) through the DA. The list of available features as attributes is stored in the DL, in the attributes catalogue sub-component. The DL does not hold any values of the features, only the variable list that is obtained and shown in the DA.

Once the Data Consumer makes his choice, the SSI-HE component receives the features defined and obtains a list of suitable datasets fulfilling the requirements. The DA obtains related details about the suitable datasets from the DL and shows them to the data consumer.

The DL also provides to the HEDF the data providers' endpoints and the operations to be executed only for the data providers that have been considered as suitable based on the user's choices.

These interactions on how the DL component is involved in providing dataset-related information to the different TRUSTEE components are also shown in Figure 139.

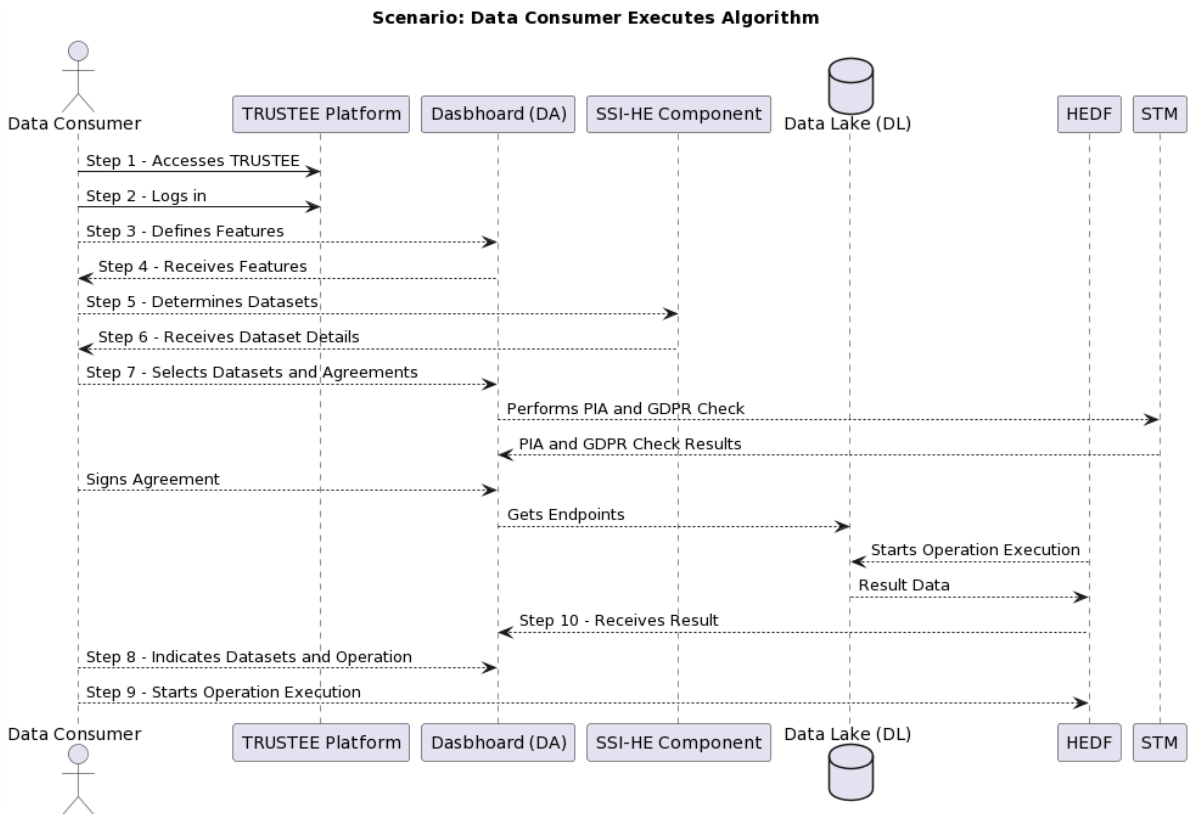


Figure 139 DL interactions for Data Consumer persona scenario

For the Data Provider persona in the initial demo scenario flow, the DL is responsible for storing all new or updated information for the Data Provider (endpoint, any new metadata attribute, indexes) and providing this information to the DA when requested, as shown in Figure 140.

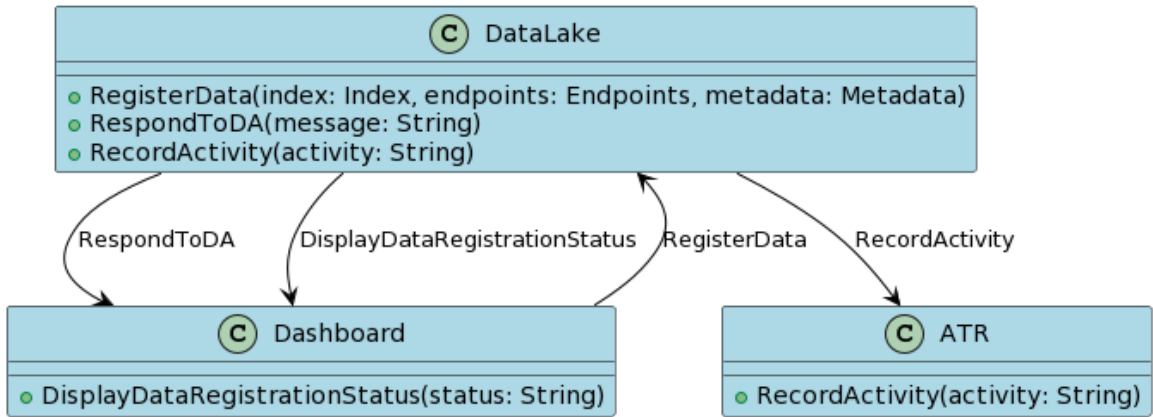


Figure 140 DL data flow for Data Provider scenario

6.5.3. TECHNICAL DESCRIPTION

This section includes information on the different technical aspects relevant to the development and configuration of the DL and its sub-components, along with information about the communication of the DL with other ABBs within the TRUSTEE ecosystem.

6.5.3.1. SEMANTIC ONTOLOGIES REPOSITORY

The DL semantic ontologies repository will host the semantic data models that will be designed to semantically represent each pilot's dataset and AI model provided by the TRUSTEE data providers. The repository will accommodate RDF data storage, such as GraphDB or Fuseki databases.

The TRUSTEE semantic data models that have been designed include:

- Space data model (data for moonquakes, solar index, neural network weights).
- Education data model (data for students, courses, teachers, assessments).
- Health data model (data for eCRF Rapid Covid19 Vitals, encounters, countries).
- Automotive data model (data and models for 2D & 3D scene analysis, Deep visual odometry, Cooperative localization).

The investigation of the Energy semantic data models is still work in progress. The identification of all semantic ontologies, their classes and properties, the relations between the datasets' concepts as well as visual representation of the semantic annotations will be provided in the next version of this live document.

SPACE DOMAIN

The Space domain provides data about moonquake events, a daily solar index documentation and neural network weights. Each dataset has been analyzed, and relevant RDF ontologies have been identified to semantically represent each dataset concept. Where existing ontologies could not be located, custom classes and properties were created. Each dataset is represented with a different data model, and in the end a single holistic semantic data model is designed to cover all Space concepts and foster interoperability and findability of data.

Tables from Table 44 to Table 46 and figures from Figure 141 to Figure 146 show the work done towards the semantic representation of the Space domain.

Table 44: Space dataset: Moonquakes (original)

Dataset concept	schema	Sub - concepts	Relevant Ontology (class, property)	Semantic
Eventinfo		public ID	quakeml:Event quakeml:publicID (<u>QuakeML Ontology</u>)	
eventType			quakeml:eventType (<u>QuakeML Ontology</u>)	
eventDescription			quakeml:description (<u>QuakeML Ontology</u>)	
Picks		stationCode, networkCode, phase, time, publicID, channelCode, locationCode	quakeml:Pick quakeml:stationCode quakeml:networkCode, quakeml:phase quakeml:time	

		quakeml:publicID quakeml:channelCode quakeml:locationCode (<u>QuakeML Ontology</u>)
amplitudes	publicID, pickID, genericAmplitude, period, unit	quakeml:Amplitude quakeml:publicID, quakeml:pickID, quakeml:genericAmplitude, quakeml:period, quakeml:unit



Figure 141: Space data model for moonquakes dataset (OWLGrEd)

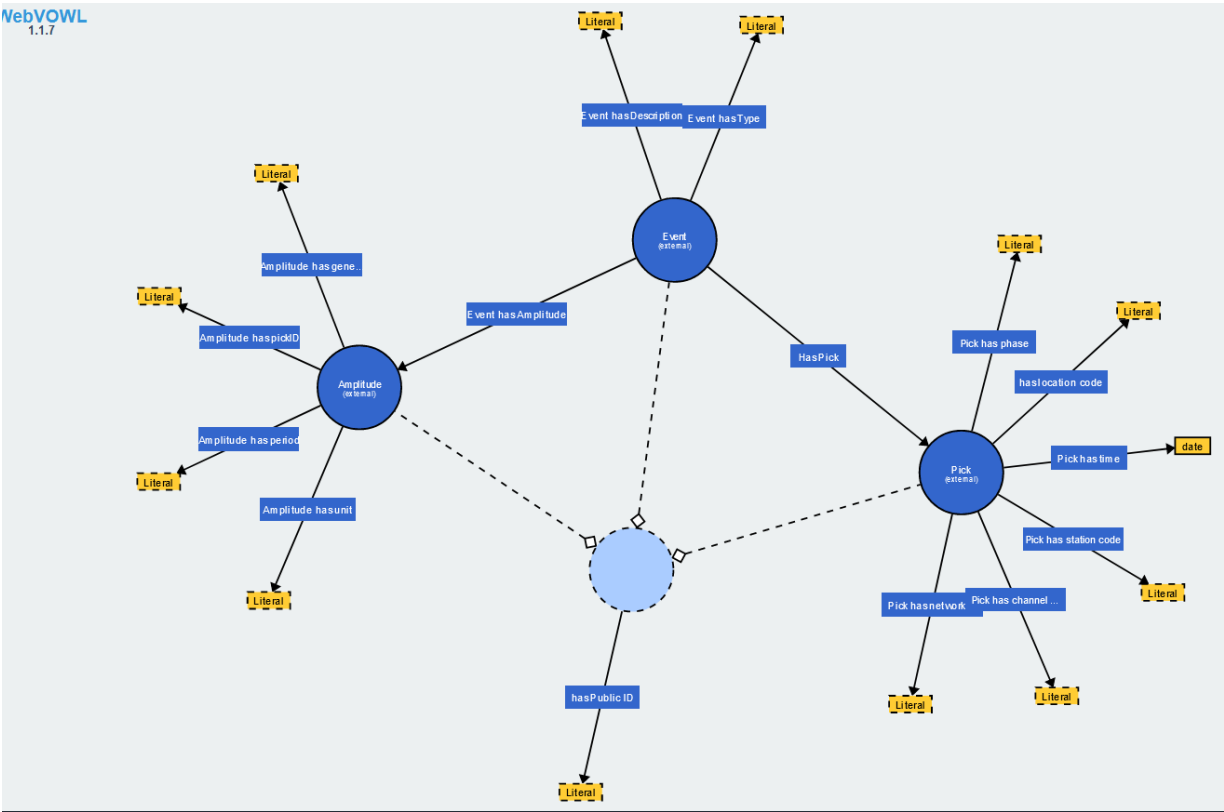


Figure 142: Space data model for moonquakes dataset (WebVOWL)

Table 45: Weights of neural networks dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
weights	<u>Custom ontology</u> nn:Weight

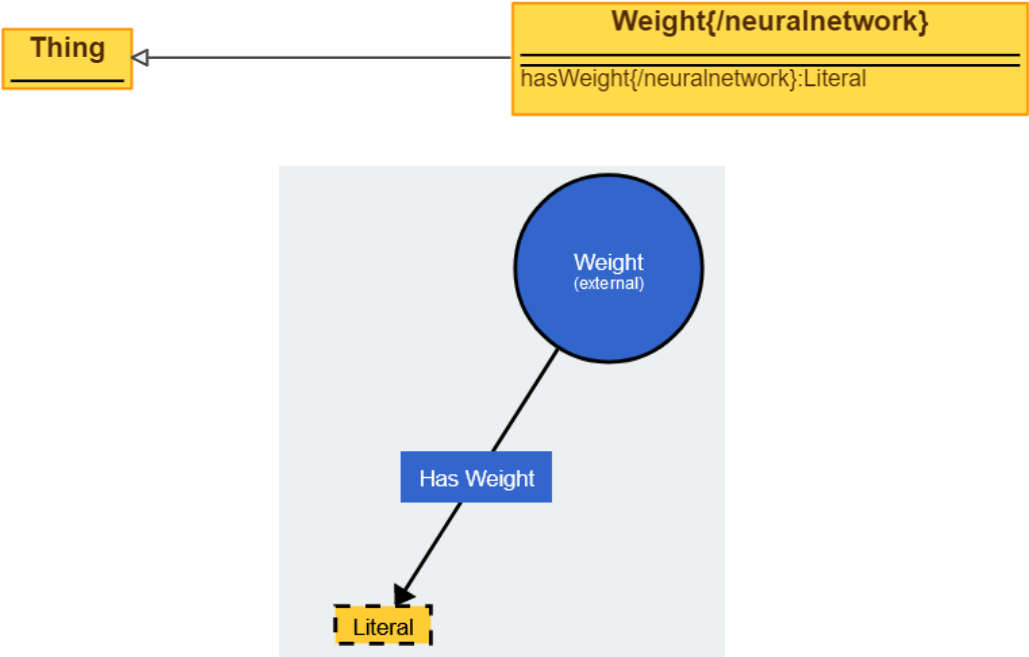


Figure 143: Space weights dataset semantic representation

Table 46: Solar index dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Daily solar index	<u>Custom ontology</u> solar:DailySolarIndex solar:date solar:indexValue

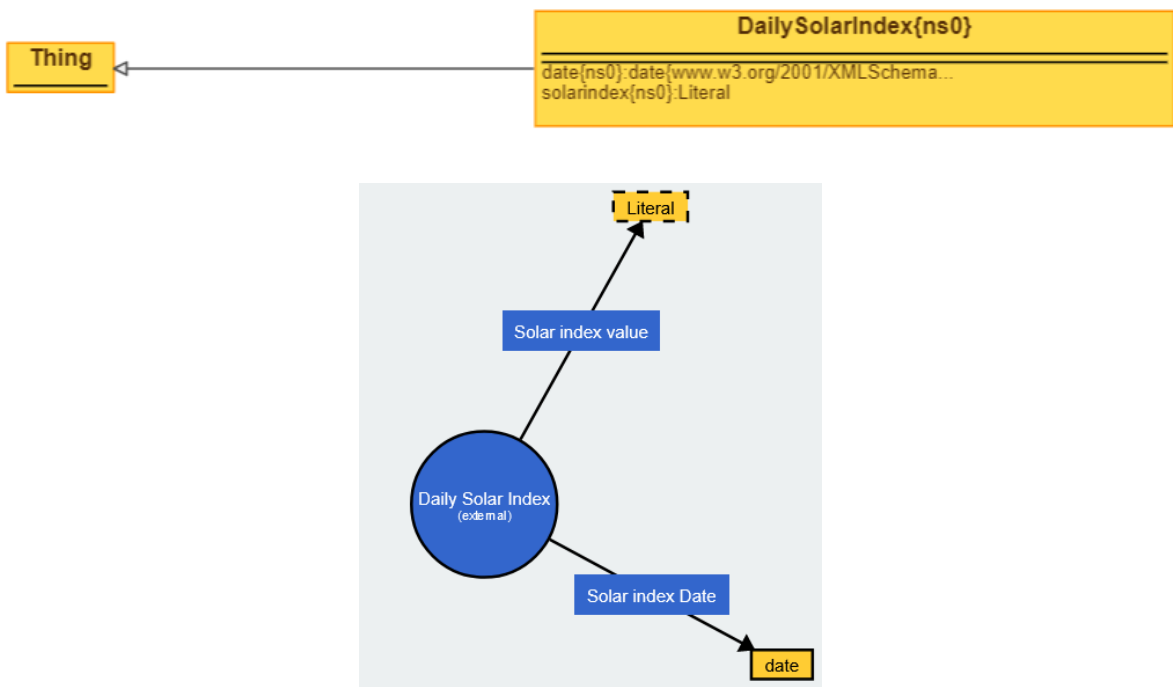


Figure 144: Solar index dataset semantic representation

The Figure 145 and Figure 146 show the holistic data model that represents all concepts provided by the Space data providers within the context of TRUSTEE.

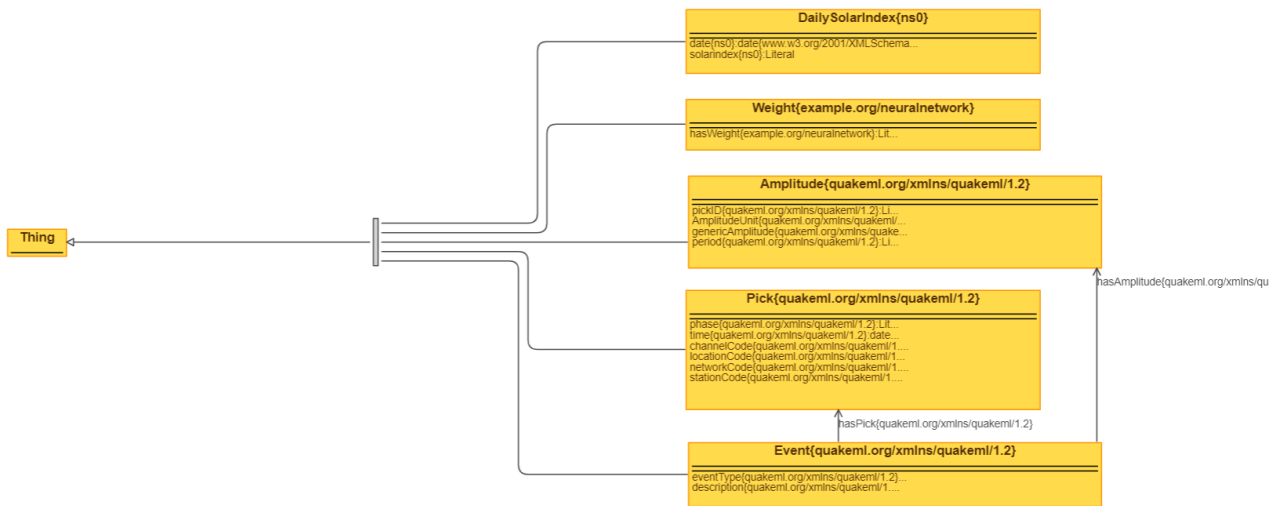


Figure 145: Holistic Space data model semantic representation

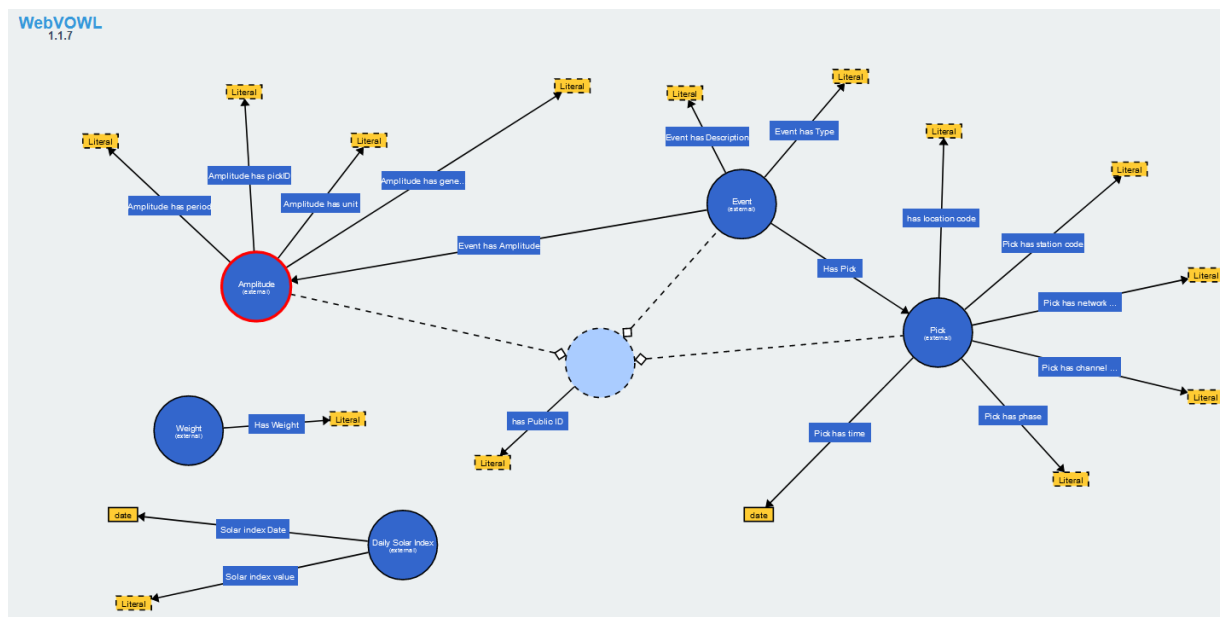


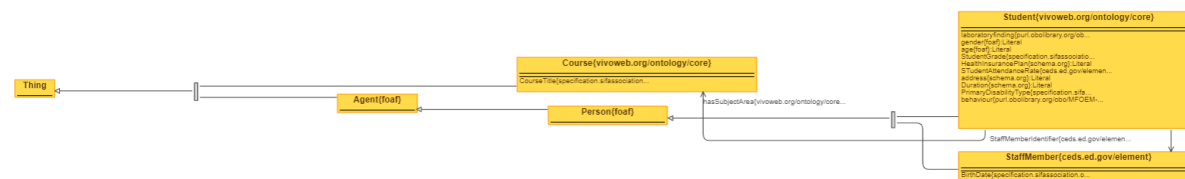
Figure 146: Holistic Space data model relational representation

EDUCATION DOMAIN

The education datasets include information about students, the courses they participate, their assessment performances on different courses, as well as the teachers for each course. In total, 8 different datasets have been provided:

- Students school data
- Students data
- Schools data
- Subjects data
- Classes data
- Assessments data
- Grades data
- Students' grades data

Each dataset was studied with focus on the different concepts they represent. A semantic data model was designed for each dataset, which led to the creation of a holistic, generic data model that encompasses and describes all datasets concepts. This information is depicted semantically as shown in figures from Figure 147 to Figure 162, using existing semantic ontologies to adhere to FAIR principles of re-usability.



Tables from Table 47 to

Table 47: Student school dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Student ID	Foaf:account (<u>FOAF: Friend of a Friend Ontology</u>) Vivo:Student (<u>VIVO Core Ontology</u>)
Name	Foaf:name (<u>FOAF Ontology</u>) Vivo:Name (<u>VIVO Core Ontology</u>)

TRUSTEE

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Age	Foaf:age (FOAF Ontology)
Gender	Foaf:gender (FOAF Ontology)
Address	Schema: address (SCHEMA.org Ontology) Vivo:address (VIVO Core Ontology)
Maths grade	SIF: StudentGrade (SIF Ontology)
English grade	SIF: StudentGrade (SIF Ontology)
Science grade	SIF: StudentGrade (SIF Ontology)
Behaviour	EMO: behavior (EMO Ontology)
Attendance	CEDS:StudentAttendanceRate (CEDS Ontology)
SEN	SIF:PrimaryDisabiityType (SIF Ontology)
EHCP	Schema: HealthInsurancePlan (Schema.org Ontology)
Hours on Device	Schema:Duration (Schema.org Ontology)
Covid Status	Covid:laboratoryfinding (COVID-19 Ontology)
Classroom Type	SIF:CourseTitle (SIF Ontology) Vivo:Course (VIVO Ontology)
Teacher ID	Foaf:account (foaf ontology) CEDS: StaffMemberIdentifier (CEDS Ontology)
Teacher Full Name	Foaf:name (FOAF Ontology) Vivo:Name (VIVO Core Ontology) CEDS:OtherName (CEDS Ontology)
Teacher DOB	SIF:BirthDate (SIF Ontology)
Semester	CEDS:Semester (CEDS Ontology) Vivo:AcademicTerm (VIVO Ontology)
Year	Vivo:AcademicYear (VIVO Ontology) CEDS: AcademicYearDesignator (CEDS Ontology)

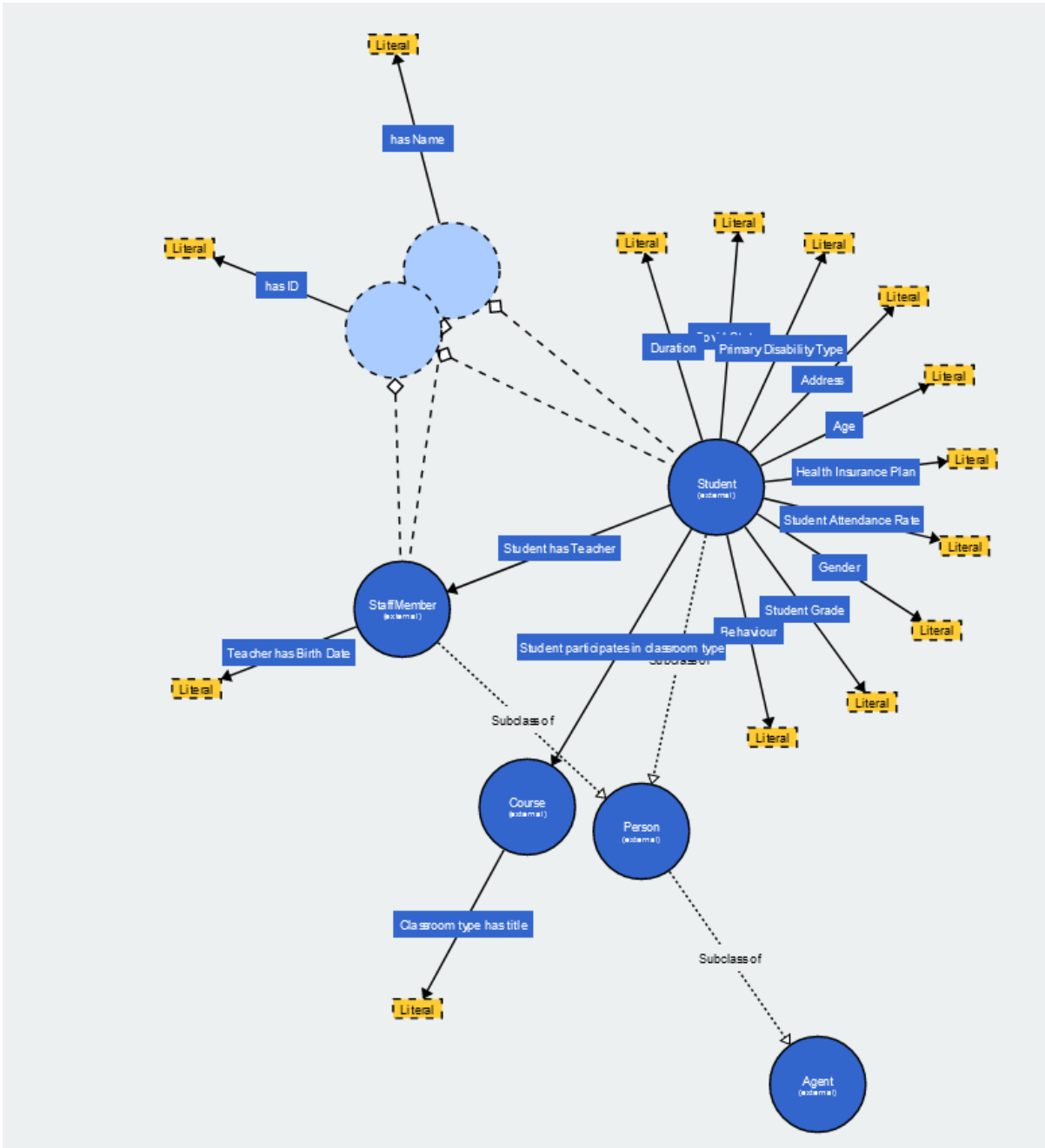


Figure 148: Students school dataset relational representation

Table 48: Students dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Student ID	foaf:account
First name	foaf:givenName
Last name	foaf:familyName
Date of birth	schema:birthDate

Gender	foaf:gender
Ethnicity	schema:ethnicity
Home language	schema:language
Free school meals	ex:freeSchoolMeals (<u>Custom ontology</u>)
Special education needs	ex:specialEducationNeeds (<u>Custom ontology</u>)

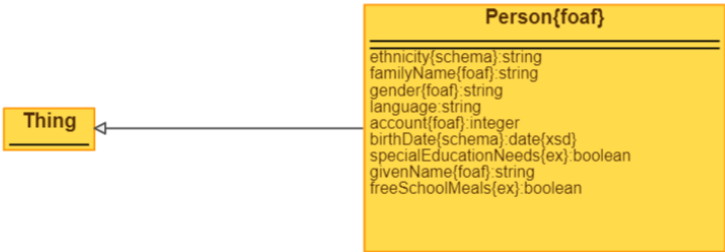


Figure 149: Simplified visualization of semantic schema for Students dataset

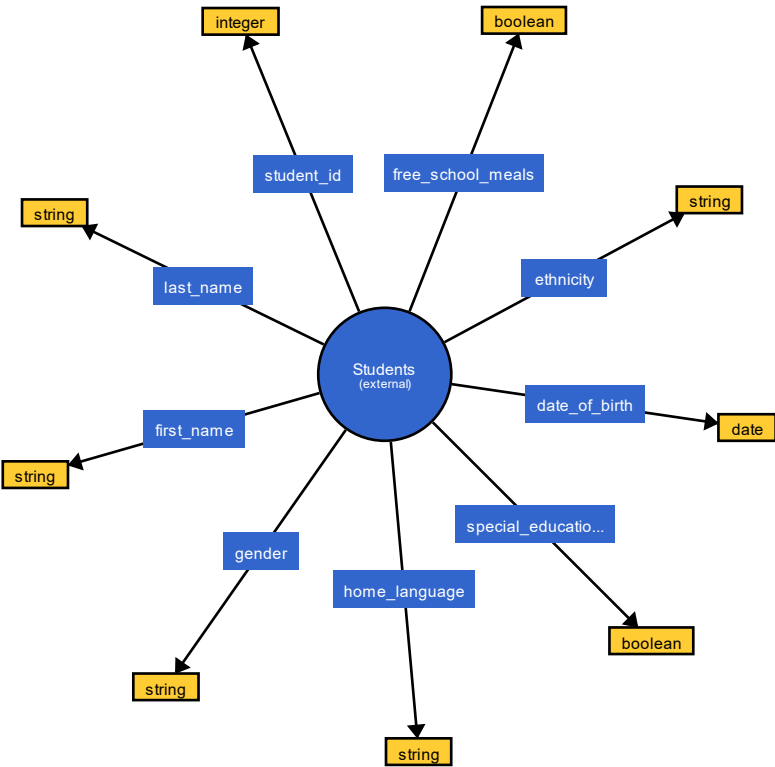


Figure 150: Relational semantic visualization of Students dataset

Table 49: Schools dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
School ID	schema:identifier (<u>Schema ontology</u> class Schema:Organization)
School name	schema:name (<u>Schema ontology</u> class Schema:Organization) vivo:organizationName (<u>VIVO ontology</u>)

School address	schema:address (Schema ontology) vivo:address (VIVO ontology)
Head Teacher	schema:employee (Schema ontology) vivo:hasPrincipalInvestigator (VIVO ontology)
Number of students	ex:numberOfStudents (Custom ontology) sif:Enrollment (SIF ontology)

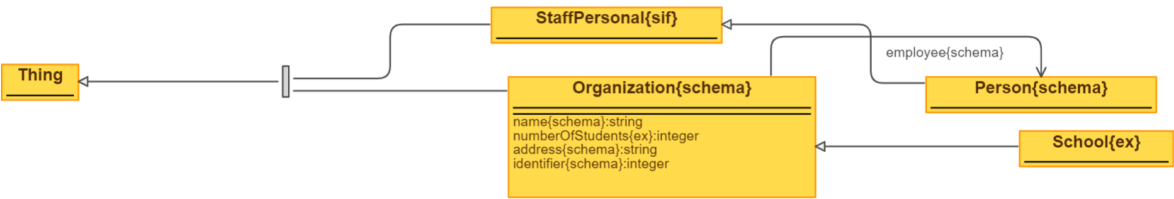


Figure 151: Simplified semantic diagram of dummy Schools dataset

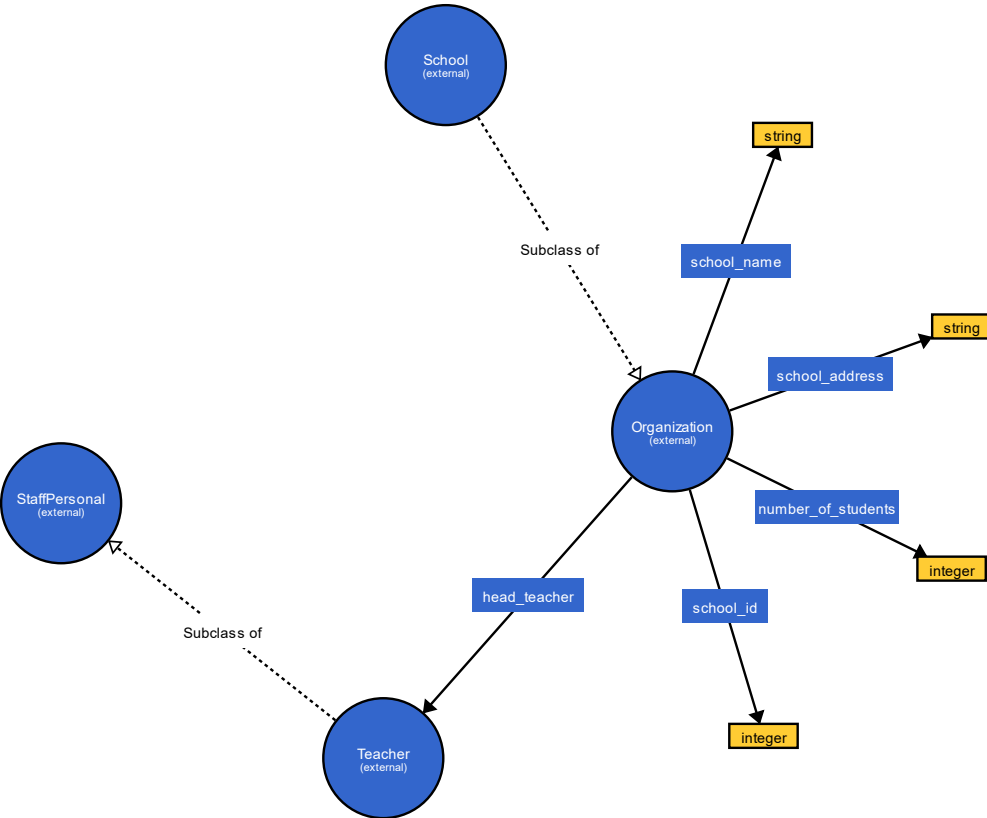


Figure 152: Relational semantic visualization of Schools dataset

Table 50: Subjects dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
------------------------	--

Subject ID	schema:identifier (Schema ontology)
	vivo:localCourseId (VIVO ontology)
	ceds:CourseIdentifie (CEDS ontology)
Subject name	schema:name (Schema ontology)
	vivo:courseName (VIVO ontology)
	ceds:CourseTitle (CEDS ontology)

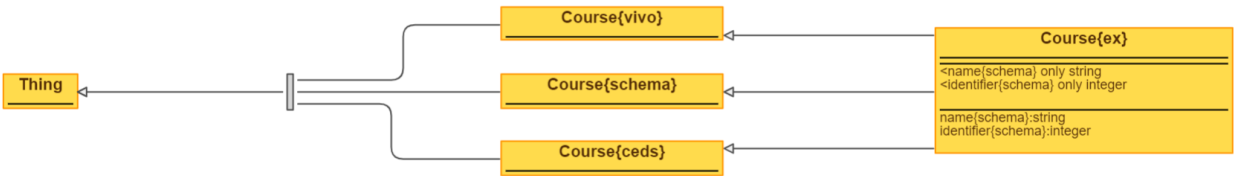


Figure 153: Simplified semantic visualization of Subjects dataset

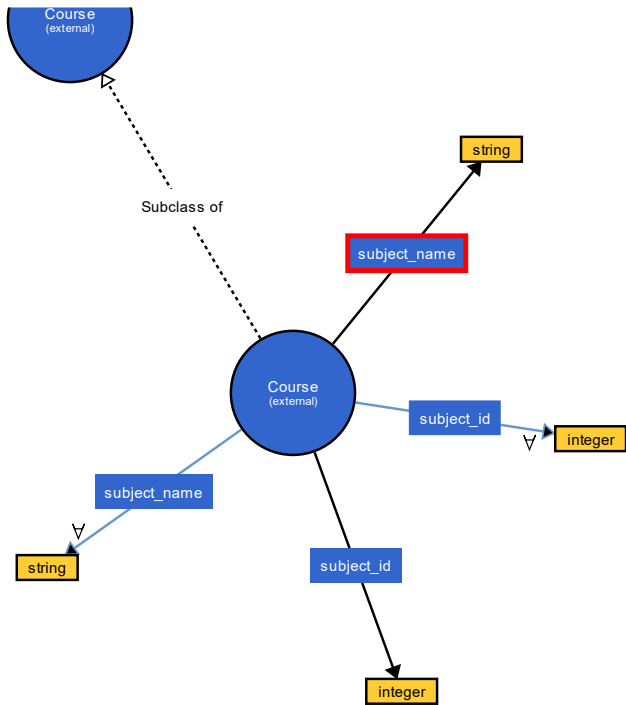


Figure 154: Relational visualization of Subjects dataset

Table 51: Grades dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Grade ID	ceds:StudentAssessmentIdentifier (class) ceds:CourseAcademicGrade (CEDS ontology)
Grade value	ceds:StudentAssessmentScore (CEDS ontology)

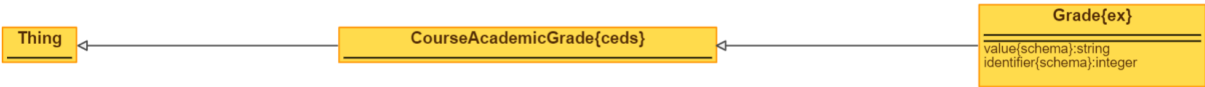


Figure 155: Simplified semantic visualization of Grades dataset

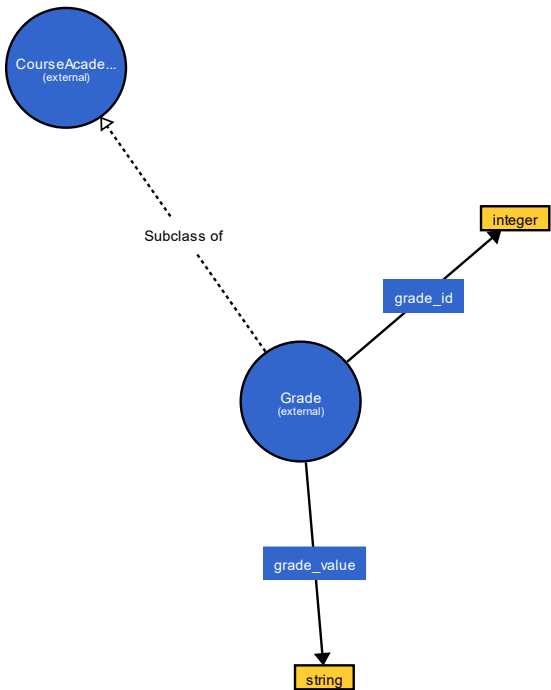


Figure 156: Relational visualization of Grades dataset

Table 52: Classes dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Class ID	sif:LocalId (SIF ontology)
Class name	vivo:courseName (VIVO ontology)
Teacher name	vivo:teacherRole (VIVO ontology) Foaf: Person
School id	schema:identifier (Schema ontology class Schema:Organization) [foreign key to School id from Schools table]

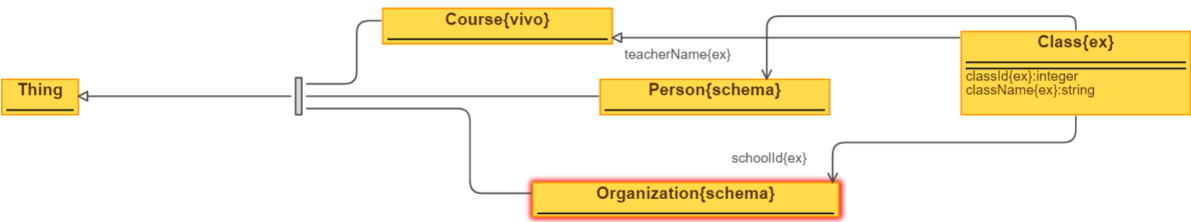


Figure 157: Simplified semantic visualization of Classes dataset

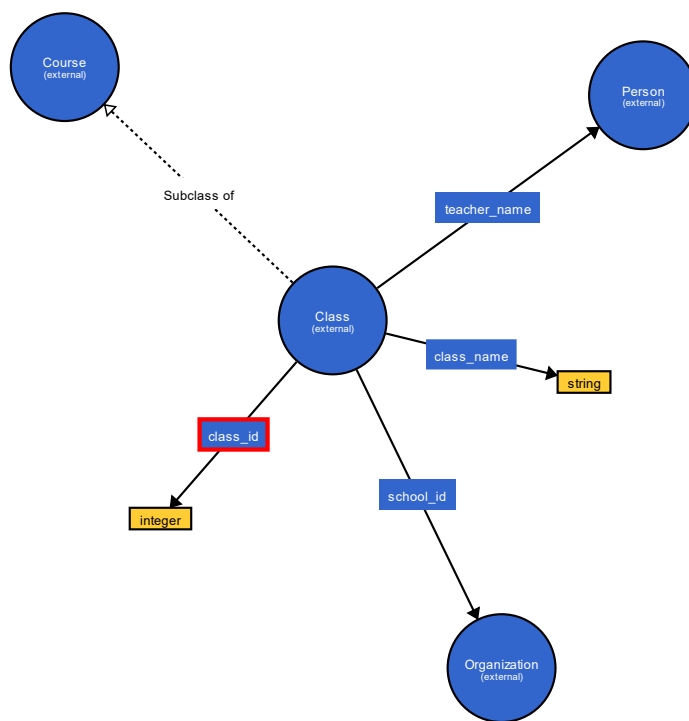


Figure 158: Relational visualization of Classes dataset

Table 53: Assessment table

Dataset schema concept	Relevant Semantic Ontology (class, property)
Assessment ID	sif:LocalId (<u>SIF ontology</u> class sif:Assessment)
Assessment name	sif:AssessmentName (<u>SIF ontology</u>)
Subject id	sif:CourseCode (<u>SIF ontology</u>) [foreign key to subject id from Subjects table]
Class id	sif:LocalId (<u>SIF ontology</u>) [foreign key to class id from Classes table]

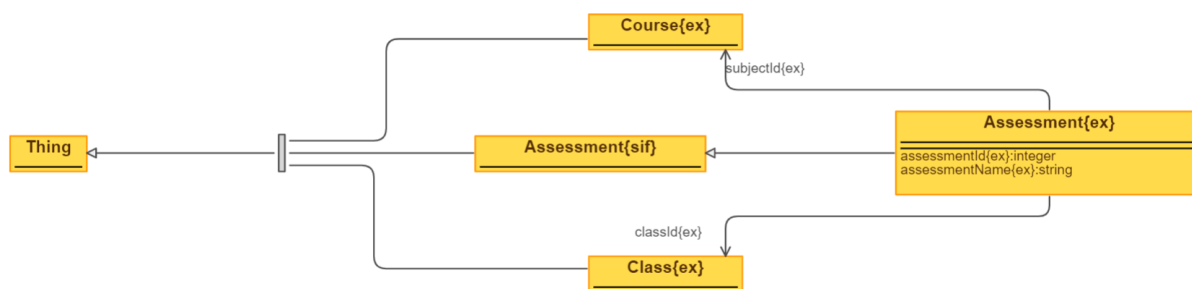


Figure 159: Simplified semantic visualization of Assessments table

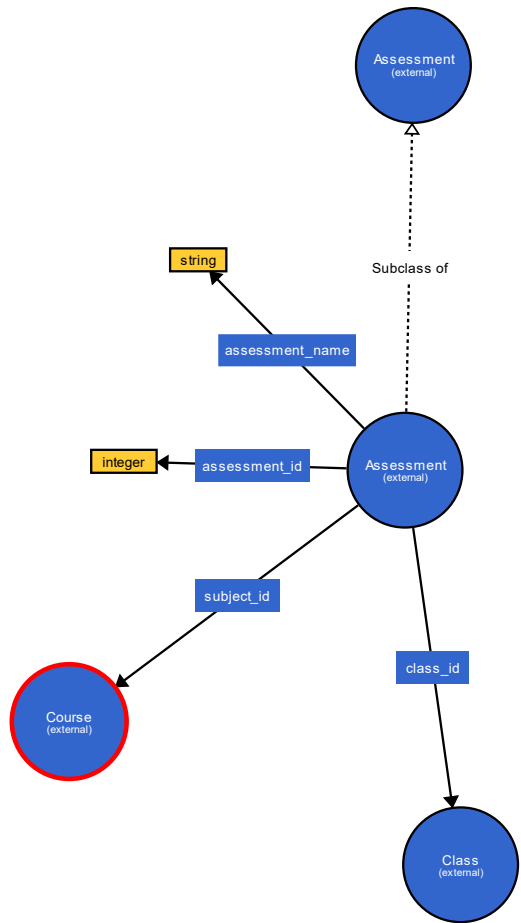


Figure 160: Relational visualization of Assessments dataset

Table 54: Student_grades dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Student ID	sif:LocalId (<u>SIF ontology</u> class sif:Assessment)
Assessment name	sif:AssessmentName (<u>SIF ontology</u>)
Grade id	sif:CourseCode (<u>SIF ontology</u>) [foreign key to subject id from Subjects table]

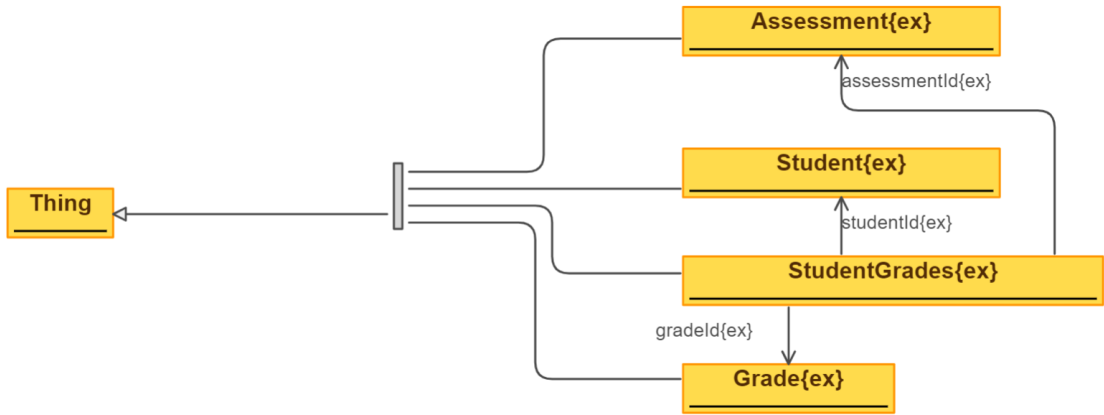


Figure 161: Simplified semantic visualization of Student_grades dataset

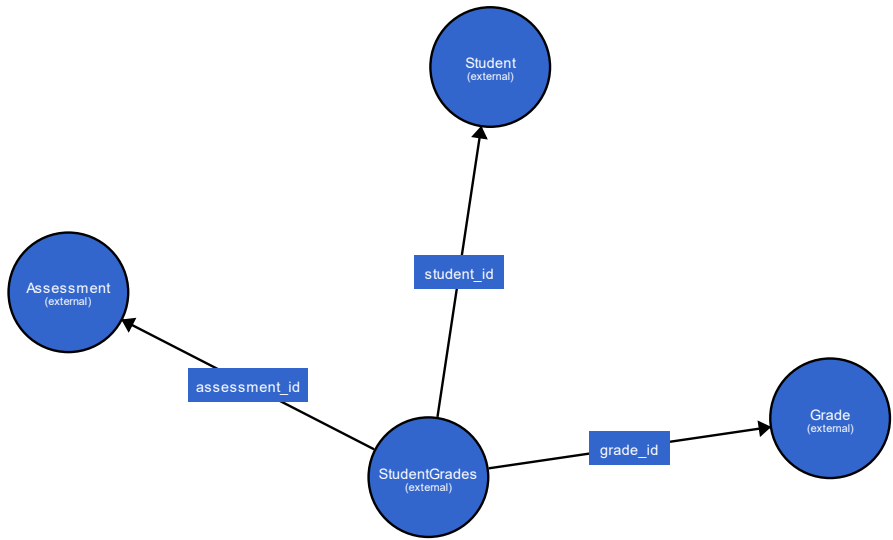


Figure 162: Relational visualization of Student_grades dataset

HEALTH DOMAIN

The Health domain data consist of 3 different datasets, regarding electronic case reports, encounters of patients, and countries. Each dataset has been analyzed and semantically annotated with a semantic data model that re-uses existing RDF ontologies mainly from the Health domain. The initial semantic representation of all the data models resulted in the design of a holistic Health semantic data model that encompasses all dataset concepts. Tables from Table 55 to Table 57 show the work done towards the design of this data model.

Table 55: eCRF_Rapid_Covid19_Module2_2A_Vitals dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
"eCRF ID"	<u>Dublin Core Ontology</u> dcterms:identifier
"Participant ID"	<u>FOAF: Friend of a Friend Ontology</u>

	foaf:account
"Date of follow up"	<u>Dublin Core Ontology</u> dcterms:date
"Temperature (°C)"	<u>Health Level Seven International (HL7) Ontology</u> vital:BodyTemperature
"Heart rate (beats/min)"	<u>Health Level Seven International (HL7) Ontology</u> vital:HeartRate vital:numericValue
"Respiratory rate (breaths/min)"	<u>Health Level Seven International (HL7) Ontology</u> vital:RespiratoryRate vital:numericValue
"Blood Pressure (systolic mmHg)"	<u>Health Level Seven International (HL7) Ontology</u> vital:BloodPressure vital:numericValue
"Blood Pressure (diastolic mmHg)"	<u>Health Level Seven International (HL7) Ontology</u> vital:BloodPressure vital:numericValue
"Severe dehydration"	<u>SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) Ontology</u> snomed: Dehydration snomed:hasValue
"Sternal capillary refill time > 2 seconds"	<u>SNOMED CT Ontology</u> snomed: Capillary refill snomed:hasValue
"Oxygen saturation (%)"	<u>SNOMED CT Ontology</u> snomed: Oxygen saturation measurement snomed: Oxygen saturation measurement snomed:hasValue
"Saturation measured on"	<u>Schema Ontology</u> Schema:MedicalEntity
"A V P U Value"	<u>SNOMED CT Ontology</u> snomed:AVPU snomed:AVPU snomed:hasValue
"Glasgow Coma Score (GCS/15)"	<u>SNOMED CT Ontology</u> snomed:Glasgow coma scale, 15 snomed:hasValuesnomed:Glasgow coma scale, 15 snomed:hasValue

Table 56: Encounter dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
Encounter ID	<u>HL7 Ontology</u> vital:Identifier <u>Dublin Core Ontology</u> dcterms:identifier <u>FOAF: Friend of a Friend Ontology</u> foaf:account
Encounter Type ID	<u>HL7 Ontology</u> vital: ActEncounterType
Positive Case ID	<u>HL7 Ontology</u>

	vital:Identifier <u>Dublin Core Ontology</u> dcterms:identifier <u>FOAF: Friend of a Friend Ontology</u> foaf:account
Contact ID	<u>FOAF: Friend of a Friend Ontology</u> foaf:account <u>Dublin Core Ontology</u> dcterms:identifier
Location Name	<u>Schema.org Ontology</u> Schema:place
Location Address	<u>FOAF: Friend of a Friend Ontology</u> Foaf:Address <u>Schema.org Ontology</u> Schema:Address
From	<u>Dublin Core Ontology</u> dcterms:date
To	<u>Dublin Core Ontology</u> dcterms:date

Table 57: Countries dataset

Dataset schema concept	Relevant Semantic Ontology (class, property)
City	<u>Schema Ontology</u> Schema:City Schema:name
Country	<u>Schema Ontology</u> Schema:Country Schema:name

AUTOMOTIVE DOMAIN

The automotive domain includes 5 different types of datasets & models. The semantic representation of each dataset and model will include a set of labels that will improve their discoverability for HE operations within TRUSTEE. The different types of data and models available are:

- 2D Scene Analysis
- 3D Scene Analysis
- Deep Visual Odometry
- Cooperative Localisation
- Lidar super resolution

All models that will be applied within the TRUSTEE project will be semantically annotated using the characteristics shown in Table 58.

Table 58: Automotive model labels

Dataset schema concept	Relevant Semantic Ontology (class, property)
input	<u>Schema Ontology</u> ³⁴ PropertyValueSpecification <u>XSD ontology</u> xsd:String
output	<u>Schema Ontology</u> ³⁵ PropertyValueSpecification <u>XSD ontology</u> xsd:String
pre-processing	<u>PROV-O Ontology</u> ³⁶ Activity <u>XSD ontology</u> xsd:String
implementation	<u>Dbpedia Ontology</u> ³⁷ Implementation <u>XSD ontology</u> xsd:String
number_of_parameters	<u>Custom ontology</u> xsd:nonNegativeInteger
inference_time	<u>Schema.org ontology</u> ³⁸ Time
execution_time	<u>Schema.org ontology</u> ³⁹ Time

Figure 163 and Figure 164 show a visual representation of the semantic annotation for ML models within the automotive domain.

³⁴ <https://schema.org/PropertyValueSpecification>

³⁵ <https://schema.org/PropertyValueSpecification>

³⁶ <https://www.w3.org/TR/prov-o/#Activity>

³⁷ <https://dbpedia.org/page/Implementation>

³⁸ <https://schema.org/Time>

³⁹ <https://schema.org/Time>

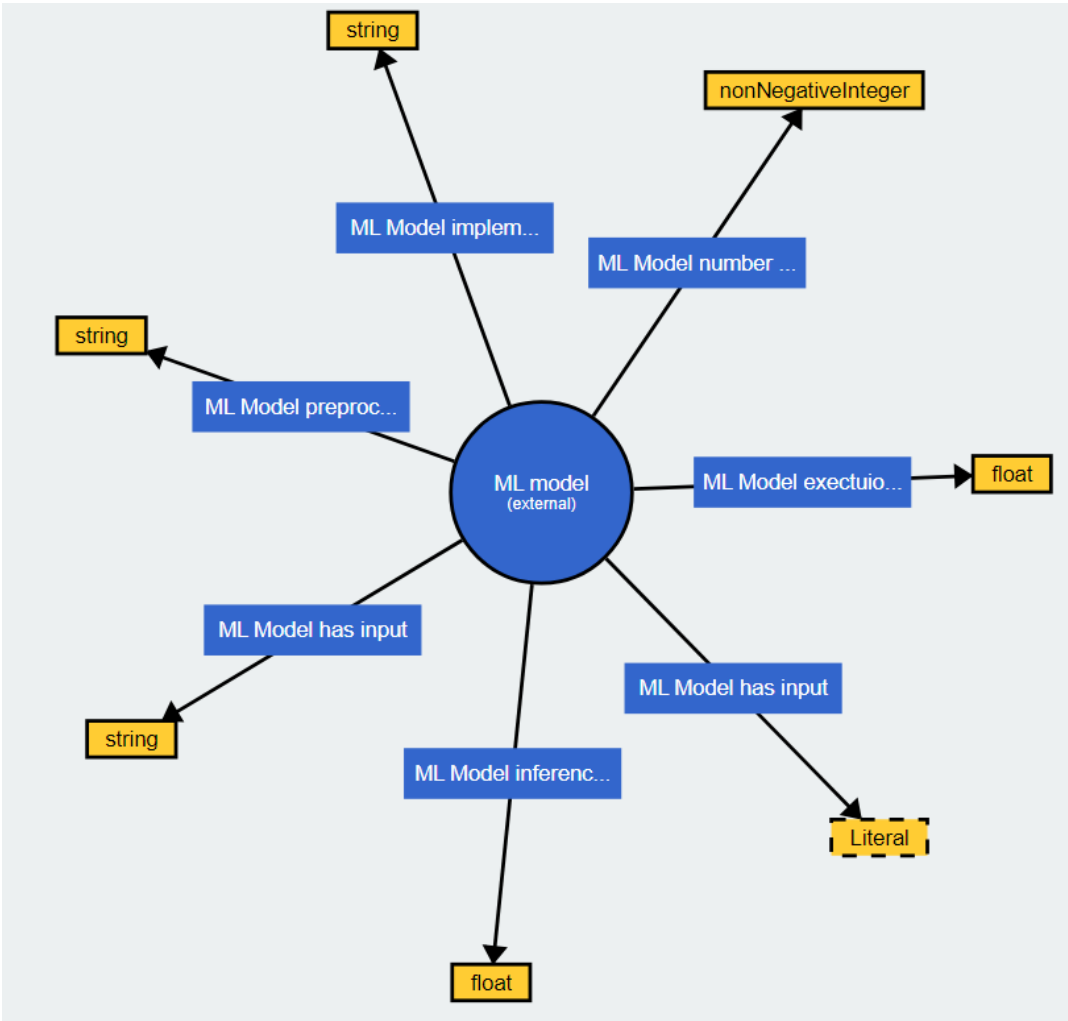


Figure 163: Relational visualization of Automotive models

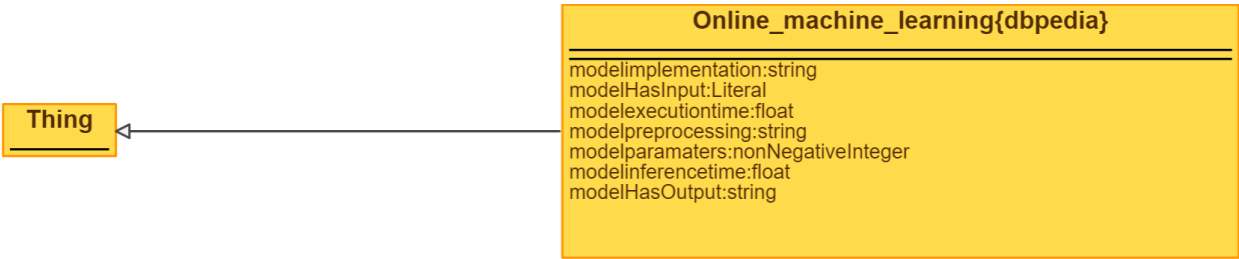


Figure 164: Simplified semantic visualization of Automotive ML models

The 2D Scene Analysis dataset consists of training images and test images. The dataset also includes a label (.txt) for each pointcloud and lidar calibration files. The labels for the 2D Scene Analysis dataset and model are gathered in Table 59, Figure 165 and Figure 166.

Table 59: Automotive 2D Scene Analysis dataset labels

Dataset schema concept	Relevant Semantic Ontology (class, property)
field of view (degrees)	<u>dbpedia Ontology</u> ⁴⁰ dbpedia:Field_of_view
weather	<u>BIMERR Weather Ontology</u> ⁴¹ bimerr:AtmosphericPhenomenon
traffic	<u>Dbpedia Ontology</u> ⁴² dbpedia:Traffic
environment	<u>Schema.org Ontology</u> ⁴³ schema:Place
speed	<u>Schema.org Ontology</u> ⁴⁴ schema:speed
detections	<u>SOSA Ontology</u> ⁴⁵ sosa:Observation

⁴⁰ https://dbpedia.org/page/Field_of_view

⁴¹ <https://bimerr.iot.linkeddata.es/def/weather/#AtmosphericPhenomenon>

⁴² <https://dbpedia.org/page/Traffic>

⁴³ <https://schema.org/Place>

⁴⁴ <https://schema.org/speed>

⁴⁵ <https://www.w3.org/TR/vocab-ssn/#SOSAObservation>

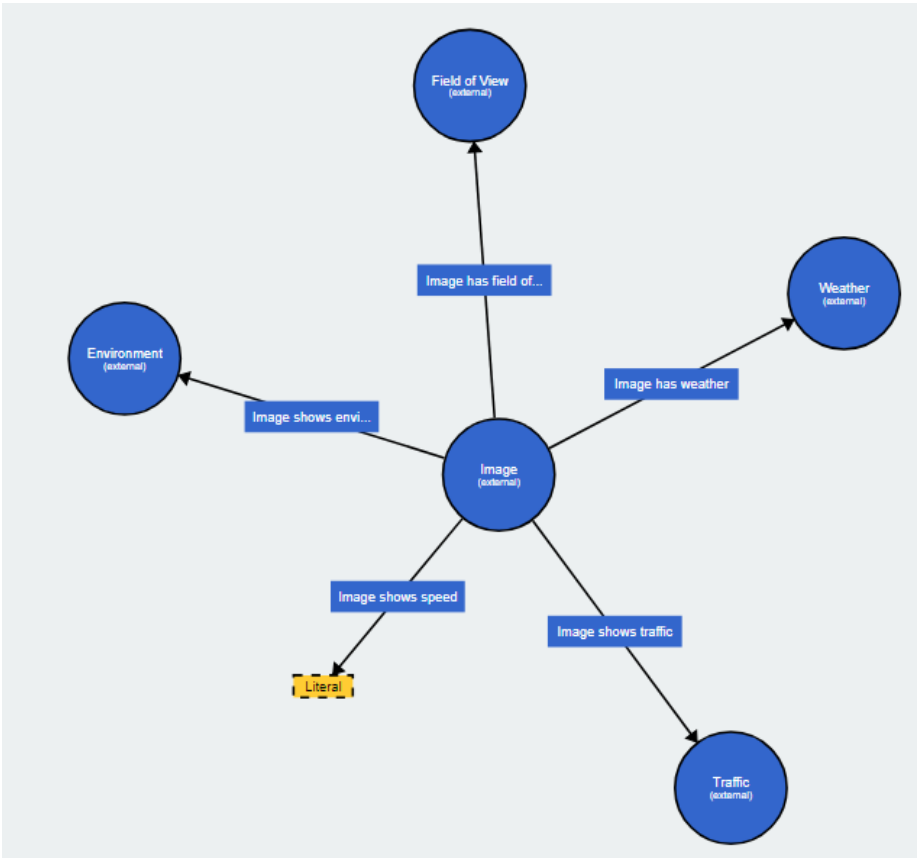


Figure 165: Relational visualization of 2D Scene Analysis dataset

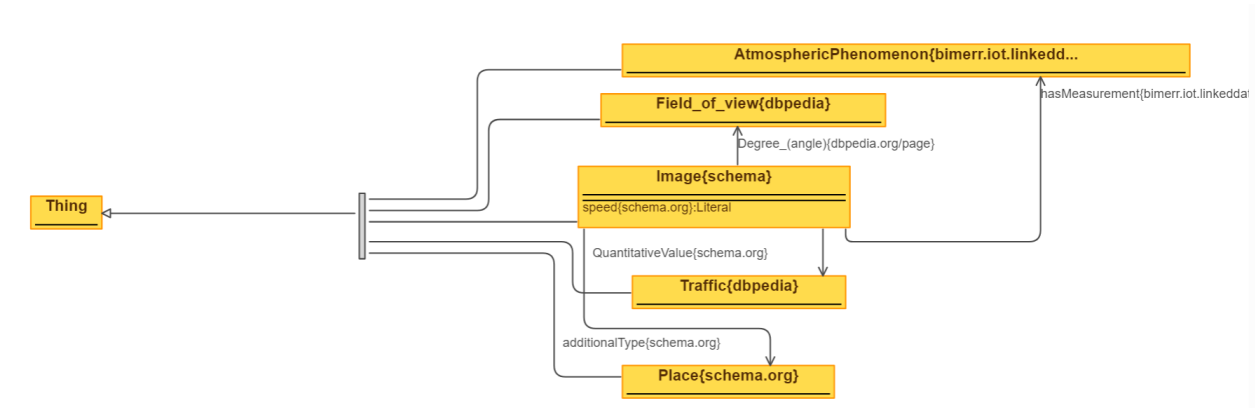


Figure 166: Simplified semantic visualization of 2D Scene Analysis dataset

Another set of the Automotive domain includes datasets for 3D Scene Analysis. These include training point clouds and test point clouds. The relevant Table for the 3D Scene Analysis is gathered in Table 60, Figure 167 and Figure 168.

Table 60: Automotive 3D Scene Analysis dataset labels

Dataset schema concept	Relevant Semantic Ontology (class, property)
------------------------	--

horizontal_resolution	<u>Exif Ontology</u> ⁴⁶ xResolution
vertical_resolution	<u>Exif Ontology</u> ⁴⁷ yResolution
lower_angle	<u>Dbpedia Ontology</u> ⁴⁸ Angle_of_view
type_of_lidar_sensor	<u>SSN Ontology</u> ⁴⁹ Sensor
field of view (degrees)	<u>dbpedia Ontology</u> ⁵⁰ dbpedia: Field_of_view
weather	<u>BIMERR Weather Ontology</u> ⁵¹ AtmosphericPhenomenon
traffic	<u>Dbpedia Ontology</u> ⁵² Traffic
environment	<u>Schema.org Ontology</u> ⁵³ Place
speed	<u>Schema.org Ontology</u> ⁵⁴ speed
detections	<u>SOSA Ontology</u> ⁵⁵ SOSAObservation

⁴⁶ <https://www.w3.org/2003/12/exif/#vocabulary>

⁴⁷ <https://www.w3.org/2003/12/exif/#vocabulary>

⁴⁸ https://dbpedia.org/page/Angle_of_view

⁴⁹ <https://www.w3.org/TR/vocab-ssn/#SOSASensor>

⁵⁰ https://dbpedia.org/page/Field_of_view

⁵¹ <https://bimerr.iot.linkeddata.es/def/weather/#AtmosphericPhenomenon>

⁵² <https://dbpedia.org/page/Traffic>

⁵³ <https://schema.org/Place>

⁵⁴ https://schema.org/docs/search_results.html?q=speed

⁵⁵ <https://www.w3.org/TR/vocab-ssn/#SOSAObservation>

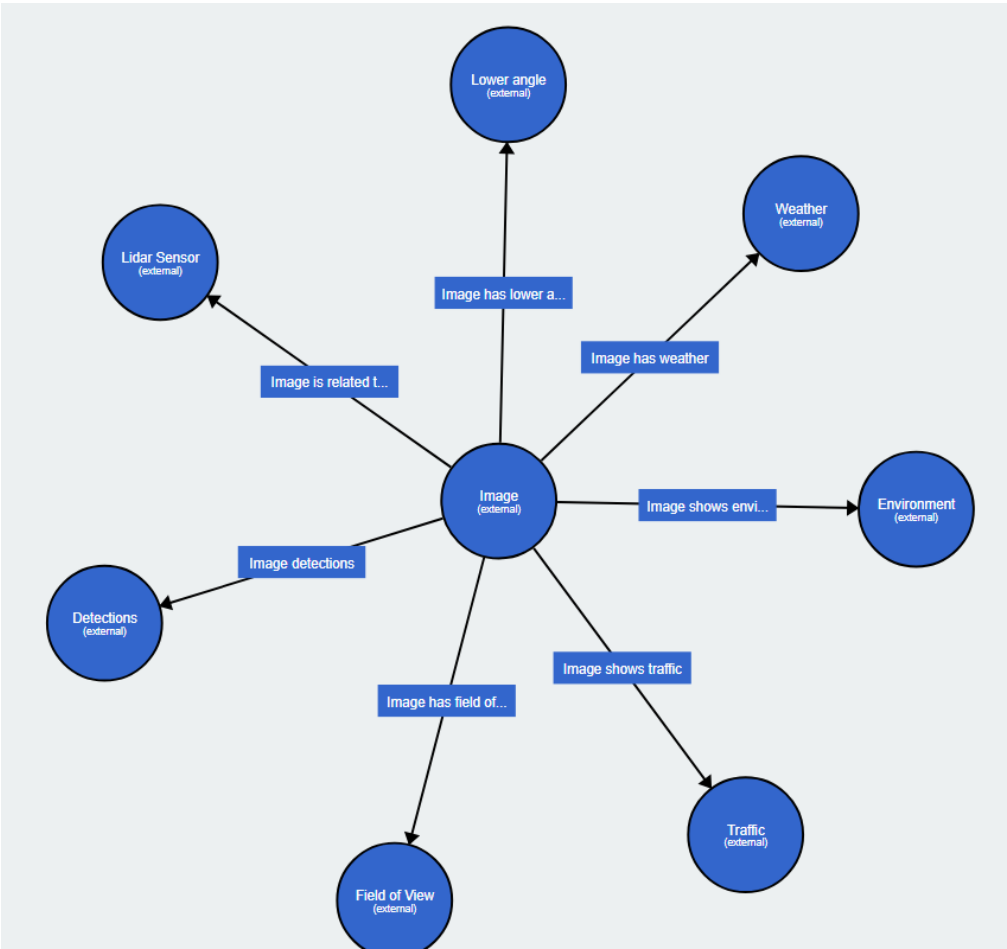


Figure 167: Relational visualization of 3D Scene Analysis dataset

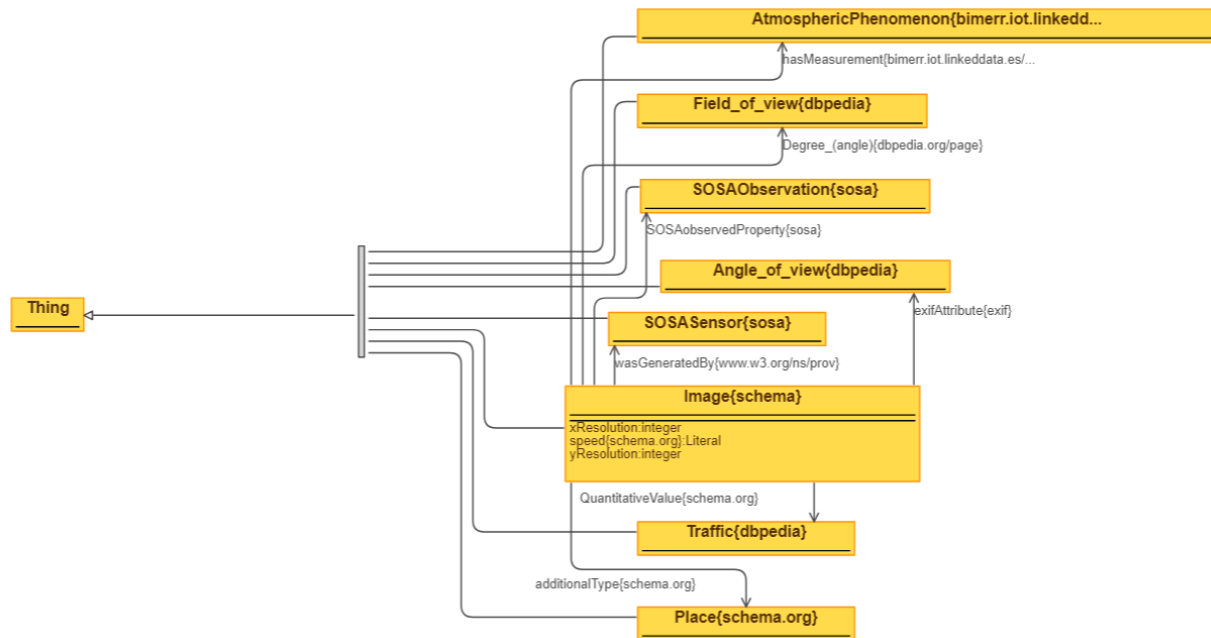


Figure 168: Simplified semantic visualization of 3D Scene Analysis dataset

The 3rd dataset in the Automotive domain includes data about visual odometry. It is comprised of sequences of images and for every sequence there is file (txt) containing the timestamps and the ground truth poses of the ego vehicle. This dataset is semantically annotated with the same labels / characteristics as the 2D Scene Analysis dataset, as shown in Table 59.

The 4th dataset regards super resolution of lidar sensors, where the dataset contains pairs of high and low resolution 3D point clouds or pairs of the corresponding high and low resolution range images. This dataset is semantically annotated with the same labels / characteristics as the 3D Scene Analysis dataset, as shown in Table 60.

The final dataset available for the Automotive pilot regards time indexes of the simulation, the ground truth, the noisy and re-estimated by KalmanNet 3D position of vehicles inside the CARLA map. The semantic annotation of the characteristics that describe the dataset are similar to the previous described in the above section, and more specifically include the concepts shown in Table 61.

Table 61: Automotive Cooperative localization & KalmanNet dataset labels

Dataset schema concept	Relevant Semantic Ontology (class, property)
weather	<u>BIMERR Weather Ontology</u> AtmosphericPhenomenon
traffic	<u>Dbpedia Ontology</u> Traffic
environment	<u>Schema.org Ontology</u> Place
speed	<u>Schema.org Ontology</u> speed

ENERGY DOMAIN

The dataset provided within the Energy domain includes information about clients and data from sensors on different PV systems. Each concept will be semantically annotated with one or more suitable RDF semantic ontologies in order to provide meaning and allow discoverability and semantic interoperability.

As aforementioned, the investigation of the Energy semantic data models is still work in progress and will be analyzed in detail in the next version of this deliverable, namely D4.2 and D4.3.

SUMMARY

All above semantic representations are stored in a GraphDB RDF store database to allow concept retrieval for TRUSTEE operations over the HE-encrypted datasets and AI models. Each concept available within a dataset has been described in RDF triplets using the schema subject-predicate-object to provide as much semantic information about a concept as possible for future searchability and interoperability. Such descriptions include:

- i. Dataset concept is a class / property
- ii. Dataset concept belongs to a Dataset with ID [...]
- iii. Dataset concept refers to a domain [...]
- iv. Dataset concept is involved in HE operation [...]
- v. Data concept has URI similar to RDF semantic ontology class [...]
- vi. Data concept has label / description [...]
- vii. Data concept has relationship with [list of other data concepts]

All of these triplets are stored within GraphDB and available for retrieval through a series of SPARQL queries (see Figure 169 and Figure 170).

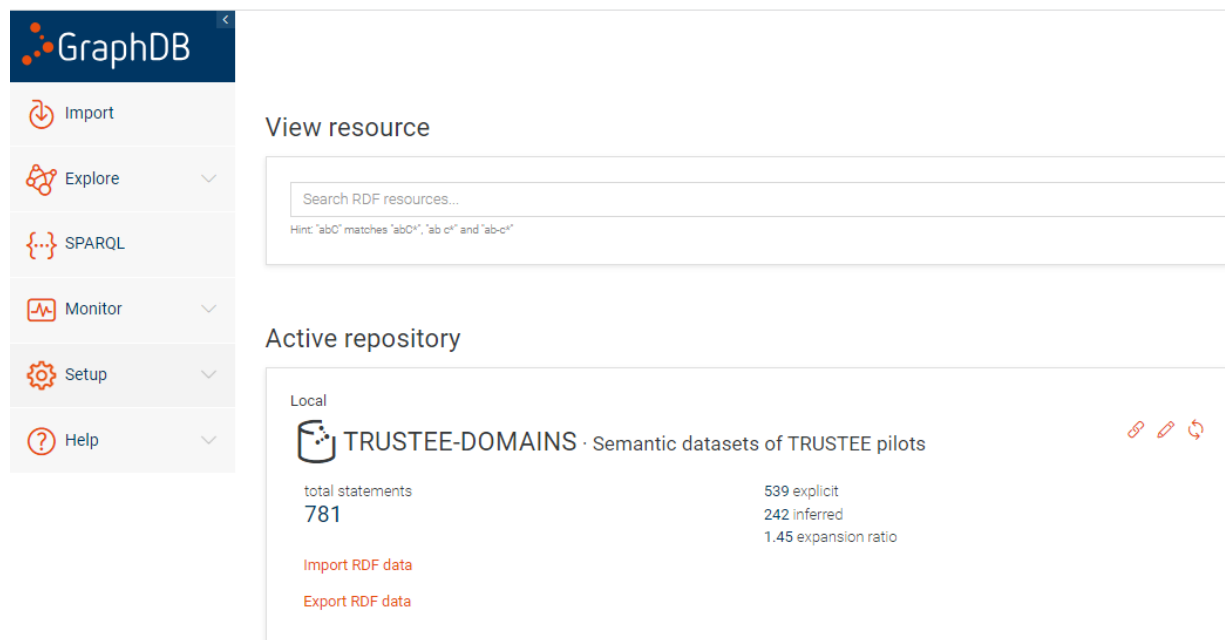
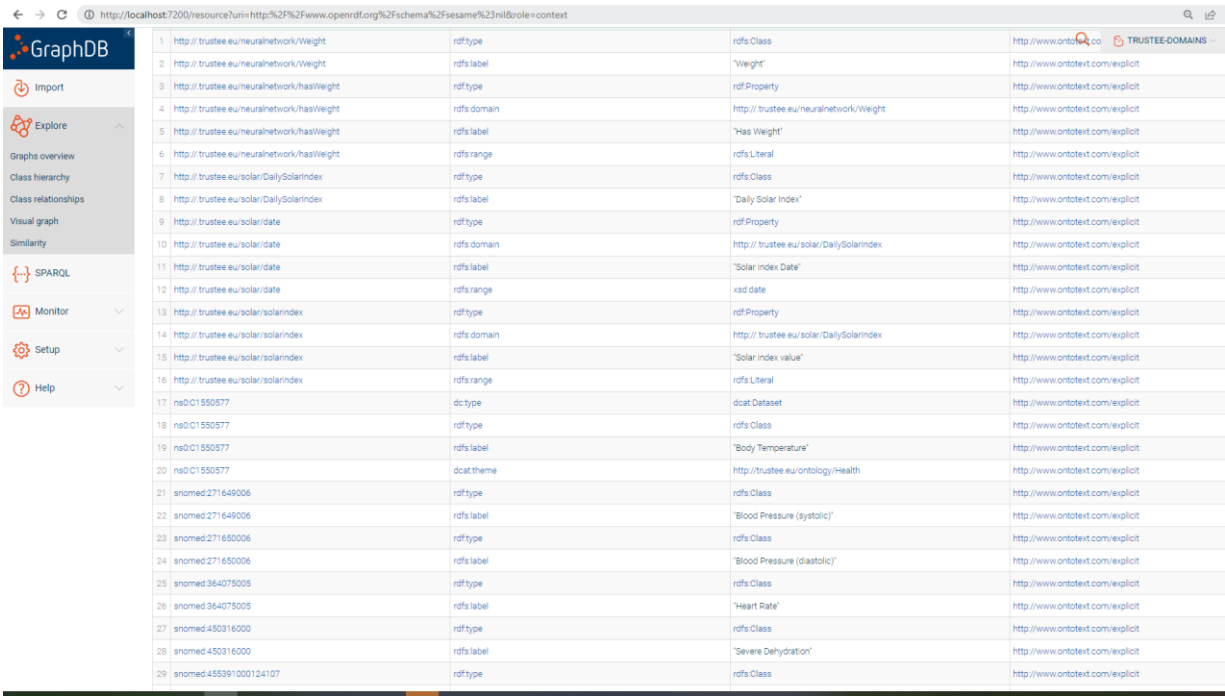


Figure 169: GraphDB Semantic data sets of TRUSTEE pilots

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework



The screenshot displays the GraphDB web interface. On the left is a sidebar with navigation options: Import, Explore (selected), SPARQL, Monitor, Setup, and Help. The main area shows a table of 29 RDF triples. Each row contains an index, an IRI, a property, a value, and a source URL. The properties include rdfs:type, rdfs:label, rdfs:domain, rdfs:range, rdfs:Class, rdfs:Literal, and dc:type. The values represent various concepts like 'Weight', 'Solar Index', 'Body Temperature', 'Blood Pressure (systolic)', and 'Severe Dehydration'.

	IRI	Property	Value	Source
1	http://trustee.eu/neuralnetwork/Weight	rdfs:type	rdfs:Class	http://www.ontology.co
2	http://trustee.eu/neuralnetwork/Weight	rdfs:label	"Weight"	http://www.ontology.co/explicit
3	http://trustee.eu/neuralnetwork/hasWeight	rdfs:type	rdfs:Property	http://www.ontology.co/explicit
4	http://trustee.eu/neuralnetwork/hasWeight	rdfs:domain	http://trustee.eu/neuralnetwork/Weight	http://www.ontology.co/explicit
5	http://trustee.eu/neuralnetwork/hasWeight	rdfs:label	"Has Weight"	http://www.ontology.co/explicit
6	http://trustee.eu/neuralnetwork/hasWeight	rdfs:range	rdfs:Literal	http://www.ontology.co/explicit
7	http://trustee.eu/solar/DailySolarIndex	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
8	http://trustee.eu/solar/DailySolarIndex	rdfs:label	"Daily Solar Index"	http://www.ontology.co/explicit
9	http://trustee.eu/solar/date	rdfs:type	rdfs:Property	http://www.ontology.co/explicit
10	http://trustee.eu/solar/date	rdfs:domain	http://trustee.eu/solar/DailySolarIndex	http://www.ontology.co/explicit
11	http://trustee.eu/solar/date	rdfs:label	"Solar index Date"	http://www.ontology.co/explicit
12	http://trustee.eu/solar/date	rdfs:range	xsd:date	http://www.ontology.co/explicit
13	http://trustee.eu/solar/solarindex	rdfs:type	rdfs:Property	http://www.ontology.co/explicit
14	http://trustee.eu/solar/solarindex	rdfs:domain	http://trustee.eu/solar/DailySolarIndex	http://www.ontology.co/explicit
15	http://trustee.eu/solar/solarindex	rdfs:label	"Solar index value"	http://www.ontology.co/explicit
16	http://trustee.eu/solar/solarindex	rdfs:range	rdfs:Literal	http://www.ontology.co/explicit
17	ns0:C1550577	dc:type	dc:Dataset	http://www.ontology.co/explicit
18	ns0:C1550577	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
19	ns0:C1550577	rdfs:label	"Body Temperature"	http://www.ontology.co/explicit
20	ns0:C1550577	dc:theme	http://trustee.eu/ontology/Health	http://www.ontology.co/explicit
21	snomed:271649006	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
22	snomed:271649006	rdfs:label	"Blood Pressure (systolic)"	http://www.ontology.co/explicit
23	snomed:271650006	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
24	snomed:271650006	rdfs:label	"Blood Pressure (diastolic)"	http://www.ontology.co/explicit
25	snomed:364075005	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
26	snomed:364075005	rdfs:label	"Heart Rate"	http://www.ontology.co/explicit
27	snomed:450316000	rdfs:type	rdfs:Class	http://www.ontology.co/explicit
28	snomed:450316000	rdfs:label	"Severe Dehydration"	http://www.ontology.co/explicit
29	snomed:455391000124107	rdfs:type	rdfs:Class	http://www.ontology.co/explicit

Figure 170: DL GraphDB RDF store

Figure 171 shows a visual graph of the semantic classes that have been used to semantically represent the pilot datasets concepts. As a total, 34 classes annotate the concepts from the TRUSTEE pilot domains. These classes are used to depict what a dataset concept is about, i.e. what do the raw data available for this concept (e.g. rows of an excel column) represent (e.g. a person, an event, a temperature, oxygen measurements, student grades, weights, a location etc.).

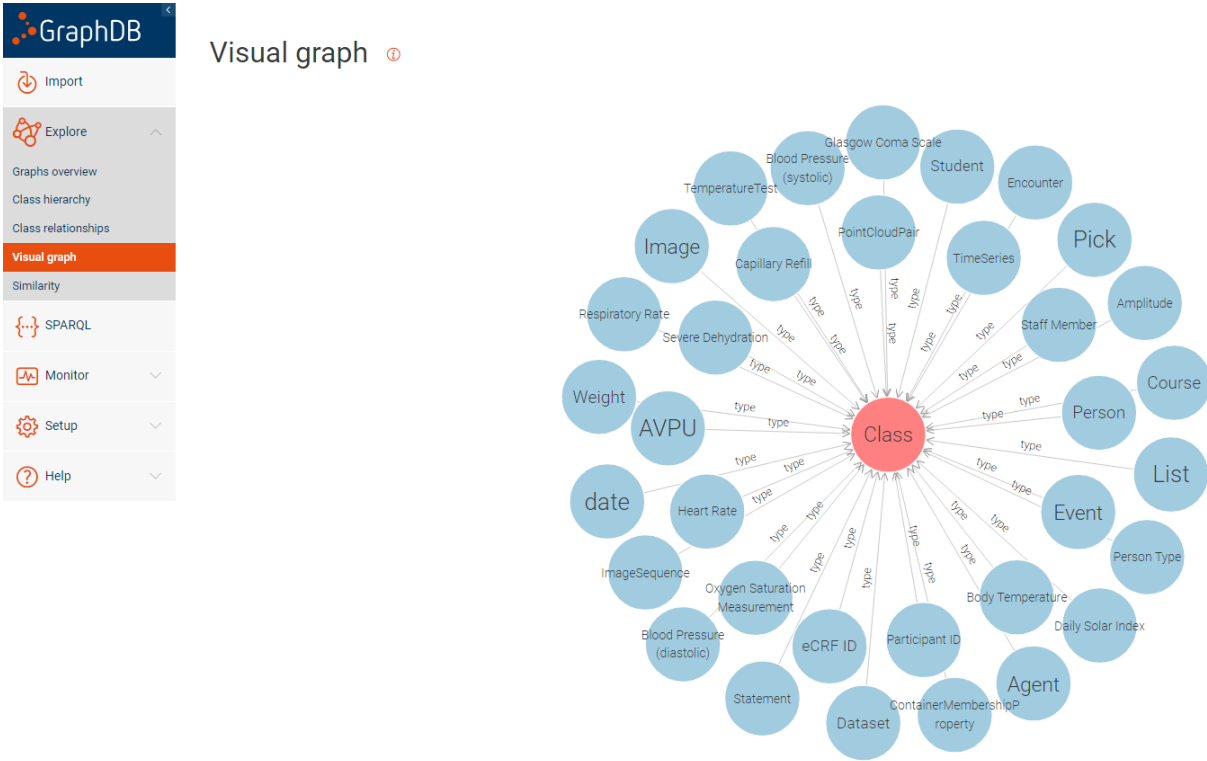


Figure 171: GraphDB visual graph of TRUSTEE semantic classes

Figure 172 shows the visual graph within GraphDB that represents all semantic properties that have been used to showcase the relationships identified for each concept. These relationships may include information to depict that a concept a) has a name, b) has a temperature, c) is a person, d) has a URI, e) has weather condition etc.

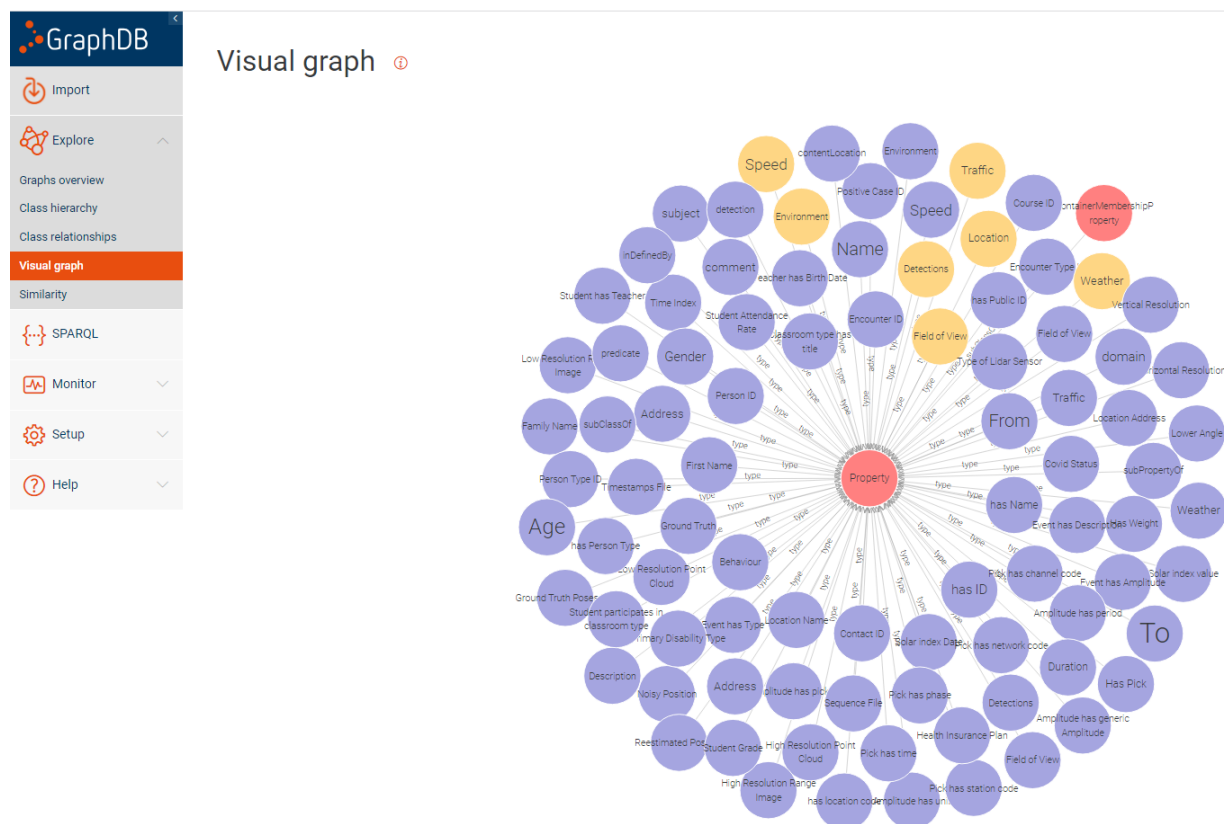


Figure 172: GraphDB visual graph of TRUSTEE semantic relationships / properties

Apart from the RDF store, a PostgreSQL database has also been set up to store the rest of the DL sub-components. For now, the database is storing HE public keys and will store the rest of the elements that will be available in the DL through the APIs' communications shown in Table 62.

Table 62: DL APIs communications

Function	API communication to	Input	Output
Store data indexes created within HEDF	HEDF	Data indexes information (e.g. id, provider, domain etc)	Status flag for storage (success, error)
Store and provide HE public keys	SSI-HE	New HE Public keys and HE context	HE Public Keys and HE context based on selection criteria
Store and provide attributes for searching	SSI-HE	Attributes (e.g. domain, date, location, HE operation etc.)	Attributes (e.g. domain, date, location, HE operation etc.)
Send records of transactions	ATR	-	Records of transactions
Store and provide data catalogs	External Connectors	New Data catalogs information (e.g. external	Data catalogs information (e.g. external connector,

	connector, dataset id, dataset description, size, etc)	dataset id, dataset description, size, etc) based on selection criteria
User authentication AM information	Authentication status	Authentication status
AI models indexes (for existing models), new AI models and logs for storage	New AI models information (id, index, dataset etc)	AI models indexes
Search for any previous logs for previous AI models runs	AI model and dataset indexes	Previous logs for existing run results

6.5.4. TECHNICAL DETAILS

Table 63 gathers some technical details of the first implementation of the DL.

Table 63: DL component first implementation technical details

DEVELOPMENT	
Type (Software/Hardware)	Software
Operating Systems	Linux
Database and databases tools	GraphDB RDF store for semantic datasets Postgresql database for storing dataset indexes, attributes catalogues, dataset catalogues.
GUI	No direct GUI access for user
Programming language	Python, JSON
Development Environment	Visual Studio Code, UNIX VI editor
INSTALLATION AND DEPLOYMENT	
Software Requirements	Python, Postgresql at the current stage
Hardware Requirements	Hosting to be provided at the next stage
Containerization	Not at the moment
Communications	HTTP / JSON

6.6. VALIDATION

The initial validation steps include storing all semantically annotated datasets into the Semantic repository (GraphDB). Each concept within a domain's dataset is characterized by a class from an existing RDF semantic ontology (when existing ontologies that can describe the concept exist) and is also annotated with additional features / relationships that provide more meaning to the concept (as shown in Figure 173).

Blood Pressure (diastolic)

Source: <http://purl.bioontology.org/ontology/SNOMEDCT/271650006>

subject	predicate	object	context	all	Explicit only
	subject	predicate	object		
1	snomed:271650006	rdfs:type	rdfs:Class		
2	snomed:271650006	rdfs:label	"Blood Pressure (diastolic)"		

Figure 173: Example of semantic dataset in GraphDB

The retrieval of information from the semantic storage has been carried out through SPARQL queries, ensuring a) retrieval of concepts from any domain based on specific search criteria and b) semantic interoperability across multiple domains.

For example, Figure 174 shows a use case where we want to retrieve the dataset concept that is related to the Health domain and is also involved in the HE operation called "Temperature Test". In Figure 174, we add these search criteria as default values; these will be dynamic values retrieved from an API communication between the DL and HEDF. The query result provides the concept that should be provided to the HEDF sub-component that will perform the HE operations and data fusion, so that the component knows which concept should be retrieved within the dataset.

SPARQL Query & Update 

Editor only

Editor and results

Results only



Unnamed 

Unnamed 



```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dc: <http://purl.org/dc/elements/1.1/>
4 PREFIX dcat: <http://www.w3.org/ns/dcat#>
5 PREFIX ont: <http://trustee.eu/ontology#>
6
7 SELECT ?resource ?label
8 WHERE {
9   ?resource dcat:theme <http://trustee.eu/ontology/Health> ;
10    rdfs:label ?label .
11   ont:relatestoHEOperation rdfs:domain ?resource ;
12    rdfs:range <http://trustee.eu.com/ontology#TemperatureTest> .
13 }
14
```

Dynamic value retrieved from API

Dynamic value retrieved from API

Run

Press Alt+Enter to keyboard shortcut

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, moments ago

	resource	label
1	ns0:C1550577	"Body Temperature"

Figure 174: SPARQL query of dataset concept in Health domain

Figure 175 shows another validation example, where the DL searches for any concepts that are related to an address. This type of querying also supports and achieves inter-disciplinary data fusion, since the results combine data from two domains (i.e. Health, Education).

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The screenshot shows the TRUSTEE SPARQL query interface. The query is as follows:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dc: <http://purl.org/dc/elements/1.1/>
4 PREFIX dcat: <http://www.w3.org/ns/dcat#>
5 PREFIX ont: <http://trustee.eu/ontology#>
6
7 SELECT ?address ?class ?domain
8 WHERE {
9     ?address rdfs:label "Address" .
10    ?address rdfs:domain ?class .
11    ?class      dcat:theme ?domain .
12 }
13
```

The results table shows three rows of data:

	address	class	domain
1	schema.address	ont:Encounter	http://trustee.eu/ontology/Health
2	schema.address	ont:Person	http://trustee.eu/ontology/Health
3	schema.address	http://vivoweb.org/ontology/core#Student	http://trustee.eu/ontology/Education

Figure 175: SPARQL query of dataset concepts in Education domain (inter-disciplinary)

The DL can also support semantic interoperability, as shown in Figure 176, where the query searches for all concepts that are involved in the HE operation "Find COVID students" from any domain.

The screenshot shows the TRUSTEE SPARQL query interface. The query is as follows:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dc: <http://purl.org/dc/elements/1.1/>
4 PREFIX dcat: <http://www.w3.org/ns/dcat#>
5 PREFIX ont: <http://trustee.eu/ontology#>
6
7 SELECT ?class ?label ?domain
8 WHERE {
9     ?class dcat:operation ont:FindCOVIDStudents ;
10     rdfs:label ?label ;
11     dcat:theme ?domain .
12 }
13
```

The results table shows four rows of data:

	class	label	domain
1	ont:encounterId	"Encounter ID"	http://trustee.eu/ontology/Health
2	ont:encounterTypeId	"Encounter Type ID"	http://trustee.eu/ontology/Health
3	ont:positiveCaseId	"Positive Case ID"	http://trustee.eu/ontology/Health
4	http://purl.obolibrary.org/obo/laboratoryfinding	"Covid Status"	http://trustee.eu/ontology/Education

Figure 176: SPARQL query of dataset concepts (semantic interoperability)

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The next step of the validation process is the design of the data models that will be available within the DL, i.e. the communication channels between the DL and other ABBs within TRUSTEE. Since the DL is the main storage unit of the project's ecosystem, the main functionalities revolve around retrieving and providing different types of data across the platform's components. Figure 177 shows the data model and API design process for all communications between the DL and other TRUSTEE components.

```
1 from flask import Flask, Blueprint, request, jsonify
2 from data_lake.DB.PGConnector import PGConnector
3 from data_lake.DB.Utils import *
4 from data_lake.HEPublicKeys.HEPublicKeyRow import HEPublicKeyRow
5 from flask import current_app
6 import shutil
7 import requests
8 import uuid
9
10 hepublickey_server = Blueprint('hepublickey_server', __name__, template_folder='data_lake.Server')
11
12 # HE Public Key
13 @hepublickey_server.route('/hePublicKey', methods=['GET'])
14 def getPublicKey():
15     params = request.args
16     id = params.get('id')
17     db = connectToPG()
18     publicKey = db.selectPublicKeyById(id)
19     if publicKey is None:
20         return {}, 404
21     return jsonify({
22         "id": publicKey.data.get("id"),
23         "url": publicKey.data.get("url"),
24         "size": publicKey.data.get("size")
25     }), 200
26
27 @hepublickey_server.route('/hePublicKey', methods=['POST'])
28 def postPublicKey():
29     jsonBody = request.get_json(force=True)
30     if "url" in jsonBody:
31         url = jsonBody["url"]
```

Figure 177: DL Data Models (APIs) design

Finally, exemplary HE public keys that have been created and provided by HEDF ABB, are stored (Figure 178) in the PostgreSQL database that has been set up within the DL and retrieved from the database for posting to other ABBs (Figure 179).

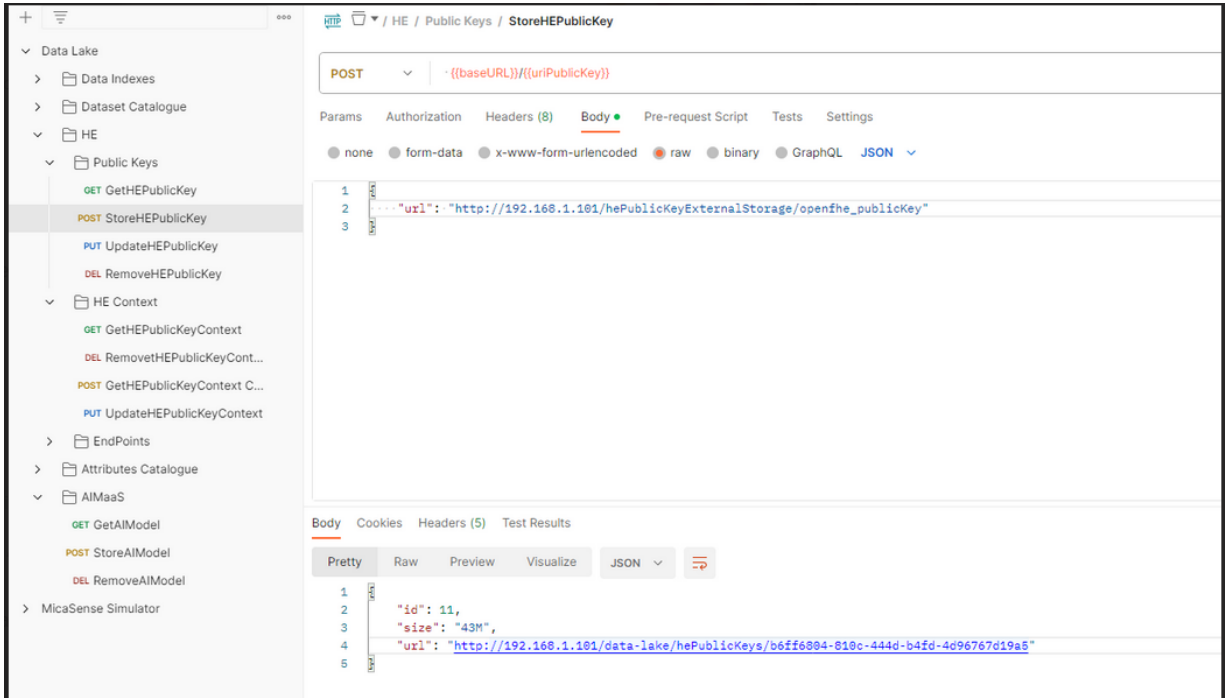


Figure 178: Storing HE Public Keys in PostgreSQL

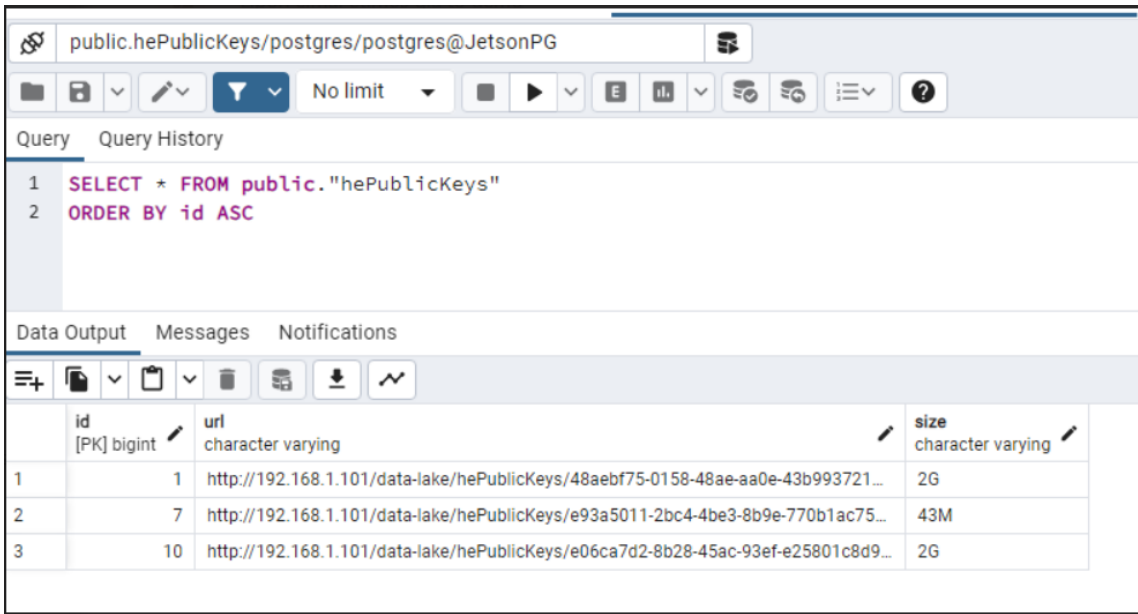


Figure 179: HE Public Keys retrieval within DL

6.7. NEXT STEPS

The initial phase of the work done for the design and implementation of the DL focused on the background research of the different elements involved (semantics, data indexing, HE context, public keys, metadata, AI models etc), and on the semantic annotation of all domains' datasets so that the next steps can be carried out.

The next versions of the DL, to be gathered in D4.2 and D4.3, will consist of:

- Adding more semantic concepts in the GraphDB (addition of relationships that map concepts with domain HE operation and column index)
- Extending the database capabilities according to needs (e.g. Dataset catalogue, Attributes catalogue etc.).
- Completing data models design and API communications with other ABBs (AIMaaS, KR, Distributed Data Sources, ATR etc.)
- Achieving dynamic querying of data stored in the DL via APIs.

6.8. CONCLUSIONS

This chapter has presented theoretical and technical information regarding the DL ABB of the TRUSTEE ecosystem. The DL aims to provide guidelines that ensure common understanding of the data and services offered by each TRUSTEE component, laying the path for ensuring semantic interoperability and flexible data management, adhering to common standards such as the EOSC Semantic interoperability v1.0 framework and the FAIR principles.

The DL is currently available for validation with internal testing. As this is a component that is heavily dependent on its communication with the other ABBs, the majority of the work to be done in the future stages requires that the other ABBs are also deployed and ready for data exchange through established APIs.

The next phase of the implementation will include the design of all data models that will ensure communications with the rest of the ABBs, the development of the sub-components of the DL and their population with entries (e.g. attributes catalogues, data indexes storage, semantic concepts storage, AI models storage etc

7. PART V: SECURITY AND TRUST MANAGER

7.1. INTRODUCTION

7.1.1. NEED AND OBJECTIVES

The STM is responsible for managing the request and collection of agreements and checking the continued validity of an agreement to permit its use via the agreement management tool. An agreement needs to be requested again if it is: withdrawn, expired, invalidated, revoked or if it needs to be modified, confirmed, or reaffirmed. Technically, the agreement management in TRUSTEE is a method, procedure, or collection of policies to enable data providers to specify what data they are willing to allow access to various data consumers. The main steps in the agreement management involve collection of agreements, storage of agreements, use of collected agreements and data in a fully transparent and GDPR-compliant way. Hence the main goal of the STM can be summarised as (i) build a tool that will enable the Data Providers to have total control over their data by specifying, monitoring and enforcing constraints on their data; (ii) enable Data Consumers to conform to GDPR regulation [32] and (iii) finally, foster trust and transparency amongst Stakeholders in data sharing.

To achieve this, the STM will seek the following technical achievements: (i) have a machine-readable data format of privacy constraints imposed to data sets; (ii) manage the life-cycle of agreements for data and AI model requests; (iii) ensure the security and maintain the authenticity, validity, and integrity of the agreements made among the stakeholders of TRUSTEE; and finally (iv) deliver a user-friendly interface.

To achieve the goal of building a tool that will enable TRUSTEE stakeholders to have total control over their data via a STM, including an agreement management platform, the STM will seek technical achievements such as:

- Building a tool that will enable the TRUSTEE stakeholder to have total control over their data via a STM manager including an agreement management platform.
- Being GDPR compliant.
- Providing total transparency to TRUSTEE stakeholders.

7.1.2. INNOVATION

The novelty of the STM is to facilitate the data sharing between the data provider and data consumer. It enables the management of GDPR-based agreement as well as the lifecycles of agreements created by the data provider and accepted/signed by the data consumer.

Unlike existing consent management, the STM accounts for the Data Provider's constraints which is then aggregated with GDPR based constraints to serve as baseline for Agreement monitoring and enforcement. Besides, both data Providers and Consumers can manage their signed agreements, amend or revoke based on the evolving context information such as the provider's compliance to GDPR.

STM will use digital signature and trusted timestamping mechanisms for signing and managing the agreement life-cycles. Trusted timestamping is a process of securely recording the time and date when a particular digital asset. The digital timestamp is created using cryptographic techniques that ensure the integrity and authenticity of the timestamp, making it tamper-proof and verifiable. Trusted timestamping is commonly used in legal and regulatory contexts where the timing of events is critical.

7.2. BACKGROUND

7.2.1. ONTOLOGY

Ontologies are a flexible way of modelling complex data models. An ontology provides a lenient schema about any modelled object from the real world, this means that complex topics or concepts, such as the nature of law and legal rights, can be turned into a human and machine-readable format using ontologies. The goal of ontologies is to represent the real world and create a shared common understanding of a subject between experts of different domains. Studer et al. define an ontology as “*a formal, explicit specification of a shared conceptualisation*” [69]. An ontology mainly consists of nodes and edges. A node represents an object, whereas an edge describes the relationship between a node and an edge. Additionally, there have been several other works that focus on modelling privacy and GDPR-related ontologies to semantically enrich data [70] [71] [72] [73]. The ontology is used for modelling the data agreement in the STM.

7.2.2. GRAPH DATABASE

A graph-oriented Not Only Structured Query Language (NoSQL) database which stores nodes and relationships instead of tables, or documents. The graph database is used from the ontology as a backend to instantiate it and communicate with the frontend.

7.2.3. GDPR PRINCIPLES

The personal data processing in any sector must follow the principles of lawful processing stated in Article 5 of the GDPR. GDPR [32] defines 6-core principles as rational guidelines for service providers to manage personal data, including: lawfulness, fairness and transparency, purpose limitation, data minimization, data accuracy, storage limitation, and integrity and confidentiality. Moreover, the personal data processing must comply with one the legal laws set out in Article 6 and 9 GDPR, which is concerning the lawful processing of sensitive data. The data processing is based either on the agreement or a contractual relationship which is important for compliance with a legal requirement.

- i. Lawfulness, fairness and transparency. The lawfulness, fairness and transparency principle ensure six legal bases required by the GDPR namely: 1. Consent, 2. Contract 3. Legal Obligation 4. Vital Interest 5. Public Task, and 6. Legitimate Interest (defined in Article 6 of the GDPR in detail). These lawful bases might need to come along with other separate conditions for lawfully processing some special category data including education data, automotive data, healthcare data, racial and ethnic origin. Specific purposes and context of the data manipulation and processing should be determined and documented before starting to process personal data. The fairness and transparency requirements, as AI/ML algorithms like deep learning are normally operated in a black-box fashion, it is limited of transparency of how certain decisions are made, as well as limited understanding of the bias in data samples and training model [74] [75] [76].
- ii. Purpose limitation. The purpose limitation principle can be interpreted that a data provider needs to clearly inform users about the purposes. The purposes concern the use of personal data analysis and machine learning model training or/and deployed.
- iii. Data minimization. The data minimisation principle in the GDPR necessitates a data consumer (e.g., research organism) to collect and process personal data that is adequate, limited, and relevant only to claimed purposes.
- iv. Data Accuracy. The data accuracy principle is to ensure that a data consumer should keep personal data updated, correct and not misleading any matter of fact.

- v. Storage limitation. The storage limitation principle guarantees that a data consumer won't keep personal data longer than is necessary for the claimed purposes. In this case, data should be erased, anonymised, or encrypted.
- vi. Integrity and confidentiality (security). The integrity and confidentiality principle constrain data consumers to implement appropriate measures to effectively protect personal data.

7.2.4. TECHNOLOGY ENABLERS

7.2.4.1. PROTÉGÉ

Protégé [77] is a free and open-source tool to create ontologies. The Protégé tool was created at Stanford University. Protégé is a well-known tool, within the semantic technology community, for creating and maintaining ontologies efficiently. Protégé is used for creating, editing, visualizing, and managing the data agreement in STM.

7.2.4.2. FLUTTER

Flutter [78] is a cross-platform application development framework created by Google. Flutter is relatively new compared to other development frameworks, such as React or Angular, with a growing development community. Flutter uses the Dart programming language and leverages native compiling for the target platform to improve performance. Furthermore, Flutter enables an improved developer experience with its just-in-time (JIT) compiler, enabling a hot-reload feature to improve the development time of new features. Currently, Flutter supports native compiling for IOS, Android, Web, Windows, and Linux. FLUTTER is used for the development of the STM user interface.

7.2.4.3. JAVA SPRING FRAMEWORK

Java Spring framework is a Java framework dedicated to the creation and management of Web applications. It is useful for the design and development of the Representational State Transfer (REST) APIs. Spring boot is an extension to the Java Spring framework, designed to facilitate the creation and deployment of Spring applications. It allows developers to develop an independent Spring application with a minimum of configuration. Java Spring framework is used for the development of the backend of the STM.

7.3. REQUIREMENTS

In this chapter, the compliance of the STM ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding with the aforementioned requirements addressed during the development and by the implementation of the STM and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements have been considered. As the implementation of the STM progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

User-centred requirements were identified through the UX Research that was conducted under T2.3. These requirements were reported in D2.1 [1], while they were also utilized in order to extract technical, functional and non-functional requirements, that were also outlined in D2.1. Table 78 below presents the interplay between user-centred

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

requirements and technical requirements, that is the functional and non-functional requirements that address each user-centred requirement. The user-centred requirement ID is used to reference the respective requirement reported in D2.1 [1] while the requirement's description is also provided alongside whether the STM is currently fully or partially compliant with the user-centred requirement, i.e., indicating whether technical requirements have been identified in order to address the respective user-centred requirement. The ID of the technical requirement, as reported in D2.1 [1], that is addressing each user-centred requirement is also provided in the last column of the table. Additionally, it is possible than more than one technical requirement addresses a single user-centred requirement.

Table 64: STM - User-centred requirements

Req id (WP2)	Req. Description	Compliance	How it is addressed
USR-001	TRUSTEE must enable collaboration among domains and enable access and use of data for secondary use from multiple domains	Yes	Req-STM-FUNC-1 Req-STM-FUNC-9 Req-STM-NFUNC-1
USR-009	TRUSTEE must protect education data from misuse that aims in profiling students.	Partially	Req-STM-FUNC-1 Req-STM-FUNC-9 Req-STM-NFUNC-1
USR-029	TRUSTEE should offer collaboration across multi-disciplinary data domains	Yes	Req-STM-FUNC-1 Req-STM-FUNC-9 Req-STM-NFUNC-1
USR-033	TRUSTEE should enable computations and analysis over the accessed data.	Partially	Req-STM-FUNC-3 Req-STM-FUNC-6 Req-STM-FUNC-8 Req-STM-FUNC-9 Req-STM-NFUNC-1

Table 65 presents the preliminary results of the investigation regarding the compliance of the STM with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether STM complies with the requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 65: STM - Functional and non-functional requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
Req-STM-FUNC-1	It offers collaboration across multi-disciplinary data domains for multiple domains: Energy, Health, Education, Automotive, Space, Multi-Disciplinary.	Yes	User-friendly GUI for facilitating collaborations across multi-disciplinary domains.
Req-STM-FUNC-3	It plays a role in the efficiency of data querying.	Partially	Inform the user about the use of its data
Req-STM-FUNC-6	Create a data agreement	Yes	Create a data agreement
Req-STM-FUNC-8	Monitoring the agreements	Partially	Develop an agreement monitoring

Req-STM-FUNC-9	The agreements are created based on the GDPR requirements.	Yes	Check each step of the development if it's GDPR compliance.
Req-STM-NFUNC-1	The meta-data is used with respect to privacy regulations, legal frameworks and ethical considerations.	Partially	User-friendly GUI for metadata
Req-STM-NFUNC-2	It facilitates to search and request a data in a user-friendly way.	Yes	User-friendly GUI for facilitating the data request

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

7.4. ARCHITECTURE

In this section, both high level and low level of the STM architecture are presented. The high level shows the main connections with the rest of TRUSTEE external interfaces. The low level shows the internal communications and STM life-cycles.

7.4.1. HIGH LEVEL

The STM consists of three main components. The first component is the user interface, for a user to interact with the STM. The user interface is a web application developed using the Flutter cross-platform application development framework supported by the Dart programming language.

The second component of the STM is the server. The server is developed with the Java programming language and uses the Spring Boot framework to create a RESTful API to handle GET, PUT, POST, and DELETE requests. The Flutter web application uses the provided RESTful API to perform actions and store data into the STM's third component, the STM GraphDB. The results from each API request to the Java Server are returned in JSON format.

The STM GraphDB is used to store users, organisations, agreements, signatures, purposes, actions, and metadata of datasets. GraphDB is a NoSQL database using semantic ontologies to support and add meaning to the stored data. Additionally, the connection between the Java Server and the STM GraphDB is established through the integrated GraphDB RESTful API, through which the Java Server can perform SPARQL queries to manipulate the stored data. The result of the SPARQL queries is then sent back to the Java Server in JSON format. Figure 180 provides an illustration of the overall STM architecture.

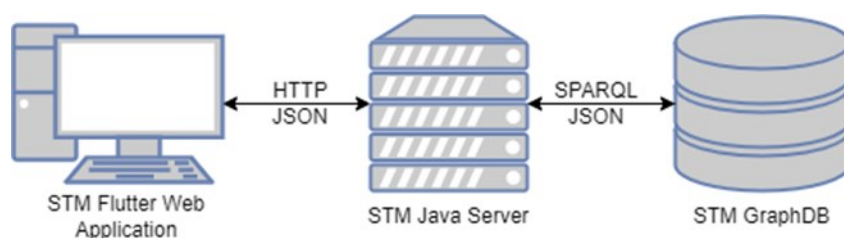


Figure 180: STM Manager's overall architecture

In a functional way, Figure 181 shows the whole architecture of the STM where the two main components of the STM are shown. These are the Agreement Manager, and the Agreement Monitoring. Additionally, Figure 181 shows the interactions with other TRUSTEE ABBs as defined in D2.1 [1].

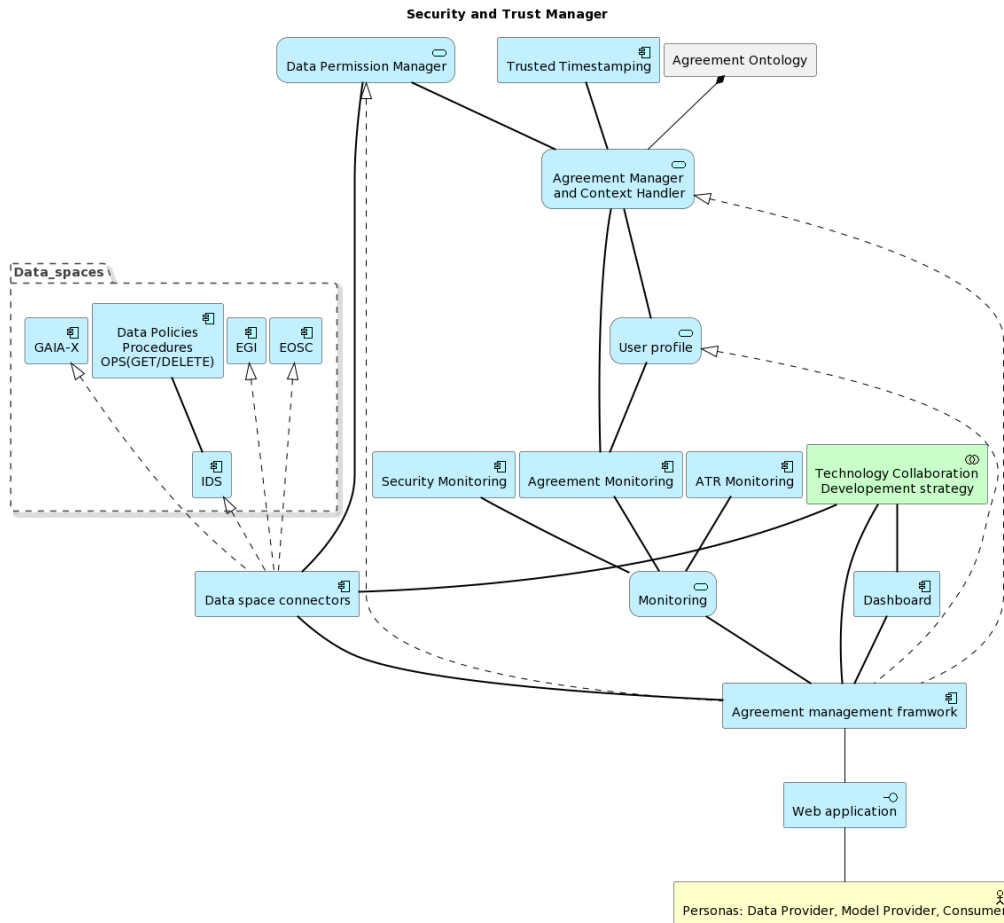


Figure 181: STM architecture

7.4.2. LOW LEVEL

On the one hand, Figure 182 illustrates a detailed architecture of the STM. The Provider state management architecture is used for the Flutter web application development. The Provider architecture is able to manage states within the web application to easily update and propagate data objects through each screen in the web application. A provider is created for each data model and is then initialized at the top-level of the web application, so that all components can read and write the necessary data models. The *Router* component of the web application is used to navigate the users throughout the application from one screen to another. The *Widget* component contains all custom developed widgets for the STM's web application. Furthermore, the *Data* component has all the relevant components to manage data models and communicate with the STM's Java Server REST API.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

The STM Java Server consists of three core components. The *Controllers* component is responsible for communicating with the Server's REST API. Following, the *Service* component translates the information from the controllers to prepare the data to be stored on (or manipulate) the STM's GraphDB. Lastly, the *Repository* component communicates with the STM's GraphDB REST API. The GraphDB provides a REST API through which SPARQL queries can be sent.

In order to provide meaning to data and enable machine-readability the STM's GraphDB is supported by the agreement ontology. The agreement ontology is used for defining the structure of each data model.

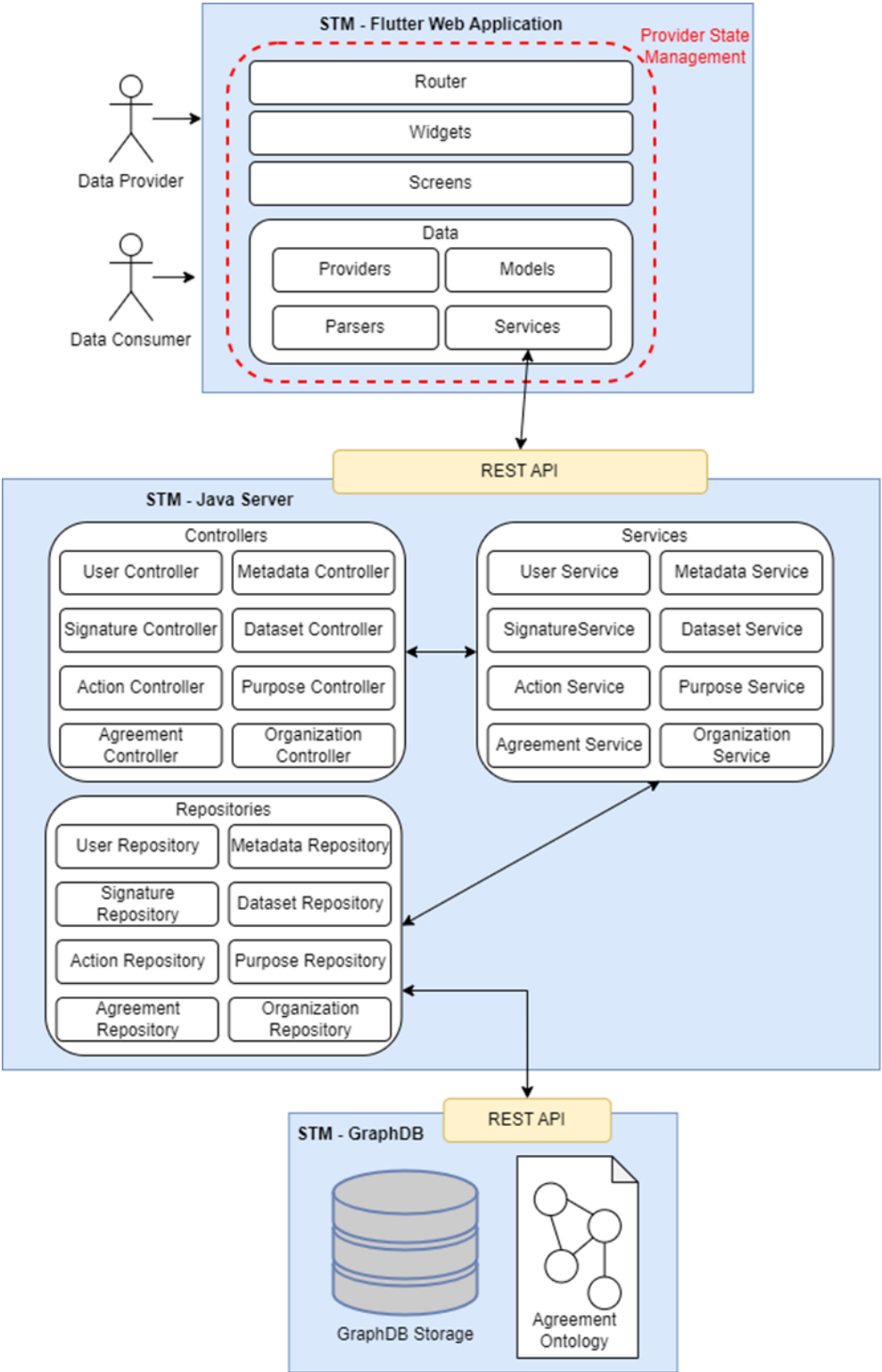


Figure 182: A detailed architecture of the STM.

On the other hand, Figure 183 shows agreement management life-cycles based on the consent life-cycles proposed in [72]. Based on the consent life-cycle proposed in [72] where four stages are represented: request, comprehension, decision and use. The proposed stages will be used in the agreement which inherits from the consent life-cycle.

- **Request of agreement:** it determines whether data sharing and processing can begin. A successful agreement request should be GDPR compliant. A semantic model for agreement, built on top of an ontology, would be beneficial to any component. Agreement requests are made to the data consumer and need to be visualized. Once the agreement is validated for acceptance, the agreement needs to be managed, for example, when stored for future compliance checking.
- **Comprehension of agreement:** the data providers and consumers have different needs and understanding of information a simple and effective visualization of consent is expected. In fact, a friendly visualization of the required data as well as the description of the agreement would be more efficient in comparison to a long text with privacy policies written in legalese.
- **Decision about agreement:** it allows to help and guide the data provider to fill-in the agreement and data consumers to take a decision to accept/decline a new agreement related to a dataset.
- **Use of agreement: an agreement can be used for the following purposes:** compliance checking, reasoning, and a proof of contract. Each purpose requires a dedicated tool. For instance, a compliance checker will be developed to check whether a data consumer is allowed to use the requested data, if yes, then, the accepting process can be enabled. Reasoning process can be performed in the ontology for inferring new relations. Finally, then storage of agreements in the ATR allows to provide a trust and proof of contract.

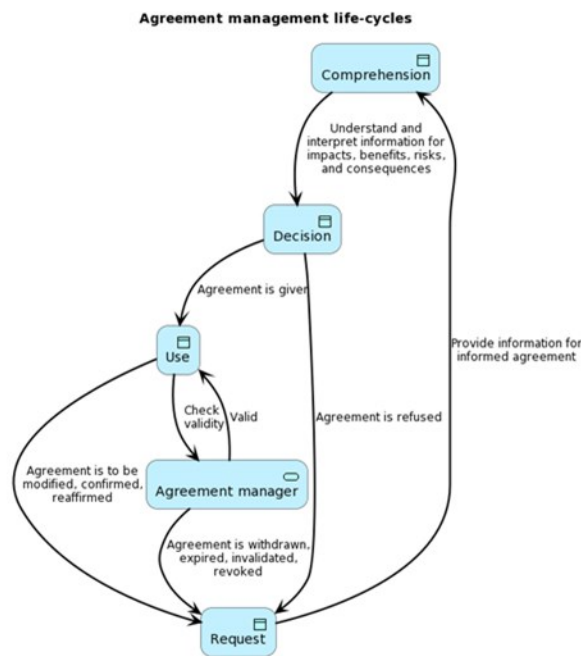


Figure 183: Agreement management life-cycles

7.5. IMPLEMENTATION

7.5.1. FUNCTIONAL DESCRIPTION

As explained in the Architecture section, the two main functional components of the STM ABB are the Agreement Manager and the Agreement Monitoring. However, there are also other related components that need to be mentioned: trusted timestamping module, context handler and security/audit monitoring. Furthermore, some new technical requirements for the different functional components have been identified.

7.5.1.1. AGREEMENT MANAGER

The Agreement Manager inside STM is a fundamental component of the TRUSTEE Platform for all data and/or AI model sharing. The agreement object describes the agreement requested from a consumer to access specific datasets or AI models based on GDPR compliance.

The Agreement Manager takes into consideration the creation and generation of an agreement by the data and/or AI model provider. An agreement is requested and signed by a data and/or AI model consumer. The Agreement Manager has the following tasks:

- It enforces that the data and/or AI model is processed according to the (pre)defined agreement terms and privacy status.
- It maintains multiple versions of the agreement for audit and tracking agreement actions, purposes, updates, and status (initial, updated, or withdrawal). All the versions are recorded in an ATR component (described later). Only the latest (current) version of the agreement is applied and shown.
- It maintains who created the agreement, for whom was the agreement created, and its validation.
- It coordinates with the rest of modules such as data manager, and context handler for both agreement and data life-cycles.
- It maintains a unique hash associated to an agreement with a trusted timestamping related to a specific data (AI model).
- It is responsible for maintaining the agreement proof.

All these characteristics of the Agreement Manager are performed by the following modules.

AGREEMENT ONTOLOGY BASED ON GDPR CONCEPTS

The previously mentioned Agreement Manager is supported by the newly developed agreement ontology, named *Ontogreement*. The main reason for providing a new ontology is to extend and improve the existing GDPR-related ontologies. Existing ontologies model B2B and B2C-related situations, however, there is still a need for an ontology that models data-sharing agreements, for any type of user, that concern specifically datasets and/or AI models. Further, the Ontogreement ontology will support the Agreement Manager in deciding if two or more agreements can be merged to enable data fusion.

During the development of the agreement ontology, we used the top-down approach [79] whilst leveraging the GDPR's text corpus to ensure we are aligned with the legal concepts described in the GDPR. Additionally, to support the FAIR principles [80], other GDPR-related ontologies from published scientific papers and journals were utilized to avoid reinventing the wheel on certain concepts in the ontology. The agreement ontology currently includes the following existing ontologies:

- Web Ontology Language [81]
- PROV Ontology [82]
- smashHitCore Ontology [83]
- Data Privacy Vocabulary [84]

- DCAT 2 Vocabulary [85]
- Schema.org Ontology [86]

Several parts of each mentioned ontology are combined and extended by the STM's Ontogreement. Below are the following concepts that, to the best of our knowledge, have not yet been modelled, until now by the new Ontogreement ontology:

- Model
- GDPR Agreement
- Process
- Personal Data Category
- Supervisory Authority

The current hierarchy of modelled concepts is illustrated in Figure 184. Furthermore, a depiction of the concepts and their relationships to another is shown in Figure 185. As ontologies consist of properties that describe the relationship between concepts, the Ontogreement ontology adds the following properties:

- allowed Process
- has Role
- has Purpose
- signed
- signed By
- used For

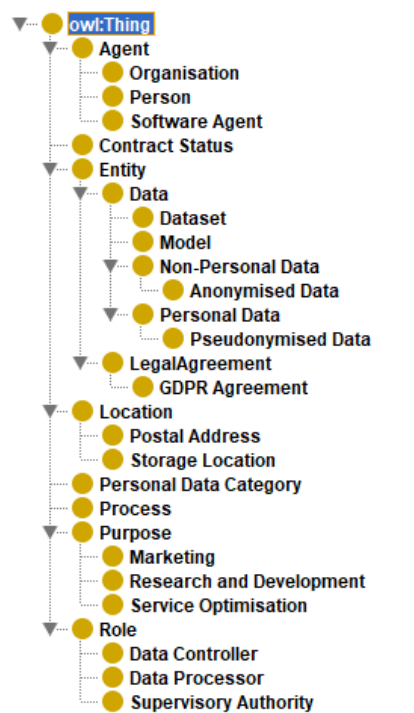


Figure 184: Agreement ontology class hierarchy

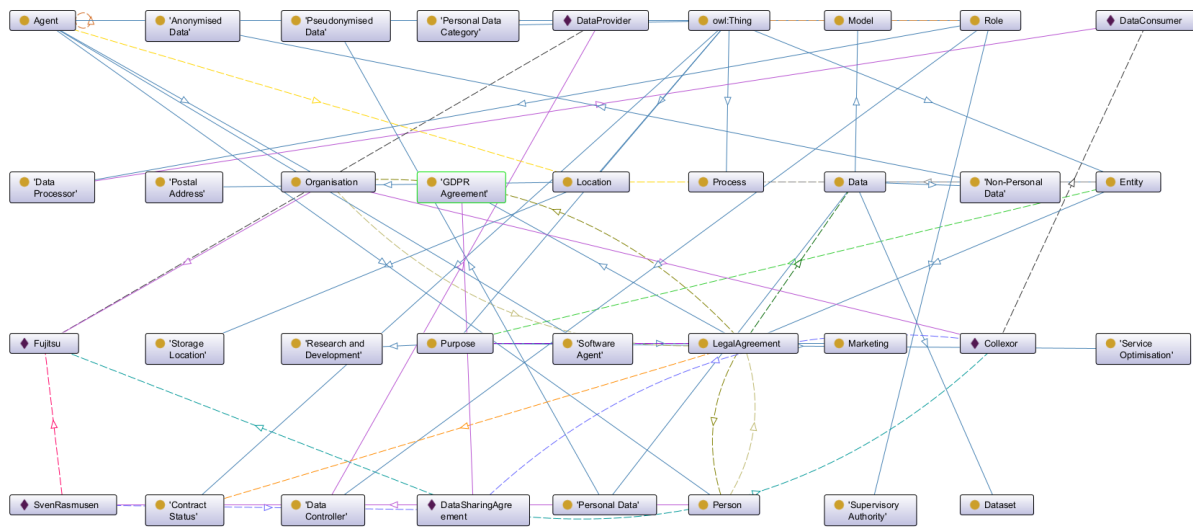


Figure 185: Agreement ontology's concepts and their relation to another.

AGREEMENT OPERATIONS

Two sets of agreement operations are allowed for the data provider and data consumer.

- Data provider:
 - **Creation:** the data provider allows to create and fill-in the information related to the agreement based on the GDPR requirements.
 - **Update:** an existing agreement created by a data provider can be updated at any time. However, the update should be done before any accepting or signing of the agreement by a data consumer. Whether the data consumers have already signed the agreement, the data provider updates the agreement and STM notifies the data consumers about the new update to accept it or decline it.
 - **Remove:** the data provider may remove an agreement in the case where the agreement is not signed by a data consumer.
- Data consumer
 - **Accept/Sign:** the data consumer accepts and signs an agreement related to the requested dataset.

AGREEMENT VALIDITY CHECKER

Agreement validity time and life-cycle checker is a key part of permission of an agreement where each permission is guaranteed for a certain time period. The life-cycle checker is responsible for end-to-end data and agreement life-cycle management in different steps of agreements. It also checks the validity or the update of the agreement. The life-cycle of an agreement is:

- Step 1: Creation: an agreement is created by the data provider.
- Step 2: Storage: the collected agreement is securely stored.
- Step 3: Process: the stored agreement is processed based on the agreement obtained between a data provider and a data consumer.
- Step 4: Modification: an agreement may be modified at any time by the data provider.
- Step 5: Revocation: an agreement may be revoked by the data provider due to agreement breach or expiration validity.

- Step 6: Archive: an agreement may be archived for regulatory, monitoring, and audit needs.
- Step 7: Destruction: an agreement may be destroyed according to the legislative requirements.

7.5.1.2. AGREEMENT MONITORING

Agreement Monitoring component provides a centralized view to TRUSTEE stakeholders of relevant Key Performance Indicators (KPIs) of the agreements in real time. It is a web-based tool to highlight the main and useful information. The agreement monitoring component further serves to enable a maintainable overview for all relevant users of the STM to stay informed about the agreements the user participates in. Additionally, through the Ontogreement ontology and the GraphDB, the overview of agreements is further supported.

7.5.1.3. TRUSTED TIMESTAMPING MODULE

Trusted timestamping module is responsible for contacting the trusted timestamping authority service for generating a timestamp cryptographic proof for embedding it into an agreement. Probable and trusted timestamping for security reasons is important as it enables to avoid malicious assertions that an agreement was generated at a particular point-in-time. Trusted Timestamping also helps to track the agreement life-cycles: agreement creation, modification, or cancellation. There are several trusted timestamps we can use such as the National Institute of Standards and Technology (NIST) randomness beacon-based timestamp proof or Digital Signature Services (DSS) which aims to follow the electronic IDentification, Authentication and trust Services (eIDAS) Regulation and related standards to prove the exact time. The trusted timestamps will be used for accepting (signing) and managing the timestamps of agreement life-cycles.

7.5.1.4. CONTEXT HANDLER

Every action performed by a user on the platform has an attached context. The context handler is a reactive service responsible for interpreting the context and triggering a relevant action.

When a data consumer requests data from a data provider and accepts the data provider agreement, then a notification is recorded and sent to the data provider as well as the DA. Then, a context handler provides the following functionalities.

- **Agreement request:** is responsible to get an agreement and send it.
- **Data access control:** controls and regulates data consumers and data access in the platform. It enables restricting access based on a set of policies specified by a data provider. It manages data consumers' access rights based on their agreements and limits by defining what data they have access to.
- **Policies and rules trigger:** A policy is a set of rules that are running in a single checkpoint. It is designed to evaluate and handle a context or potentially risky contexts that are created or requested in different operations of agreement management. This module maintains and checks the validity of a context and its associated rules for correctness or conflict.

7.5.1.5. SECURITY/AUDIT MONITORING

The idea behind this component is to provide monitoring capabilities over the Blockchain network used in the ATR for the STM.

A monitoring system is a set of tools designed to observe, collect, and analyze data or metrics from various sources within a system or network. It provides real-time visibility into the performance, health, and behavior of components, applications, or infrastructure. By monitoring these metrics, organizations can proactively identify issues, detect anomalies, and ensure optimal performance and reliability. Monitoring systems often include features like alerting, reporting, and visualization to help administrators and operators effectively manage and troubleshoot their systems.

Architecture

The Blockchain network considered in Trustee is a Hyperledger Fabric (HLF) network. Therefore, the security and audit monitoring component must have the capability to read from an HLF network (through Blockchain events) and provide information about its current state. Therefore, it must include specific tools to interact with a HLF network.

Consequently, the security and audit monitoring component have the following sub-components, namely Blockchain listener, Blockchain Orchestrator, and Event broker, which are further described below.

Blockchain listener

Sub-component that listens for events from the Blockchain network. This component is responsible for connecting to the Blockchain network and monitoring events. It interacts with the Blockchain's API or uses an Software Development Kit (SDK) Fabric to listen for new events or changes in the Blockchain state.

It receives events as they occur and passes them to the orchestrator. It must be able to connect with an HLF network and listen for events produced by the smart contracts of this network. Furthermore, it must be able to query the network and get relevant information requested by the users. It plays a crucial role in facilitating communication between various network participants, such as peers, orderers, and clients. It acts as an interface between the external world and the Fabric network, allowing external entities to interact with the Blockchain network by sending requests or receiving notifications about network events.

The listener establishes and maintains network connections to receive incoming requests from clients, peers, and orderers. It listens on specific network ports or channels for incoming messages. Once a request is received, the listener processes the request by validating it, verifying the identity of the sender, and performing necessary checks to ensure the request adheres to the network's rules and policies. In addition to handling requests, a listener can also broadcast network events to interested parties. These events could include block creation, transaction endorsements, chaincode invocations, or other significant updates in the network. By notifying relevant participants, the listener enables real-time updates and facilitates timely responses.

Blockchain Orchestrator

The Blockchain Orchestrator component acts as the central coordinator of the event flow between the Blockchain and the broker. It receives events from the Blockchain and handles their processing and storage in the broker. The orchestrator is responsible for managing the event subscription, event processing, and interaction with the broker component.

An orchestrator that receives events from a Blockchain and stores them in a broker consists of several components working together to ensure seamless event processing and storage.

The main components are the following:

- **Event Subscription:** The orchestrator subscribes to the Blockchain network to receive real-time updates about events using the Blockchain listener component. The subscription mechanism can involve subscribing to a specific event or a wildcard event.
- **Event Validator:** The event validator component performs necessary checks and validations on the parsed event data to ensure its integrity and authenticity. It verifies the event's signature, checks for any inconsistencies or anomalies, and confirms that the event adheres to the expected format and protocol. If an event fails validation, it may be rejected or flagged for further investigation.
- **Event Parser:** The event parser component receives the raw events from the Blockchain listener and extracts relevant information from them. It parses the event data, including event type, transaction details, event parameters, and any associated metadata. This component transforms the raw event data into a more structured format that can be easily processed and understood by other components.
- **Event Processor:** The event processor component handles the business logic associated with each event. It receives the validated event data and sends the event to the Broker component.

Event broker

The Event broker is a message-oriented middleware, which serves as a mediator between publishers and subscribers, facilitating the transmission of events across various components of the system. This sub-component forms the foundation of event-driven architecture, as all event-driven applications rely on an event broker in some capacity to exchange information.

Focusing on a HLF network, as the one considered in TRUSTEE, an event broker plays a crucial role in facilitating communication and information sharing among network participants. It broadcasts and delivers events related to the network's transactions, state changes, and other relevant activities. The event broker ensures that participants are notified in real-time about specific events they are interested in. Network participants, such as applications, smart contracts, or client programs, can register their interest in specific events by subscribing to the event broker. This can be done through an API, or the SDK provided by the HLF framework. When an event occurs, the event broker receives the event information and publishes it to all registered subscribers. The event broker acts as a message bus, ensuring that the events are efficiently propagated to interested parties. The event broker can provide flexible filtering mechanisms, allowing subscribers to define their criteria for event selection. Subscribers can specify conditions based on the event type, transaction ID, specific attributes, or any other relevant parameters. This enables participants to receive only the events they are interested in, reducing unnecessary network traffic, and improving efficiency.

The main components are the following:

- **Broker:** The broker acts as an intermediary between the orchestrator and the storage system. It receives processed events from the event processor and queues them for storage. The broker ensures reliable delivery and provides functionalities such as event buffering, ordering, and persistence. It abstracts the storage layer and provides a unified interface for the orchestrator to interact with different storage systems.
- **Storage System:** The storage system component is responsible for persistently storing the events received from the broker. The storage system stores the events in an organized manner for future retrieval and analysis.

TRUSTEE Blockchain Agreement Transaction Monitoring

STM is redirected to the Blockchain agreement transaction monitoring to monitor the agreements in the Blockchain which is a web portal providing graphical interaction to the audit and monitoring system. In fact, the user is redirected to the monitor tool's user interface through the dashboard.. It acts as a surveillance system, providing real-time or near-real-time insights into the Blockchain's transactions, blocks, and other related data. The monitor gathers data from the

HLF network. This can be achieved through various methods, such as connecting to Blockchain nodes, or other data providers. The monitor processes and analyzes individual transactions taking place on the Blockchain. It extracts relevant information, such as timestamps or transaction IDs. The monitor also keeps track of newly mined blocks, ensuring that the Blockchain is functioning correctly. They can detect and notify users about potential issues like forks or network congestion. By monitoring the network's health, they help ensure the reliability and stability of the Blockchain.

The functionality of a metrics dashboard typically includes the following:

- **Data Visualization:** The dashboard presents metrics and data in the form of visual elements such as charts, graphs, tables, gauges, or heatmaps. Visualizations help users quickly grasp trends, patterns, and comparisons, making it easier to derive insights and make data-driven decisions.
- **User Authentication and Access Control:** the monitoring tool may require user authentication and access control mechanisms sent by AM. This ensures that only authorized individuals or groups can access and view sensitive or confidential metrics.
- **Responsive Design and Device Compatibility:** A well-designed dashboard should be responsive and compatible with different devices and screen sizes. It should adapt its layout and visualizations to provide an optimal user experience across desktops, laptops, tablets, and mobile devices.

Figure 186 summarizes the security and audit component with its different sub-components:

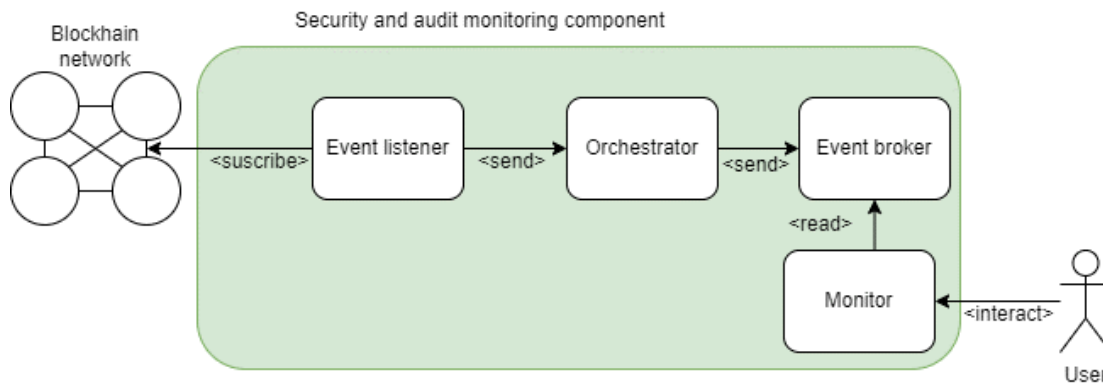


Figure 186: Security and audit monitoring architecture

7.5.2. OPERATIONAL DESCRIPTION

The STM provides two main functionalities. The first functionality is the creation of a dataset agreement. The second functionality is the signing of a dataset agreement. The following sections describe both functionalities in greater details with their corresponding sequence diagrams.

7.5.2.1. CREATING A DATASET AGREEMENT AS A DATA PROVIDER

Before the creation of a dataset agreement, the Data Provider must first navigate to the STM from the DA. Consequently, a user ID or a user object is sent to the STM to identify the user within the STM. Following, the Data Provider must first create an agreement related to the dataset that will be processed before any data processing can begin. The data provider can set the constraints and allowed processes within the agreement to determine by whom the

dataset can be processed, how the dataset can be processed, under what conditions the dataset can be processed and for how long the dataset can be processed.

Creating the dataset agreement starts at the STM's Flutter web application. Here, the Data Provider selects the dataset to which an agreement should be associated with. Additionally, the title of the agreement, the start and end dates, the allowed purposes, user roles and actions can be defined by the Data Provider in the Flutter web application. After completing the agreement creation process, the Data Provider submits their newly created dataset agreement and the STM's Flutter web application leverages the STM's internal REST API, using a HTTP POST endpoint "/api/v1/agreements" to send and create a persistent dataset agreement in the STM's GraphDB. Thereafter, if all of the information inserted by the Data Provider is correct, the STM's Java Spring Server responds to the STM's Flutter web application with a successful message, triggering the Flutter web application to display the newly created dataset agreement to the Data Provider. The dataset agreement creation process is further illustrated in Figure 187.

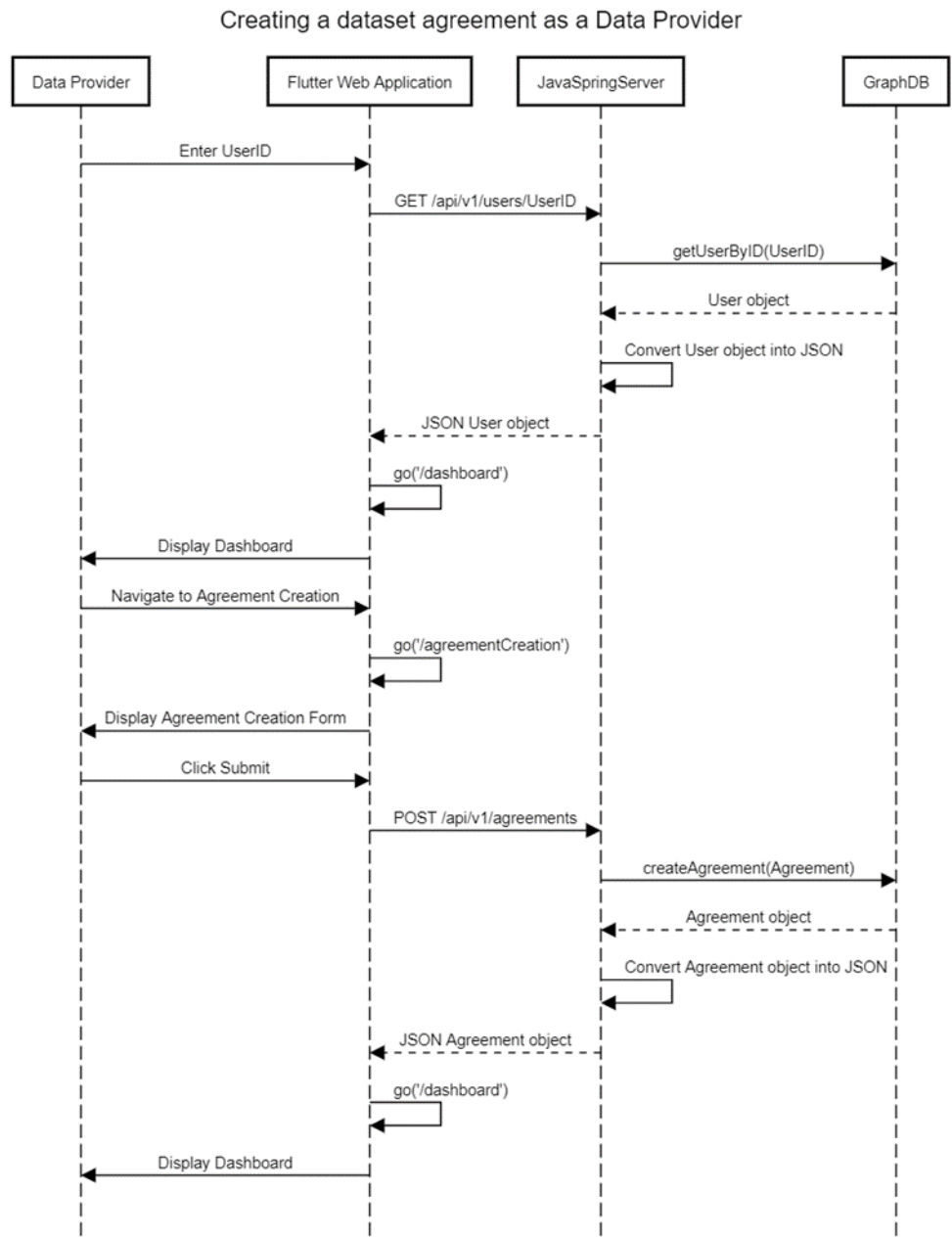


Figure 187: The sequence diagram for creating a dataset agreement as a Data Provider

7.5.2.2. SIGNING A DATASET AGREEMENT AS A CONSUMER

Initiating the data processing of a dataset requires the agreement and signature of both the Data Provider and the Consumer. The signing process is depicted in Figure 188. To sign a dataset agreement the Consumer may select the relevant dataset and see within the STM what agreement has been created for the selected dataset. Several agreements can be created for one dataset, meaning the Consumer must select an agreement should there be more than one. After selecting and reading through a dataset’s agreement the Consumer can sign the agreement, which performs an HTTP POST request to the STM’s internal API endpoint “/api/v1/agreements/AgreementID/sign. Consequently, the STM’s

Java Spring Server is triggered to create a signature and store persistently within the STM’s GraphDB. Lastly, once the signing process has been completed, the Flutter web application is notified that the signature has been successfully created and displays the agreement that the Consumer has signed in the Consumer’s internal STM agreement dashboard.

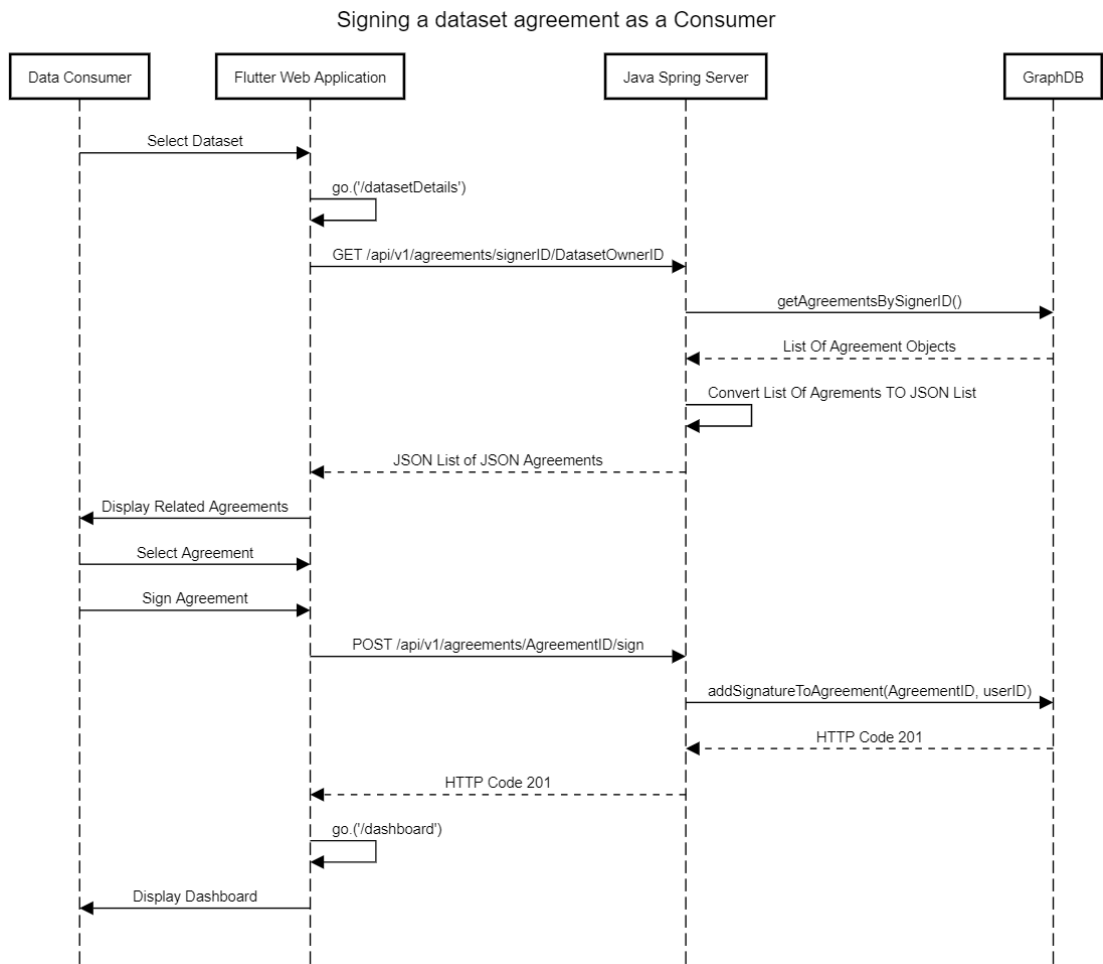


Figure 188: The sequence diagram for signing a dataset agreement as a Consumer

7.5.2.3. MONITORING BLOCKCHAIN AGREEMENTS

The main processes of the Security and Audit Monitoring Platform are as follows:

STORE BLOCKCHAIN NETWORK CREDENTIALS

Figure 189 describes the process of getting and storing credentials from the Fabric CA, which is needed to establish a connection with the HLF Blockchain network.

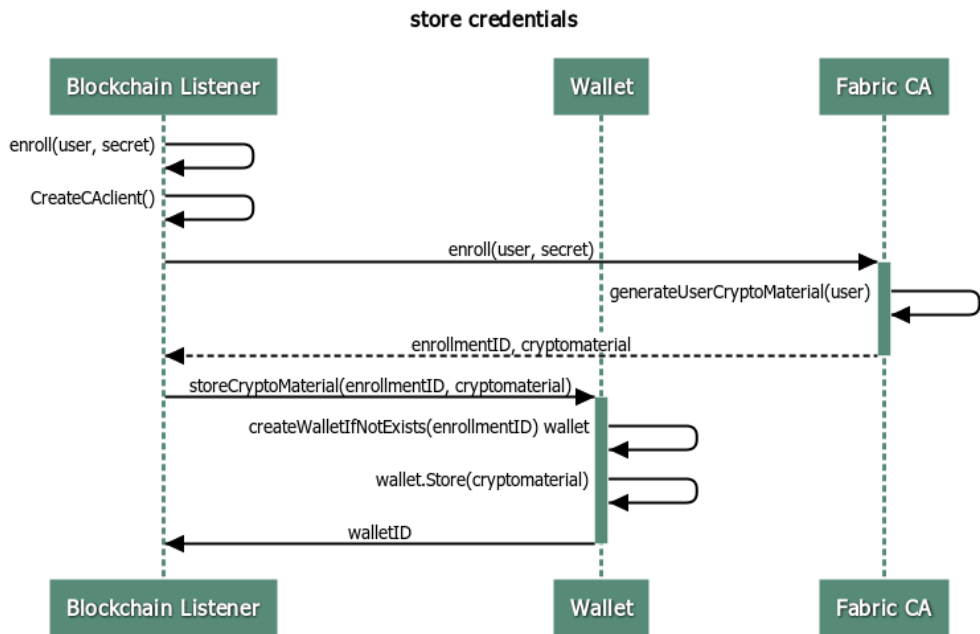


Figure 189: Store Blockchain Network Credentials sequence diagram

SUBSCRIBE TO CONTRACT EVENT

Figure 190 describes the process of connecting to the Blockchain network using the stored credential in the wallet and subscribing to an event emitted by the Smart Contract.

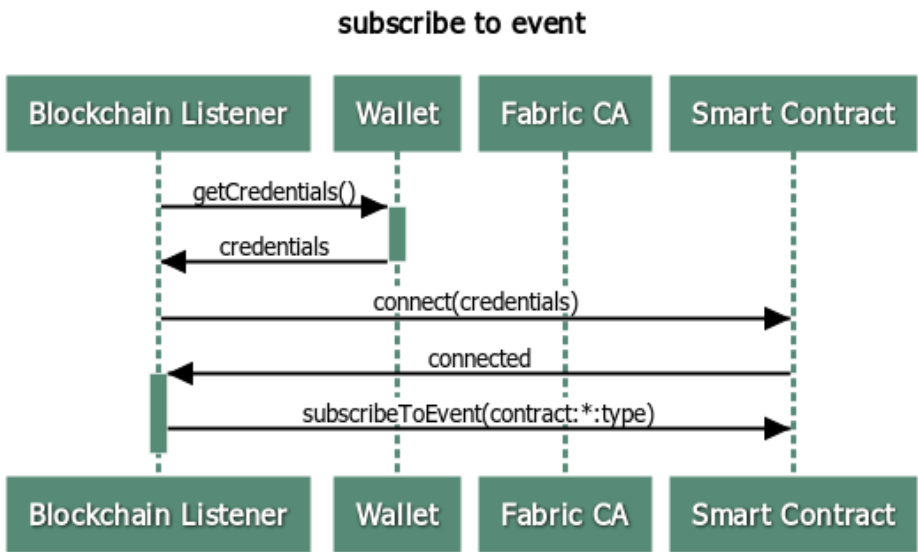


Figure 190: Contract Events Subscription sequence diagram

STORE CONTRACT DATA

Figure 191 describes the process of receiving, parsing, and storing an event received from the Smart Contract.

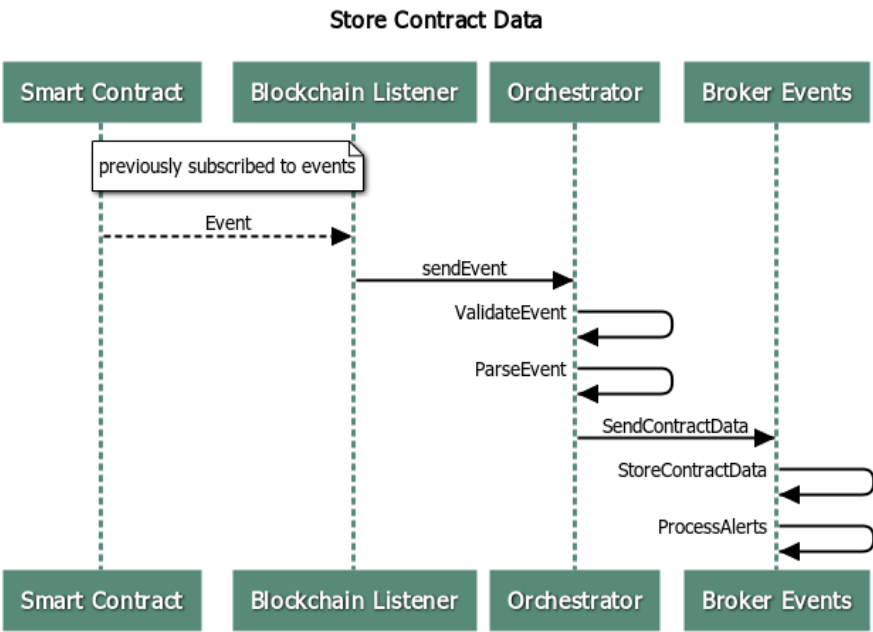


Figure 191: Store Contract Events sequence diagram

ACCESS CONTRACT DATA

Figure 192 describes the process of accessing Contract data from the TRUSTEE Blockchain Agreement Transaction Monitoring.

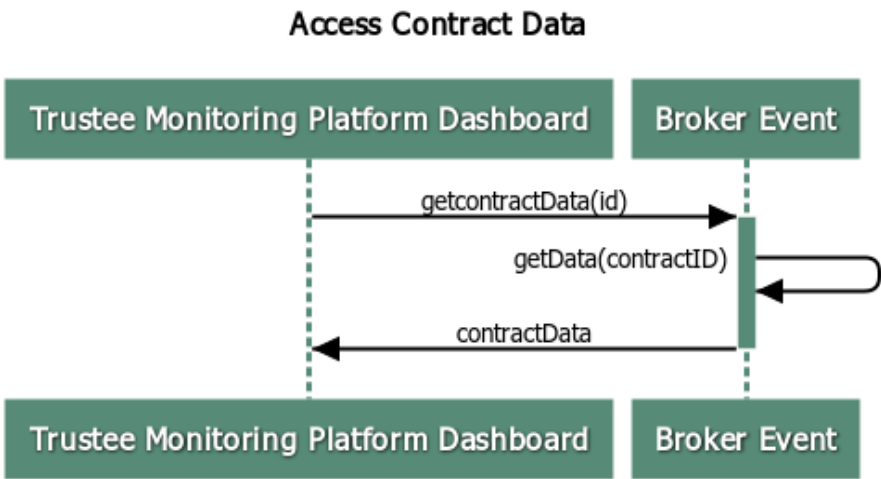


Figure 192: Access Contract Events sequence diagram

7.5.3. TECHNICAL DESCRIPTION

7.5.3.1. IMPLEMENTATION DETAILS

This section briefly elaborates on the methodology used to develop the STM components. The methodology is inspired by the agile development methods, allowing the STM's development process to be iterative and flexible for any necessary changes that might need to be performed in the future. Figure 193 depicts the methodology used for developing the STM.

Before any implementation starts, the user requirements are analyzed, so that the STM fulfills its functionalities for the TRUSTEE platform. Following, a common data structure is required. GDPR-compliant agreements are complex data structures and can therefore be modelled using ontologies as a data structure. Once the agreement ontology has been created, the implementation of the STM's Java server and the STM's Flutter web application commences. Both the Java server and Flutter web application are developed in parallel and later on integrated with each other through the Java server's REST API. As soon as the integration is successful the next functionality can be implemented and tested. The testing for the Flutter web application and the Java Server can be done simultaneously through the REST API.

Furthermore, when a functionality has been tested, then the next functionalities can be implemented to further extend the STM's capabilities. Lastly, a review on the STM's state is done and any missing functionalities can be added, resulting in an iterative development process for the STM.

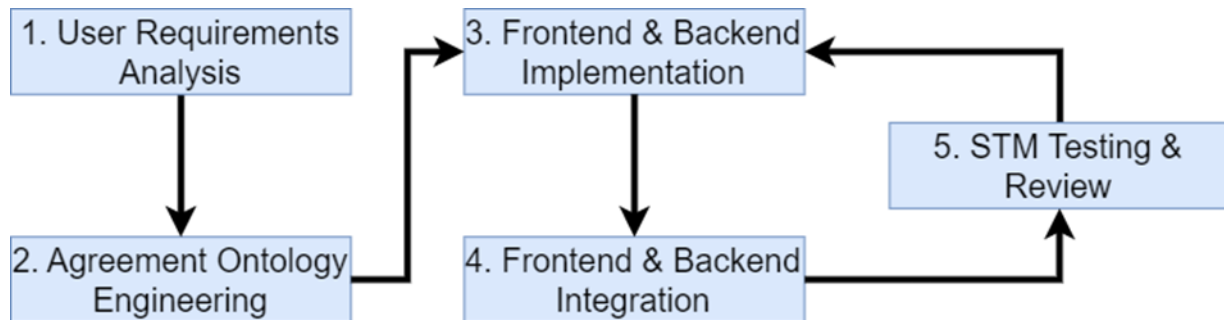


Figure 193: The methodology used to develop the STM.

The following subsections will further elaborate on the STM components. Firstly, a general description of the architecture is given. Adding on, a deeper explanation of each individual component is presented.

GRAPHICAL USER INTERFACE

In order to fulfill the STM requirements a GUI is required. The GUI is provided by the previously mentioned Flutter web application. However, before implementing the GUI and all of the web application's functionalities, it is essential to initiate the development with a mock-up. The mock-up of the Flutter web application is created using the free licensed version of the Figma prototyping tool. Figma enables fast GUI prototyping to progress throughout the development lifecycle. The Figma prototype only serves to provide a faster development of the look and feel of the Flutter web application and does not provide any functioning features.

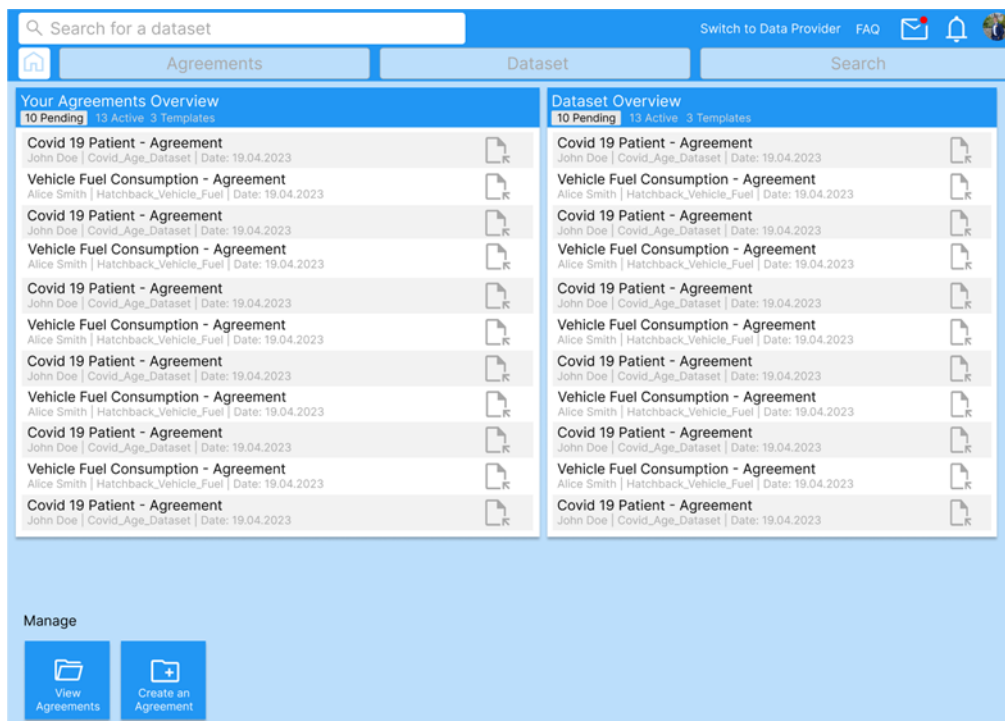


Figure 194: The prototype Dashboard Page for the STM using Figma.

The focus during the prototyping is to first create the main screen in the STM web application. The main screen is the *Dashboard Page* displaying all relevant information to the user without requiring the user to perform any actions to navigate throughout the application to improve the user experience and user-friendliness of the Flutter web application. Figure 194 depicts the prototype of the *Dashboard Page*, within which a user may see all their currently active agreements and their accessible datasets. At the bottom of Figure 194 the prototype also enables the user to create additional agreements with a button. This button then navigates the user to a new page with an agreement creation form.

DATA MODELS

The STM stores 8 different data objects (Figure 195). All data objects are structured to fit to GDPR's requirements for a data sharing agreement to be considered legal. The *Purpose* and the *Action* data objects are linked to an *Agreement* object. With *Purpose* and *Action*, it is possible to perform rule-based checks to analyze if an agreement is valid and if two or more datasets may be merged for data processing based on their corresponding agreement. The *Agreement* object contains a *datasetID* to identify which dataset is linked to the agreement. Furthermore, a *Signature* object is created to link an *Agreement* object to a *User* object. The *Metadata* object represents the metadata that the STM will receive from the HEDF building block. The STM then internally constructs a *Dataset* object with the help of the *Metadata* object to enable users within the system to view overall information of a dataset before signing the corresponding agreement. Lastly, should a user belong to an organization and act on behalf of the organization, then the STM associates the user with the organization by linking the *Organisation* model to the *User* model with the *organizationID*.

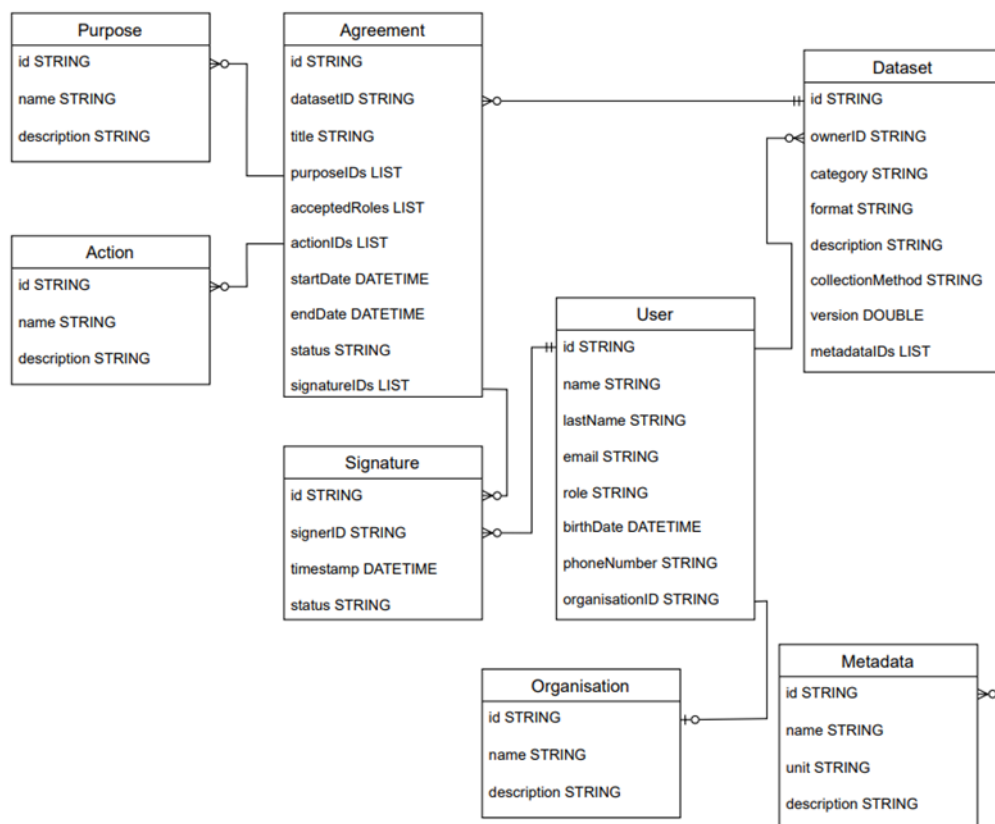


Figure 195: All 8 data objects and their relationships toward each other in the STM

SECURITY AND PRIVACY-BY-DESIGN

The STM core principles are to enable and secure legal data sharing under the GDPR's regulations, by managing data sharing agreements and using the agreements to analyze whether data sharing can start or not. Security-by-Design and Privacy-by-Design are two methodologies that are used as a guide throughout the designing and implementation of the STM.

- Security-by-design. The Security-by-design methodology is a guide to ensure that security aspects are considered from the very beginning of the development of a system.

Privacy-by-design. It is related to Art. 25 of GDPR [32] of the data protection through technology design which is the basis of the STM, especially the agreement manager. The requirements should be specified in the Terms of Use of the agreement manager. The user must accept or decline the use of its identity user data for security and trust purposes as well as their processing, the period of their storage and their accessibility.

7.5.3.2. INPUT/OUTPUT

AUTHENTICATION MANAGER

The AM is used for the connection and use of the STM using eIDAS over SSI developed in the AM component. eIDAS is a EU regulation on electronic identification and trust services for electronic transactions.

SMART CONTRACT AGREEMENTS

The ATR is used by the STM to ensure that signed agreements are not altered. The ATR can serve as a logging and tracking tool to prove whether or not agreements have been signed in the past, as well as guarantee that the agreements have not been manipulated without the confirmation of all parties involved in an agreement.

The STM can send JSON-formatted data to the ATR indicating who signed which agreement at what time, under which conditions, and for which dataset (see Figure 196). The sent information contains the IDs and timestamps of all the necessary information to be able to monitor the agreements. Later, the STM could request the IDs from the ATR and provide the correct agreements based on the stored IDs in the ATR, since the STM stores the complete objects related to the IDs within the STM's GraphDB. The following JSON object can be sent to the ATR:

```
{
  "ProviderID": " STRING or INTEGER ",
  "ConsumerID": " STRING or INTEGER ",
  "OrganisationID_Provider": "STRING or INTEGER",
  "OrganisationID_Consumer": "STRING or INTEGER",
  "DPIA_ID_Provider": "STRING or INTEGER",
  "DPIA_ID_Consumer": "STRING or INTEGER",
  "AgreementID": "STRING or INTEGER",
  "DatasetID": "STRING or INTEGER",
  "PurposeIDs": [ "STRING", "STRING"],
  "ActionIDs": [ "STRING", "STRING" ],
  "IsPersonalData": "Boolean",
  "START_TIMESTAMP": "DATETIME in ISO 8601 format",
  "END_TIMESTAMP": "DATETIME in ISO 8601 format",
  "AgreementStatus": "STRING"
}
```

Figure 196: An overview of the Smart Contract Agreement data model

The “ProviderID” and the “ConsumerID” represent the Data Provider’s unique ID and the Data Consumer’s unique ID respectively. The “OrganisationID” is the unique ID of the Data Provider. The “OrganisationID” can be NULL if the Data Provider does not belong to an organisation. The “AgreementID” is the unique ID to the agreement object within the STM’s database, which has a set of “ActionIDs” and “PurposeIDs” that determine the constraints of the agreement. The “DatasetID” is the unique ID of a dataset’s metadata which is used to associate the agreement with the dataset metadata. “IsPersonalData” represents a Boolean flag to directly determine if the dataset contains personal data. Both the “START_TIMESTAMP” and the “END_TIMESTAMP” are used to store the length of an agreement within the STM and the ATR. Lastly, to inform any monitoring entity about the status of the agreement within the ATR we add the “AgreementStatus” in the JSON data.

SMART CONTRACT AGREEMENT MONITORING

An ATR will be implemented utilising HLF, a permissioned Blockchain framework, to record every single transaction of TRUSTEE. Thus, it is crucial to have an understanding of the Blockchain network by visualising and monitoring the transactions stored. Hyperledger Explorer is the first-ever Blockchain explorer tool for permissioned ledgers and can be configured to integrate with HLF (see Figure 197). It is a web application tool which provides a rich user interface enabling authorised entities to obtain significant information from the Blockchain. It enables obtaining the status of the transactions, viewing transaction information, searching transactions, inspecting newly generated blocks, filtering blocks based on the set criteria, acquiring real-time notification of newly generated blocks, etc

The explorer comprises various functionalities, such as the Dashboard, which can be leveraged to show the latest activities of the Blockchain. The built-in navigation tab is separated into various parts such as blocks, transactions, chaincodes, and channels, each providing distinct functionality of the Blockchain. Similarly, the notification feature informs about the latest activity performed and transactions recorded by a particular organisation. In addition to that, one can obtain information about historical metrics, the latest activity per selected channel, blocklist and so on. Thus, the transaction list feature will allow TRUSTEE to navigate to the list of all transactions recorded on the ATR, specifically linked to particular organisations, channels and more.

The explorer is embedded with the latest technologies such as ReactJS, Google material UI, web-sockets, and so on. It is a blockchain module which can listen and query for the transactions recorded for TRUSTEE. The explorer provides REST API endpoints in order to obtain particulars and metrics from the Blockchain. The REST API is outlined to be utilised within the Node.js JavaScript runtime environment. It leverages the built-in middleware function in Express to appropriately handle incoming requests which consist of JSON payloads. The REST API implements security with a token and thus any request made to the REST API should comprise the token created by the API

Hyperledger Explorer is an effective solution to monitor, visualise and analyse, and can be integrated to obtain various blockchain functionalities, such as:

- **Monitoring and Visualisation:** The explorer interface can help to monitor and visualise the inner functioning of the blockchain network. Every transaction recorded to the blockchain ATR can be monitored and any anomalies in the network can be spotted instantly.
- **Transaction Analysis:** Every recorded transaction can be analysed by reviewing the size, timestamp and content. The acquired data can be beneficial in identifying patterns, detecting fraudulent activities and improving network efficiency.
- **Network Assessment:** Full overview of the blockchain network can assist in identifying bottlenecks, improve performance and enhance the overall effectiveness of the network.
- **Security and Compliance:** Hyperledger Explorer provides tools to establish the security and regulatory compliance of the blockchain. E.g., TRUSTEE can leverage such functionalities to comprise access control, monitor user identities and also detect irregularities on the blockchain network.
- **Customisation:** The explorer can be customised to meet the user requirements by elevating network performance, integrating additional tools and so on.

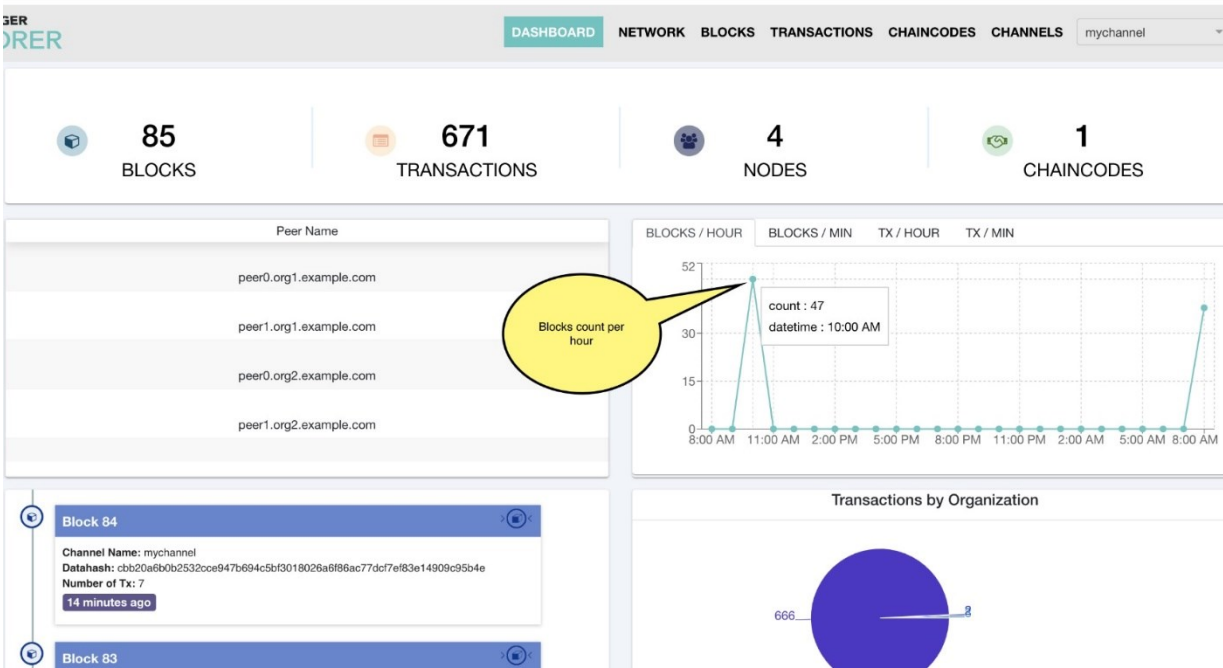


Figure 197: An overview of the Hyperledger Explorer Dashboard

DATA ACCESS IN DATA SPACES

In the architecture of the STM, the International Data Spaces Association (IDSA) principles play a pivotal role in guiding data access in data spaces. This influence is particularly evident in the operations of the STM's agreement management platform, where it promotes secure and standardized data exchange while adhering to GDPR compliances. The STM operates by handling request and collection of agreements, validating their ongoing legitimacy, and reassessing them if they're withdrawn, expired, invalidated, revoked, or if modification, confirmation, or reaffirmation is required. These agreements define the framework for data providers to determine the extent of data access granted to various data consumers.

The STM's adherence to IDSA principles is evident in its comprehensive management of GDPR-based agreement lifecycles. Here, agreements initiated by the data provider and signed by the data consumer are managed securely and transparently, facilitating a more efficient and trustworthy data exchange. Additionally, the user-friendly interface of STM, built on IDSA guidelines, enables data providers and consumers to navigate their way through data access controls seamlessly. The STM platform also assures machine-readable data format compatibility with GDPR regulations, contributing to the overall efficiency of the data access process. With the STM tool, TRUSTEE will successfully embrace IDSA's vision of a secure and self-determined data exchange. By providing stakeholders total control over their data within a transparent and GDPR-compliant environment, STM represents a significant step forward in the evolution of data spaces and data sovereignty.

STM and International Data Space (IDS) Connectors

Expanding upon the existing STM architecture, the incorporation of IDS connectors further enhances data exchange by providing a standardized way of accessing and transferring data. These IDS connectors act as an interface between the STM and the data sources, allowing the secure and controlled transmission of data in accordance with IDSA principles. In the TRUSTEE ecosystem, IDS connectors work in tandem with the STM, complementing its operations.

As data requests are initiated and agreements are forged between data providers and consumers, IDS connectors facilitate the actual transfer of data. They ensure that data transactions adhere to the terms and conditions specified in the agreement and that the data's integrity and security are maintained during the transfer.

IDS connectors also play a crucial role in enhancing the STM's GDPR-compliant operations. The connectors ensure that each data transaction complies with GDPR provisions by following data sovereignty principles. For instance, they check for explicit consent from the data providers, uphold the "purpose limitation" principle by ensuring data is used only for the agreed purpose, and enable data erasure if the data subject invokes their "right to be forgotten". Moreover, these connectors support a machine-readable format of GDPR laws, thereby facilitating automated GDPR compliance checks and enhancing the efficiency and reliability of the data exchange process. Thus, the integration of IDS connectors into the STM not only empowers TRUSTEE stakeholders with total control over their data but also ensures secure, seamless, and compliant data exchange across the platform. This constitutes a significant stride toward realizing the vision of self-determined, trust-based data spaces in line with the principles set forth by IDSA.

In Figure 198, an indicative interrelation is depicted.

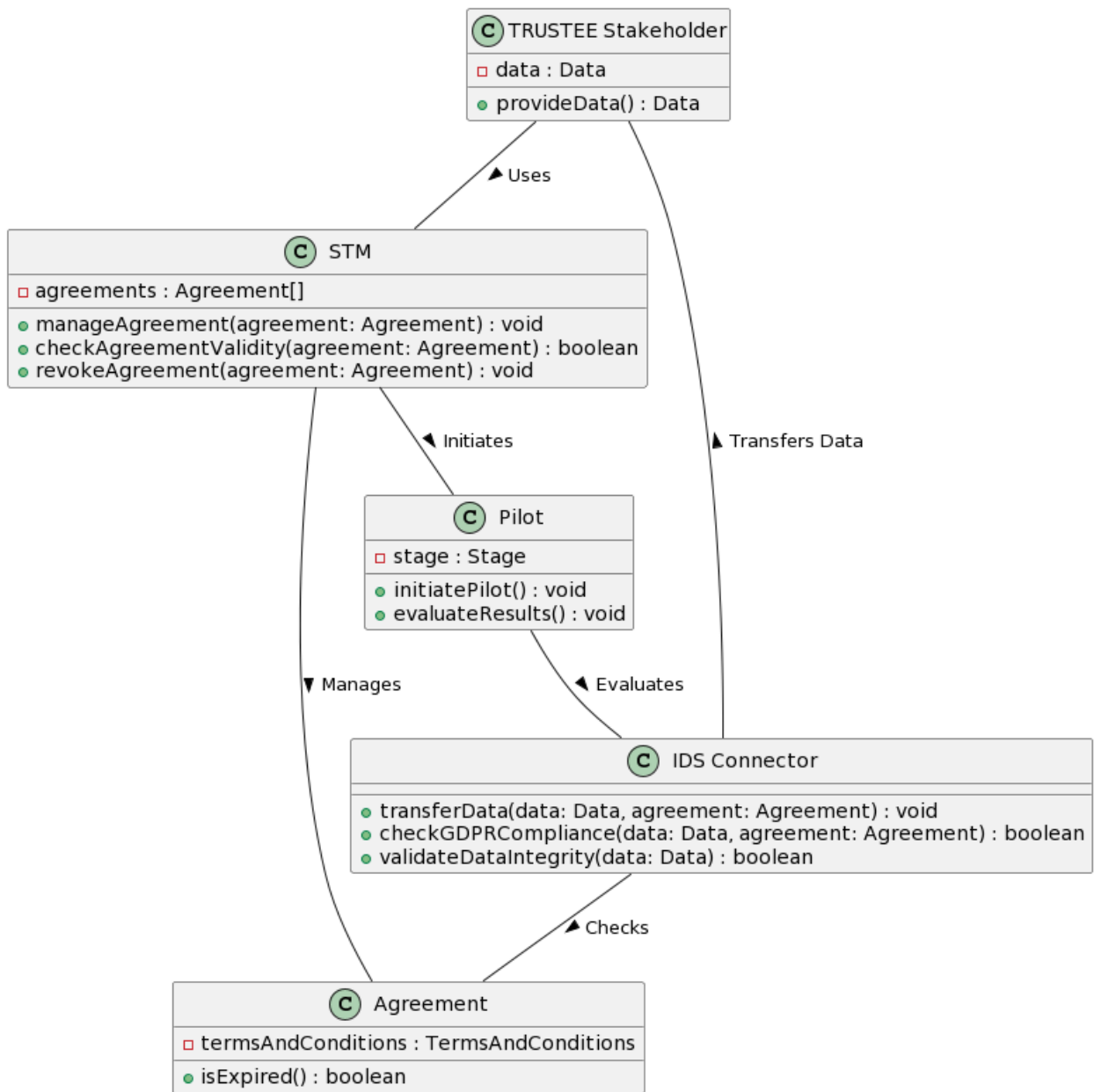


Figure 198: STM and IDS Connectors Interrelationship

DASHBOARD

The primary UI of the DAN ABB is used as an initial way for the user to access the STM user interface. With the use of a button and/or a link or through the VA, the user will be redirected to a landing page of the STM depending on the initial intentions of the user to add data (Data provider) or to consume data (Data consumer).

TRUSTWORTHY AI SUPPORT DESIGN FRAMEWORK /GDPR COMPLIANCE CHECKER TOOL

Overview of GDPR compliance checker

The GDPR Compliance checker is an inner module of TAI-SDF. Therefore, an external interface has been designed between GDPR Compliance checker and STM as shown in Figure 199. The report of the GDPR compliance checker will provide STM with a compliance score and also all the non-compliances and the partially compliances to GDPR requirements, with related relevant information, descriptions, and evidence through two different JSONs as shown in Figure 200.

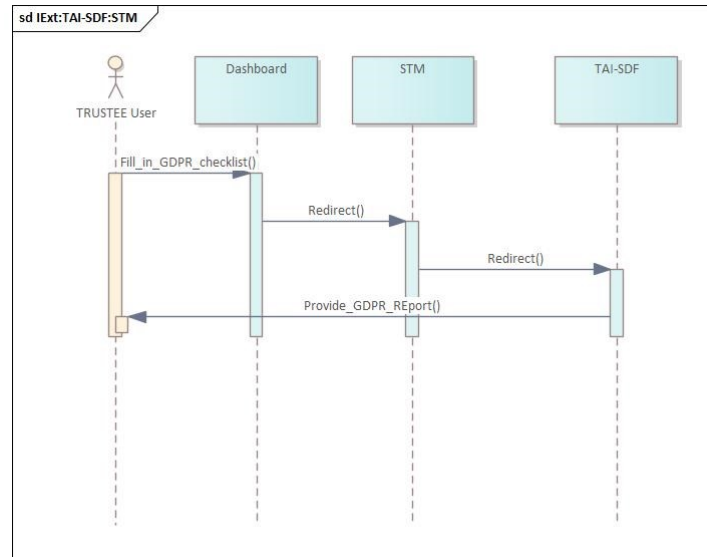


Figure 199: Sequence diagram STM- TAI-SDF

```

JSON 1: {
  questions:[id: number, text: string]
}

JSON 2: {
  answers: [idQuestion: number, response: string]
}
    
```

Figure 200: An overview of the TAI-SDF-STM data model

The operation flow is as follows:

- User Login & Consent - Users log in to the DA to access TRUSTEE functionalities. Consent is provided by the user where required.
- Initial Redirect to STM UI - Upon successful login and consent, users are redirected to the STM UI for further interactions.
- GDPR Compliance Check - Subsequently, users are redirected to the TAI-SDF UI. At this juncture, users perform a GDPR compliance check to ensure adherence to data protection regulations.
- Compliance Confirmation & Acknowledgement - Once GDPR compliance is confirmed, TAI-SDF sends an acknowledgement (ACK) to the Agreement Manager, including the GDPR compliance form ID.

- Agreement Creation - Finally, users are navigated back to the STM UI. Here, they complete the agreement creation process to finalize their interaction within the system.

PRIVACY IMPACT ASSESSMENT

To further adhere to the GDPR, the STM requests the DPIA to provide a PIA status. Here the link between the STM and the DPIA is created, where the DPIA sends the DPIA status, time of the DPIA and the Data Provider's ID to the STM in the JSON format gathered in Figure 201.

```
{
  "UserID": "192-423-df4",
  "DatasetID": "1987-05-cf1",
  "statusDPIA": "completed",
  "timestamp": DATETIME in ISO 8601 format
}
```

Figure 201: An overview of the DPIA-STM data model

From the formal and legal perspective, the communication between STM and DPIA will provide the information whether data processing activities planned by the user (Data Provider and Consumer) with the use of specific type of data shared by Data Provider are compliant with the data protection regime as well as whether they correspond with the requirements which may be set out by the Data Provider and which could be a prerequisite for enabling data processing by the Consumer. It should be stressed that DPIA should be conducted prior to the data processing.

Based on the example where the Data Provider and the Consumer would sign an agreement on a specific data re-use by the latter, the provision of the output of a DPIA to the STM should ensure that the Data Provider is well-informed about the outcome of the Consumer PIA's process, knows what kind of data processing activities the Consumer plans to conduct with the use of the data shared by the Data Provider, knows the risks associated with it and the mitigation measures, and based on that can consider whether to grant the Consumer the access to the data and allow for performing computation with the use of / on Data Provider's data or not.

7.5.3.3. API

The RESTful API provided by the backend application of the agreement manager is the main communication interface for the Flutter-based frontend as well as the communication between the agreement manager and the rest of application building blocks. Designing an API that follows conventions and best practices is crucial for the STM as well as for the integration with the rest of the applications. See ANNEX 6: STM ENDPOINTS for more details.

OPERATIONS

The operations are represented by the standard HTTP methods GET, POST, PUT and DELETE. Table 66 shows the description of each method.

Table 66: A description of the standard HTTP methods.

HTTP method	Operation
-------------	-----------

GET	Retrieves data from the server. When you send a GET request, you're asking the server to fetch the specified resource. For example, you might use a GET request to retrieve the details of a specific user.
POST	Sends data to the server to create a new resource. When you send a POST request, you're asking the server to accept the data enclosed in the body of the request as a new object of some kind. Using the example, a POST request is used to add a new user to the database.
PUT	Updates a current resource with new data. A PUT request replaces all current representations of the target resource with the request payload.
DELETE	Removes data from a server. A DELETE request asks the server to delete the specified resource. Using the example again, a DELETE request is used to delete a user from the database.

This separation based on HTTP methods performs operations on one API endpoint.

REQUEST AND RESPONSE PAYLOAD

The payload of operations requests and responses is in JSON format. It returns a specific HTTP status code based on the response type which indicates if the operation was successful (status 200 or 201) or not. If the operation fails, then the status code indicates the reason for the failure (e.g. 401 for not being authorized or 500 for a server error). The format of the response is standardized.

7.5.4. TECHNICAL DETAILS

The technical implementation information of the different components of STM is detailed in Table 67, Table 68, Table 69 and Table 70.

Table 67: Technical details of STM frontend

DEVELOPMENT – STM FRONTEND	
Type (Software/Hardware)	Software
Operating Systems	Windows 10 (64-bit)
Database and databases tools	None
GUI	Web-based application using a web browser (Microsoft Edge, Chrome)
Programming language	Dart
Development Environment	Visual Studio Code, Microsoft Edge
INSTALLATION AND DEPLOYMENT	
Software Requirements	Flutter (3.7.11)
Hardware Requirements	
Containerization	None
Communications	

Table 68: Technical details of STM backend

DEVELOPMENT – STM BACKEND	
Type (Software/Hardware)	Software
Operating Systems	Windows 10 (64-bit)
Database and databases tools	Graph database (GraphDB)
GUI	None
Programming language	Java
Development Environment	IntelliJ, Protégé, GitHub
INSTALLATION AND DEPLOYMENT	
Software Requirements	Java 20 JDK, GraphDB
Hardware Requirements	
Containerization	None
Communications	

Table 69: Technical details of STM Server

DEVELOPMENT – STM Server	
Type (Software/Hardware)	Software
Operating Systems	Windows 10 (64-bit)
Database and databases tools	None
GUI	None
Programming language	Java
Development Environment	IntelliJ, GitHub
INSTALLATION AND DEPLOYMENT	
Software Requirements	Java 20 JDK
Hardware Requirements	
Containerization	None
Communications	The server allows to connect the frontend and backend. It offers a RESTAPI for different endpoints such as getting and posting agreements.

Table 70: Technical details of STM connector

DEVELOPMENT – STM connector	
Type (Software/Hardware)	Software
Operating Systems	Windows 10 (64-bit)/Linux
Database and databases tools	None
GUI	None
Programming language	Java

Development Environment	IntelliJ, GitHub
INSTALLATION AND DEPLOYMENT	
Software Requirements	Java 20 JDK
Hardware Requirements	No
Containerization	None
Communications	The STM connector is an API used for connecting the STM with the rest of ABB connected to it.

7.6. VALIDATION

The Testing Pyramid shown in Figure 202, represents the ideal distribution for the STM. First and foremost, the majority of tests should be automated to ensure repeated execution on new software versions and promptly detect any regressions. Only the top of the pyramid consists of manual tests, which are still necessary to identify edge cases and explore alternative paths but require human effort and are therefore performed on a smaller scale compared to automated tests.

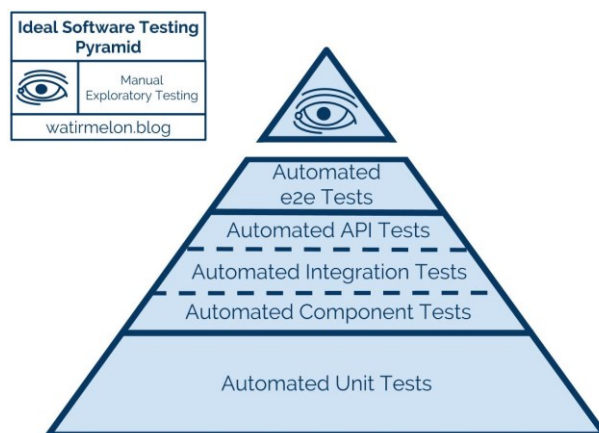


Figure 202 Testing Strategy⁵⁶

Regarding automated tests, the principle is that most test cases should be covered by the lower parts of the pyramid, which include tests that are less costly in terms of time and execution resources and can cover various detailed scenarios. This is the case, for example, for unit tests. As we move up the pyramid, tests become more expensive and time-consuming, so they should be limited to the most common and critical execution paths. Automated end-to-end tests fall into this category.

7.7. NEXT STEPS

⁵⁶ <https://servian.dev/the-test-pyramid-and-data-engineering-with-julia-e4678c3f8dff>

This section presents a roadmap that outlines the next steps to continue the development of STM. Following a list with the main points to be implemented:

- Adding roles, actions, and purposes to the agreements.
- Implementation of AI Model agreements.
- Implementation of Trusted timestamp and integration of signature.
- Collection of the consent for KR storage.
- Improve the GUI: Adjust to UX Guidelines, Separate Provider and Consumer Uis, etc.
- Update the agreement ontology.
- Completing data models design and API communications with other ABBs (ATR, KR, AM etc.).

Following versions of the STM implementation will be gathered in D4.2 and D4.3.

7.8. CONCLUSIONS

The STM is a GDPR-by-design component, responsible for creating and managing the request and collection of agreements. The STM periodically checks the continued validity of an agreement to permit its use via the agreement management tool. The STM part presents the internal and external components and communications between them. Internal communications are related to the agreement management tool which is the kernel of the STM. There are six TRUSTEE components and external communications used for security, trust, and transparency reasons. Finally, the STM covers the end-to-end data and AI agreements life-cycles.

8. PART VI: ONE STOP SHOP

8.1. INTRODUCTION

This chapter provides information about the OneSS ABB of the TRUSTEE platform.

8.1.1. NEED AND OBJECTIVES

To assure TRUSTEE's user adoption, it is necessary to provide SDK which will enable developers to use the TRUSTEE platform in their own applications in an easy and straightforward way. Also, a high quality documentation containing relevant information about TRUSTEE architecture and functionalities needs to be provided, as well as user guides, tutorials, code samples, etc.

In addition to the needs related to the user adoption and usability of the platform in general, OneSS is also aiming to cater to the demands of growing the data economy and enabling data exchange between the existing European data spaces in a secure and trustworthy way, which will enable the development of new services with added values for the end-users and better utilization of available data.

To address the stated needs, the OneSS has the following objectives:

- Provide TRUSTEE SDKs that can be used by third parties.
- Provide documentation, libraries, tutorials, and guides for developers who want to use TRUSTEE platform.
- Enable communication with European Cloud Spaces and provide support of the data economy growth by enabling secure and trustworthy data exchange between TRUSTEE and external data spaces.

8.1.2. INNOVATION

Since one of the main objectives of the OneSS is to enable communication with European Cloud Spaces, TRUSTEE's OneSS aims to drive innovation in the area of enabling such communication in a secure and trustworthy way by providing external connectors and SDKs for developers who want to use TRUSTEE platform. Providing secure and trustworthy exchange of data with external data spaces and enabling developers to implement services based on such data exchange is also one of the key drivers for the development of data economy which TRUSTEE is aiming to contribute to.

8.2. BACKGROUND

The OneSS ABB is envisioned as an online repository that acts as the main entry point for the TRUSTEE technical users, and eventually the Developer Persona. The OneSS is intended to contain the Developer handbook, code repositories, a toolset including code samples, SDKs, libraries, API documentation, and the SDK connectors module. In this deliverable, we explain the different segments of the OneSS – namely, the TRUSTEE Developer Handbook, TRUSTEE SDK Internal Connectors, and TRUSTEE SDK External Connectors.

TRUSTEE Handbook is the component of the OneSS that contains relevant information about the TRUSTEE platform, namely its architecture and building blocks, as well as information that is relevant for the onboarding of new users, and information relevant for Developer Persona, such as tutorials, guidelines and best practices. The objective of this component is to provide relevant information about the TRUSTEE platform and facilitate the adoption of the platform by end-users.

OneSS SDK consists of internal and external connectors which are meant to enable access to the TRUSTEE platform by third parties, as part of TRUSTEE's incremental strategy, and communication with external systems (external connectors), as well as to enable the connection with the different components of the TRUSTEE platform (internal connectors). As the internal connectors, due to their functionality, will be developed as the implementation and maturity of TRUSTEE platform components and their integration progress, more details on the internal connectors will be provided in the next versions of this deliverable, namely D4.2 and D4.3. In the current version of this document, the OpenAPI specification which will be used for their implementation is specified.

For the external connectors of the OneSS SDK, this version of the deliverable presents information on the investigation of several European Data Spaces and external connectors related to them that have the potential to be developed as TRUSTEE SDK external connectors. Integration cases are presented where possible, due to the analysis and external system's maturity level. Further details regarding the implementation of external connectors will be presented in the next versions of this deliverable, namely D4.2 and D4.3, as the investigation and implementation move forward.

8.3. REQUIREMENTS

In this chapter, the compliance of the OneSS ABB of TRUSTEE is investigated with regards to the user-centred, technical, legal, and socio-ethical requirements that were reported in D2.1 [1]. The results of this investigation provide further preliminary insight regarding with the aforementioned requirements addressed during the development and by the implementation of the OneSS and are presented in the following sub-sections. It is worth noting that only the Mandatory, Desirable, and Optional requirements have been considered at this point. Furthermore, only the currently compliant or partially-compliant requirements have been considered. As the implementation of the OneSS progresses during the development of the TRUSTEE Platform and the maturity of the ABB increases, the investigation of the compliance will evolve and further updates and/or enhancements will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

The preliminary results of the analysis related to the compliance of the OneSS with the user-centred, technical, and legal and socio-ethical requirements, as they are presented and defined in the D2.1 [1], are presented in this section. The tables containing requirements related to OneSS are available for each group of requirements. They contain a description of the requirements, as well as information on the compliance related to OneSS and how OneSS addresses the requirements. Due to the current progress of the OneSS, the focus is mostly on addressing the requirements with the Developer Handbook functionalities, however, as the maturity and development related to OneSS connectors will progress in the following months, the requirements list will be updated and information on how they are addressed by OneSS will be provided in more detail. Table 71 shows the user-centred requirements related to the related to the OneSS

Table 71: OneSS – User-centred requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
USR-058	TRUSTEE should enable users to access data catalogs, tutorials, and analysis outputs easily	Yes	OneSS covers the tutorials part of this requirement, by offering tutorials, guides and best practice information via Developer Handbook in the form of wiki pages.
USR-061	TRUSTEE should have access to support regarding platform functioning	Yes	Support regarding platform functioning is offered to the users by the Developer Handbook of the

			OneSS, in the form of tutorials, guides, and documentation.
USR-063	TRUSTEE should have example applications and tutorials for what is offering to be used by end-users	Yes	In the current contents available in TRUSTEE Developer Handbook, an initial set of tutorials is already provided. The content and number of tutorials will be updated and expanded in the following period. Code examples will be added in the following period.

Table 72 presents the preliminary results of the investigation regarding the compliance of the OneSS with the technical, functional and non-functional requirements specified in D2.1. The ID of the respective technical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether OneSS complies with the requirement and how. Further updates and/or enhancements regarding the approach presented below for addressing each technical requirement will be provided in upcoming versions of this deliverable, namely D4.2 and D4.3.

Table 72: OneSS – Functional and non-functional technical requirements

Req id	Req. Description	Compliance	How is addressed
Req-OneSS-NFUNC-StdAPIs	APIs and connection points of TRUSTEE ABBs shall be compliant with existing standards.	Yes	Using standardized specifications for OneSS connectors (Open API).
Req-OneSS-NFUNC-StdAPIDoc	APIs and connection points of TRUSTEE ABBs shall be documented according to existing established standards (e.g. OpenAPI for RESTful interfaces) using available open source tools.	Yes	Using standardized specifications for OneSS connectors (Open API). Additional documentation will be provided as the connectors' development will progress.

A preliminary analysis of the compliance with the legal and socio-ethical requirements defined in the D2.1 [1] is shown in ANNEX 2: LEGAL REQUIREMENTS and ANNEX 3: SOCIO-ETHICAL REQUIREMENTS, respectively.

8.4. ARCHITECTURE

8.4.1. HIGH LEVEL

The following figure (Figure 203) shows a high level view of the OneSS ABB architecture, focusing on the functionalities of the TRUSTEE Developer Handbook and the establishment of the connections with the systems outside TRUSTEE. The OneSS offers the functionalities of the Developer Handbook (implemented through the TRUSTEE Wiki pages), offering information such as documentation, best practices, tutorials, and guides.

Another part of the OneSS are SDK connectors. They simplify the integration process by providing a unified interface that abstracts the complexity of the underlying technology, in our case cloud-based TRUSTEE external or TRUSTEE internal technology. This enables developers to focus on the functionality of the application rather than the integration details. SDK connectors can be considered an umbrella term for custom connectors realizing interoperability and

integration. The main idea is to achieve efficient interoperability with a great user experience for the technical personnel doing the integration.

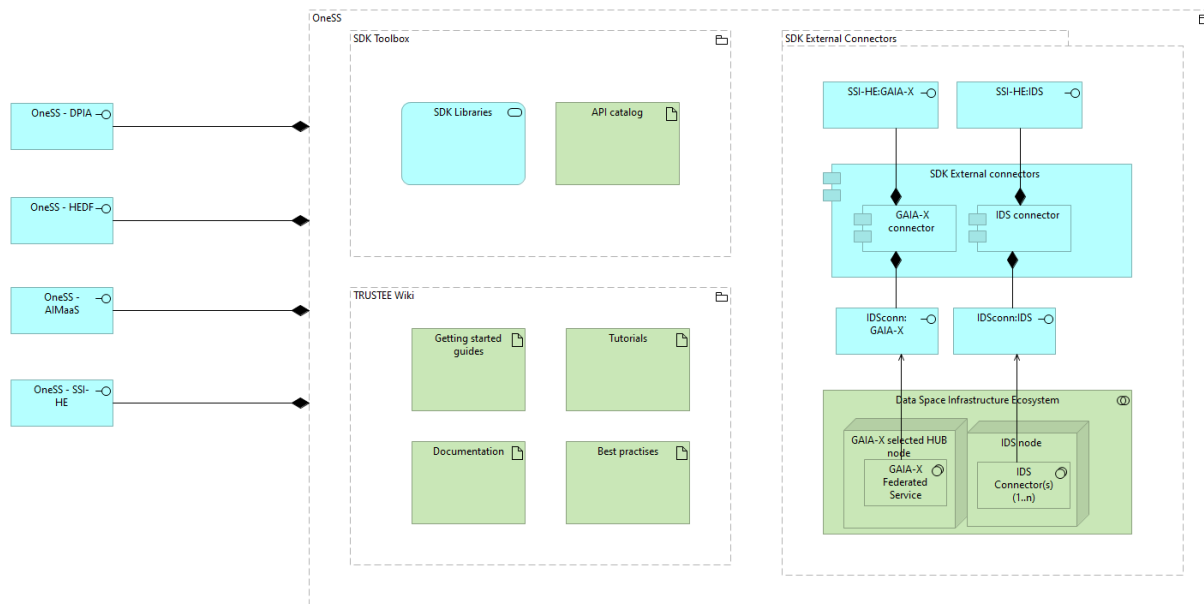


Figure 203: OneSS building block architecture (from D2.1 [1])

The external connectors are used for establishing a connection with external systems and dataspace. In this document, we describe several data spaces and systems that are under consideration for the implementation of external connectors used for integration with the TRUSTEE platform. In Figure 203, they are presented through the example of GAIA-X and IDS connectors in the SDK External Connectors block. Other than the external connectors, internal connectors offer the integration of building blocks within the TRUSTEE platform, and they are depicted in Figure 104 as providing the connection towards several TRUSTEE components (DPIA, HEDF, AIMaaS, HE). Details related to the internal connectors will be provided in the following versions of this deliverable, namely D4.2 and D4.3.

8.4.1. LOW LEVEL

This section brings a low level architecture of the OneSS subcomponents, focusing on the GAIA-X and IDS external connector and TRUSTEE Wiki (Developer Handbook). It is important to note that the external connectors implementation is still in progress and the version of architecture presented in this version of deliverable will be updated as the implementation advances.

8.4.1.1. DEVELOPER HANDBOOK

In terms of the Developer Handbook, by architecture we refer to the organization of the content within the Handbook. Current structure of the TRUSTEE Developer Handbook is presented in Figure 204. Further details on the implementation and content provided by the Developer Handbook can be found in the following section.

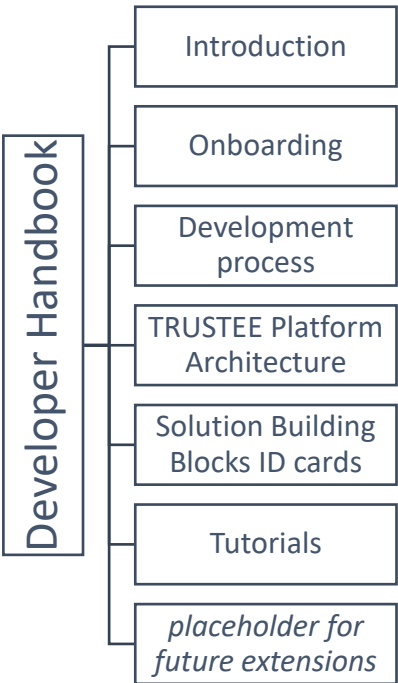


Figure 204: TRUSTEE Developer Handbook

8.4.1.2. SDK EXTERNAL CONNECTORS

IDS CONNECTOR

Figure 205 presents an architecture for realization of the integration cases described in the Section “IDS Connector”, where prerequisites and components involved in realization of the integration case are also explained. As IDS infrastructure VTT’s Data Spaces Innovation Lab (DSIL) infrastructure based on IDSA reference testbed components will be used. Further information can be found in the IDS Connector section.

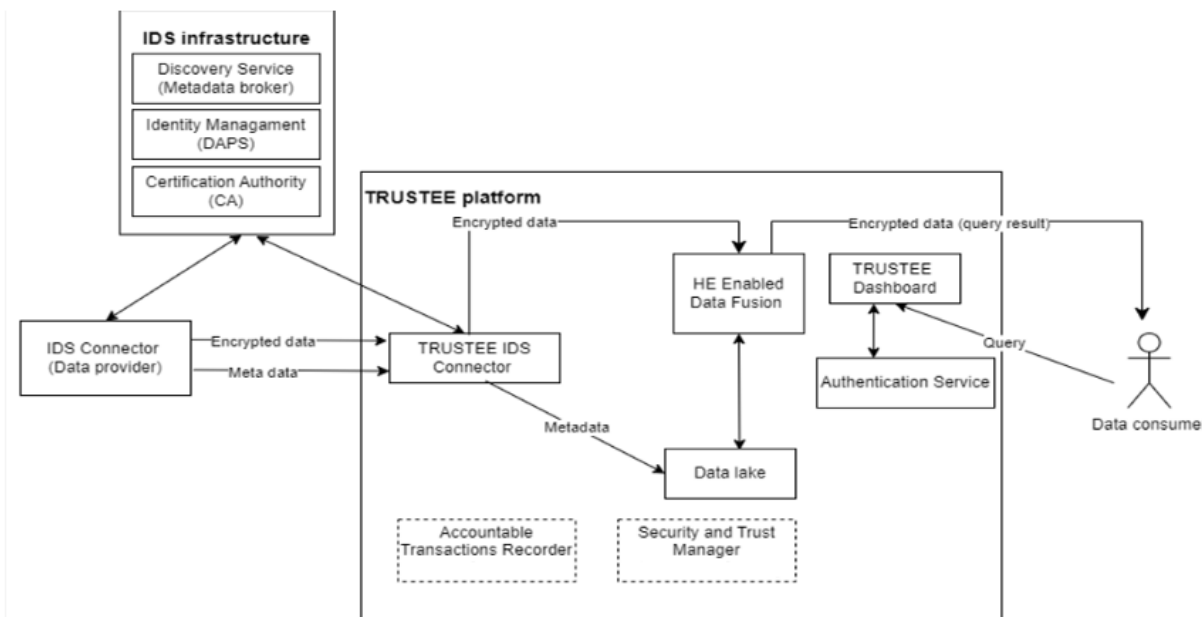


Figure 205: IDS connector in integration with TRUSTEE

The low-level architecture of VTT's IDS connector (i.e., the DSIL Connector) adheres to the general IDS connector architecture specification as outlined in the IDS Reference Architecture Model 3.0 (RAM 3.0)⁵⁷, and is depicted in Figure 205.

The IDS connector serves as the cornerstone of IDS data spaces, functioning as the gateway that connects existing systems and their data to the IDS ecosystem. The connector facilitates data exchange, data enrichment with metadata, and empowers the definition, management, and implementation of usage conditions. These conditions govern how data consumers can utilize the data received from data providers.

⁵⁷ <https://internationaldataspaces.org/wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf>

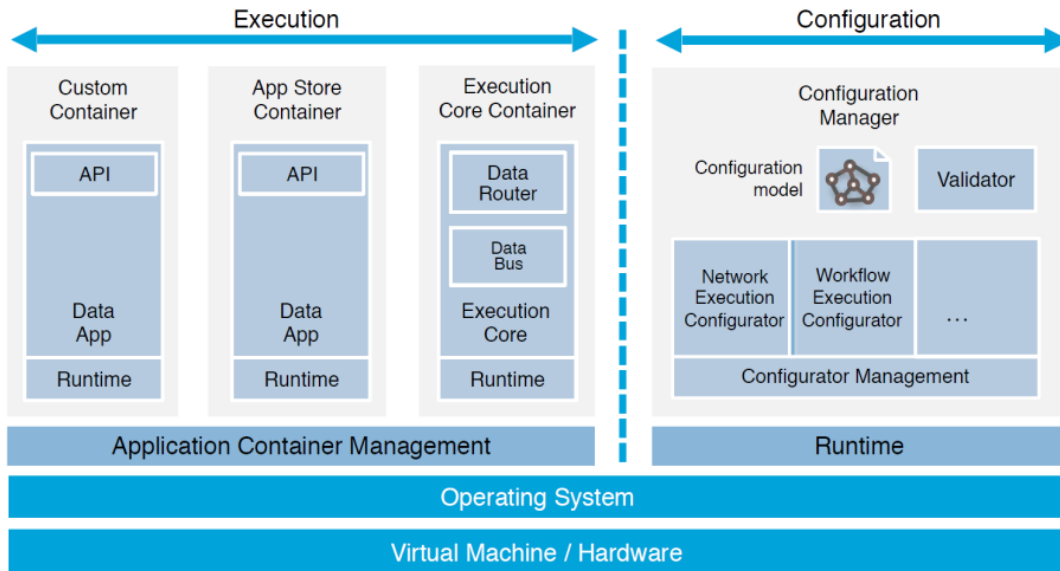


Figure 206: IDS connector architecture

As depicted in Figure 206, the IDS connector architecture comprises several sub-modules and services crucial for the successful deployment of the connector. One of these components is the Execution unit, which consists of the Execution Core Container module. This module specifies the message system, encompassing necessary data buses and routing services. Furthermore, it defines dedicated containers for managing customized, case-specific features and data applications that can be deployed within the connector. Additionally, this module will facilitate the deployment of features and functionalities that are necessary for the seamless integration with the TRUSTEE platform.

The second core component of the architecture is the Configuration unit, which includes the Configuration Manager component. The Configuration Manager is responsible for configuring and managing various settings and parameters of the connector. It enables administrators and users to configure aspects such as network settings, security policies, data sharing rules, and connection profiles. The Application Container Management module plays an important role in deploying, managing, and orchestrating various containers within the connector. Finally, the Operating System provides the fundamental execution environment in which the IDS connector operates. It manages processes, allocates system resources (CPU, memory, disk space), and schedules tasks to ensure the connector functions effectively.

GAIA-X

Figure 207 represents the required architecture to integrate GAIA-X Data Providers in the TRUSTEE platform in order to allow a secure collaboration between GAIA-X and TRUSTEE data spaces. Only GAIA-X Data Providers will be considered to be part of TRUSTEE as external Data Providers.

In order to allow TRUSTEE to interact with GAIA-X, some Verifiable Credentials (VC) from the GAIA-X digital clearing house are needed. First of all, a VC for TRUSTEE (as an entity) is needed. Furthermore, a VC for the SSI-HE service (verifier) is also needed to allow GAIA-X providers to be considered as potential TRUSTEE providers. Finally, a VC for the HEDF service could be also needed to avoid TRUSTEE Data Consumers to need to be part of GAIA-X (it is shown in blue in Figure 207 as it is only needed in some cases). More details can be found in the Implementation section.

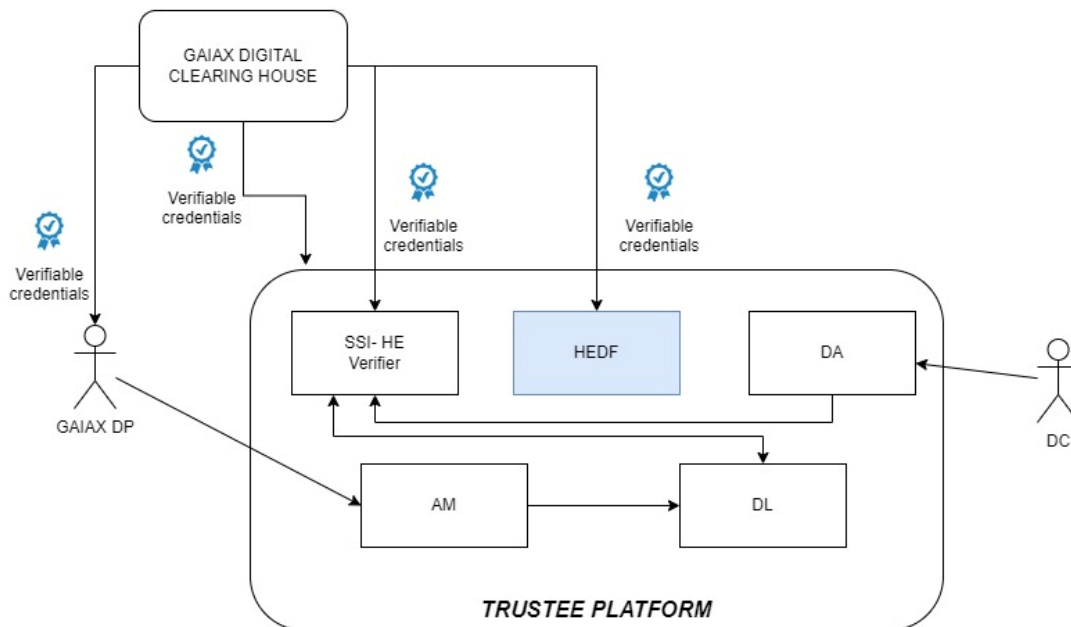


Figure 207: GAIAX integration with TRUSTEE

8.5. IMPLEMENTATION

As previously stated in the Background section, this version of the deliverable presents information on the investigation of several European Data Spaces and external connectors related to them that have the potential to be developed as TRUSTEE SDK external connectors.

Since the OneSS external connectors subcomponent required thorough investigation and their development is expected to progress in the following period of the project, instead of the implementation section structure considered for other TRUSTEE components, this section focuses on the definition of the potential integration cases that have been analyzed and will probably be implemented and described in the following versions of the deliverable, namely D4.2 and D4.3, as the investigation and implementation move forward.

The content related to the implementation status, analyses, and integration cases is structured according to the subcomponents of the OneSS ABB.

8.5.1. DEVELOPER HANDBOOK

The Developer Handbook of OneSS contains documentation in the form of wiki pages, which includes information on the TRUSTEE architecture and building blocks, “getting started” guides that facilitate the onboarding process for users of the TRUSTEE platform, as well as information related to best practices and tutorials. The initially provided content will be expanded and also improved in accordance with the feedback received by users which will be collected during the TRUSTEE pilot campaign designed in D5.1 [68] and evaluated in D5.2.

The TRUSTEE Developer Handbook is published within the project’s GitHub repository (Figure 208 and Figure 209), and it is available at: <https://github.com/Trustee-Horizon/Wiki>. As

previously stated, the contents of the Handbook will be expanded and improved, which will be reported in the following version of this deliverable, namely D4.2 and D4.3.

TRUSTEE Developer Handbook

The structure of the repository is as shown below:

Topic	Description
Intro	Introduction to TRUSTEE Developer Handbook
Onboarding	Documents related to new TRUSTEE developer onboarding
Development Process	Documents related to TRUSTEE software development process
Architecture	Documents related to TRUSTEE software architecture
Solution Building Blocks Design Docs	Documents related to TRUSTEE Solution Building Blocks (concrete implementation of the TRUSTEE Architecture Building Blocks)
Best Practices	Good development practices (methods or techniques) that have been shown to provide improved results
Tutorials	Tutorials

Figure 208: TRUSTEE Developer Handbook contents (status in July 2023)

Tools

The TRUSTEE developer can use make use of the recommended tools

Synthetic data generators, fake APIs	Description
FakerAPI	Faker API it's a collection of completely free APIs that helps web developers and web designers generate fake data in a fast and easy way. No registration is required. No tokens, no authentication. Faker API allows you to generate and download files with fake data to use as you prefer: populating databases, creating API requests for another service
JSON placeholder	Free fake API for testing and prototyping

Online tools related to OpenAPI	Description
OpenAPI auto generators:	Generating OpenAPI specification from your code
OpenAPI Converters:	Various tools to convert to and from OpenAPI and other API description formats.
OpenAPI Data Validators:	Check to see if API requests and responses are lining up with the API description.
OpenAPI Description Validators:	Check your API description to see if it is valid OpenAPI.
OpenAPI Documentation:	Render API Description as HTML (or maybe a PDF) so slightly less technical people can figure out how to work with the API.
OpenAPI DSL:	Writing YAML by hand is no fun, and maybe you don't want a GUI, so use a Domain Specific Language to write OpenAPI in your language of choice.
OpenAPI Gateways:	API Gateways and related tools that have integrated support for OpenAPI.
OpenAPI GUI Editors:	Visual editors help you design APIs without needing to memorize the entire OpenAPI specification.

Figure 209: An example of development-related content available within the handbook

8.5.2. SDK EXTERNAL CONNECTORS

Another part of the OneSS is the OneSS component related to SDK external connectors. The external connectors segment of OneSS focuses on establishing a connection with external systems. As a component that encapsulates access to an external system or resource, it can also be considered a gateway to an external system. As such, the SDK external connectors of the OneSS serve to provide the functionality of accessing external data sources. In this section we provide more information on the external connectors towards external data spaces that were explored within the first year of the project. These are: IDS, GAIA-X, EOSC, Copernicus Data Space Ecosystem, and European Grid Infrastructure (EGI).

Due to the different levels of maturity of the considered data spaces, some of them can only be presented in terms of their functionalities and also possibilities regarding the integration with the TRUSTEE platform. For more mature ones, how they could potentially be integrated with TRUSTEE is also described.

8.5.2.1. IDS CONNECTOR

INTRODUCTION

The IDS connector is defined by IDSA as “the central technical component for secure and trusted data exchange.”⁵⁸ Dataspace connector can be utilized to extend existing software by the functionalities of IDS connector and integrate it into the IDS data ecosystem. Furthermore, the dataspace connector can be utilized to develop its software which can be connected to the IDS data ecosystem.

IDS connector is a fundamental element of the IDS architecture, designed to facilitate secure and controlled data exchange in a digital ecosystem. It acts as a gateway which connects different participants and entities within the IDS network, allowing them to access and share the data securely.

The functionalities and architecture of the IDS connector follow an IDS RAM and are specified by a certification criterion. The dataspace connector uses application container management technology (docker container) to provide a secure and isolated environment for individual data services. The dataspace connector supports a segmented network. Every running container has an associated network zone by providing its own virtual network stack. The connector, being the core container, contains root rights which makes it capable of managing security system and network configurations, for all the networks and separated containers.

The IDS connector ensures the protection of data privacy, sovereignty, and integrity by providing encryption, access control mechanisms and data usage policies. It enables authorized users to discover, negotiate, and establish trusted connections with other participants, facilitating the exchange of data in a peer-to-peer manner.

PRE-REQUISITES

VTT DSIL (Dataspace Innovation Lab) Overview

VTT will deploy in the TRUSTEE project the DSIL infrastructure that is based on IDSA reference testbed components. The DSIL can support the development, utilisation and testing of data spaces and different data space concepts and it facilitates preserving data sovereignty in confidential data sharing between organisations. Additionally, the DSIL provides the experimental data space as a service for organizations interested in data economy and data sharing. Currently, two main use cases are offered:

- The DSIL (or one of its components) can be used as a baseline for developing customized data space solutions for specific application domains or use cases. It allows e.g. the testing of business concepts and building of proof of concepts before investing in real-life solutions
- DSIL enables performing compatibility tests for existing IDS connectors. The testing procedure consists of verifying that the connector under test can successfully communicate and share data with the components hosted by the DSIL.

At present, in addition to example IDS connectors, the DSIL contains the following IDS infrastructure services:

- **Identity provider**— consists of three components, namely the Certificate Authority (CA), the Dynamic Attributes Provisioning Service (DAPS), and the Participant Information Service (ParIS). The role of the CA is to grant X.509 certificates, i.e., machine-readable certificates that safeguard against man-in-the-middle attacks. DAPS enables connectors to authenticate themselves using those X.509 certificates received from the CA. After successful authentication, the DAPS issues OAuth2 (“Open Authorization”, an open standard for

⁵⁸ <https://internationaldataspaces.org/offers/ids-components/>

access delegation) access tokens for connectors, who need these tokens to access the services and data of other connectors on the user's behalf.

- **Metadata Broker**—stores information about the data endpoints offered by participants of the data space. It does not store the data sets itself, but only the meta information and is thus referred to as the metadata broker. The broker provides a query interface for a connector and can hence be considered the search engine for data space information. This means that data providers can register their data sets at the broker and, subsequently, data consumers can search for data they need for their use for case-specific purposes. In TRUSTEE the broker service can be utilized as an additional metadata repository enabling e.g., the TRUSTEE compatible IDS connector to discover data resources that other IDS connectors can offer for TRUSTEE platform users.
- **Certificate Authority** - is responsible for issuing, validating, and revoking digital certificates. According to the IDS RAM specification, a digital certificate can be provided for a participant if both a valid certification for the operational environment and a valid certification for the core component is available. However, in the TRUSTEE project, the DSIL enables generating test certificates for different participants for experimental use.

In TRUSTEE, the DSIL will be applied for providing the mandatory infrastructure services that are needed for establishing IDS-based data exchange connections between the TRUSTEE-compatible IDS connector and other IDS connectors (acting either as data providers, data consumers or both).

INVOLVED COMPONENTS

VTT Dataspace Innovation Lab Connector (DSILC)

The VTT DSILC is based on the IDSA dataspace connector RAM which aims to provide an easy way to install and test IDS connector. VTT DSILC has some additional features which are added as extension points to the original IDSA RAM. The two extension points provide improved identity/access management with Keycloak integration and support for different data types by integrating Open Platform Communication Unified Architecture (OPC-UA) integration.

In the Dataspace connector by IDSA, the access management can only be done at a connector level. This means data providers can define other IDS connector instances that can access different data resources they offer. However, based on our experiences from our on-going projects there are situations where more fine-grained access management is needed. In these use cases, data providers must be able to define individual persons (i.e., specific users of data consumer connectors) entitled to access offered data resources.

The first extension point is restricting the access/usage of a data resource based on (i) the identity of a user (user authentication) that is currently logged in to the consumer side IDS connector, (ii) the role of the user and (iii) the generated access token. The user logs in and is given the access token and makes a request.

The second extension point extends the reference implementation and provides support for OPC-UA data. With OPC, access to devices, machines and other systems is standardized for industrial environments and manufacturer-independent data exchange is enabled. OPC UA stands for Open Platform Communications Unified Architecture, where UA is the latest specification of the standard. OPC with UA extension differs from its predecessor as it is platform independent and provides support for semantic data description.

IDENTIFIED INTEGRATION CASES

This section defines the integration of IDS connectors with the TRUSTEE platform, and the goal is to ensure interoperability between the TRUSTEE platform and data spaces. By the moment, three integration cases have been identified to ensure communication between IDS connectors and the TRUSTEE platform:

- i. Data can be provided from the IDS connectors to the TRUSTEE platform users.
- ii. TRUSTEE platform extends IDS-based data exchange capabilities.
- iii. IDS connectors can retrieve data from the TRUSTEE platform.

The overall view of how the first integration cases can be implemented is shown in Figure 210.

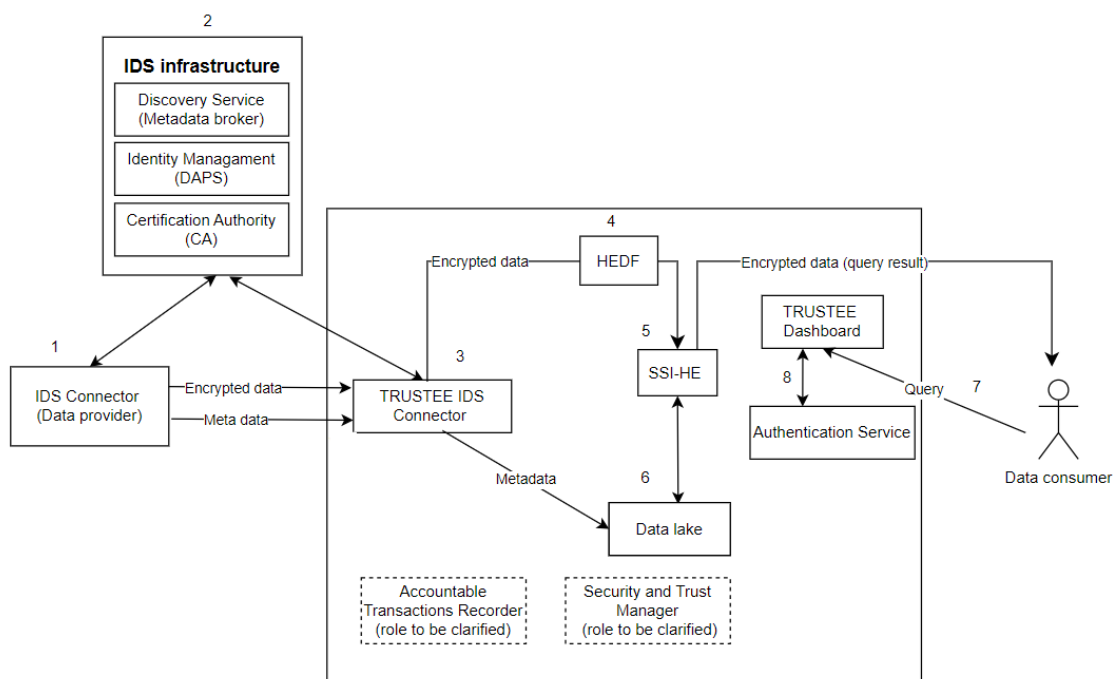


Figure 210: IDS Connector integration cases – data provided from the IDS Connector to the TRUSTEE users

ANALYSIS

Although three potential scenarios have been identified, by the moment, the focus is on the first integration case, where the IDS connectors operating data spaces would act as data providers for TRUSTEE and TRUSTEE platform users would be the data consumers.

The flow in this case is as follows:

- i. Encrypted data is provided from IDS connectors to the TRUSTEE platform users, where the TRUSTEE platform has its own IDS connector. The data is shared between the two IDS connectors where the connector outside the TRUSTEE platform acts as a provider IDS connector and provides the information on metadata to the TRUSTEE IDS connector.
- ii. The data exchange between the two IDS connectors uses the services provided by the IDS infrastructure (Metadata Broker, DAPS and CA).
- iii. HEDF ABB receives encrypted data from IDS connectors and executes data fusion.
- iv. SSI-HE ABB then provides metadata attributes for visualization in the DA.
- v. The DL then stores and manages the metadata (indexes) describing the data resources offered by IDS connectors.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- vi. Data consumer from outside the TRUSTEE platform can authenticate themselves by the AM and query for the available data sources from the DA.
- vii. The DA will be a cloud-based, web application that allows the user to fully interact with and utilize the TRUSTEE Platform.

The main functionalities that the DA will provide to the users of the platform are:

- Registration and login functionality by communicating with the AM ABB.
- Search operations as well as computational operations over the encrypted data of data providers by communicating with HEDF and SSI-HE ABBs.
- Computational operations over requested/specified data.
- The ability to select and use of ML/AI algorithms by communicating with the AIMaaS ABB.
- Visualization of the results produced by the end-user's requested computations.
- Navigation assistance and workflow automations through the VA and RPA inner building blocks.

Functionalities which are yet to be tested:

- STM: test and validate the data sovereignty (e.g., contract management) features provided by the STM module and compare them to the data sovereignty framework provided by the IDS reference architecture.
- ATR: provides information about the events associated to the confidential metadata sharing (logging reasons).

This integration case helps to engage additional data providers from IDS by extending the opportunity for data providers to take benefit from TRUSTEE platform capabilities. Moreover, TRUSTEE users can also benefit from the data hosted on IDS connectors. The proposed architectural model shows that TRUSTEE would have its own IDS connector to make data offering easy for the data consumers by enabling point to point communication. The TRUSTEE IDS connector would also have its own certificate which it will use for authentication.

8.5.2.2. GAIA-X

One of the clearer relations between TRUSTEE and GAIA-X is considering GAIA-X participants as additional data providers for TRUSTEE, extending the use of their data also to TRUSTEE data consumers/processors.

Taking this premise into account, two different integration cases have been considered for connecting GAIA-X data providers to TRUSTEE:

- Integration case 1, in which the TRUSTEE platform takes part in GAIA-X and acts as a data consumer inside GAIA-X and resends data to TRUSTEE data consumers/providers.
- Integration case 2, in which, TRUSTEE data consumers/processors directly take part in GAIA-X as data consumers.

INTEGRATION CASE 1 – TRUSTEE PLATFORM IN GAIA-X

The operation for considering a GAIA-X data provider as a TRUSTEE data provider establishing a communication directly between the GAIA-X data provider and TRUSTEE platform as data consumer should follow a similar process to that of native TRUSTEE data providers.

Prerequisites

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- PRE1. Using the GAIA-X Clearing house, the TRUSTEE Platform should obtain the verifiable credentials to become GAIA-X compliant.
- PRE2. Using the GAIA-X Clearing house, the TRUSTEE Platform should obtain the verifiable credentials for the two main services needed for receiving data from GAIA-X data providers. These two services should indicate all the requirements for being compliant with the TRUSTEE platform.
- CERTIFIED SSI-HE-VERIFIER consumer (metadata). This component will receive and validate SSI proofs with encrypted attributes in GAIA-X.
 - CERTIFIED HEDF consumer (data). This component will indicate the public key and the encryption tool needed for the HEDF operation and will receive the encrypted data.
- PRE3. GAIA-X data providers interacting with TRUSTEE must expose two services fulfilling TRUSTEE requirements (for both metadata and data provision)

These prerequisites are supposed to be done in advance. GAIA-X is still under construction, so, these steps are only analyzed from the conceptual point of view at the moment.

Involved components

TRUSTEE components in this integration case with GAIA-X are presented in the Figure 211.

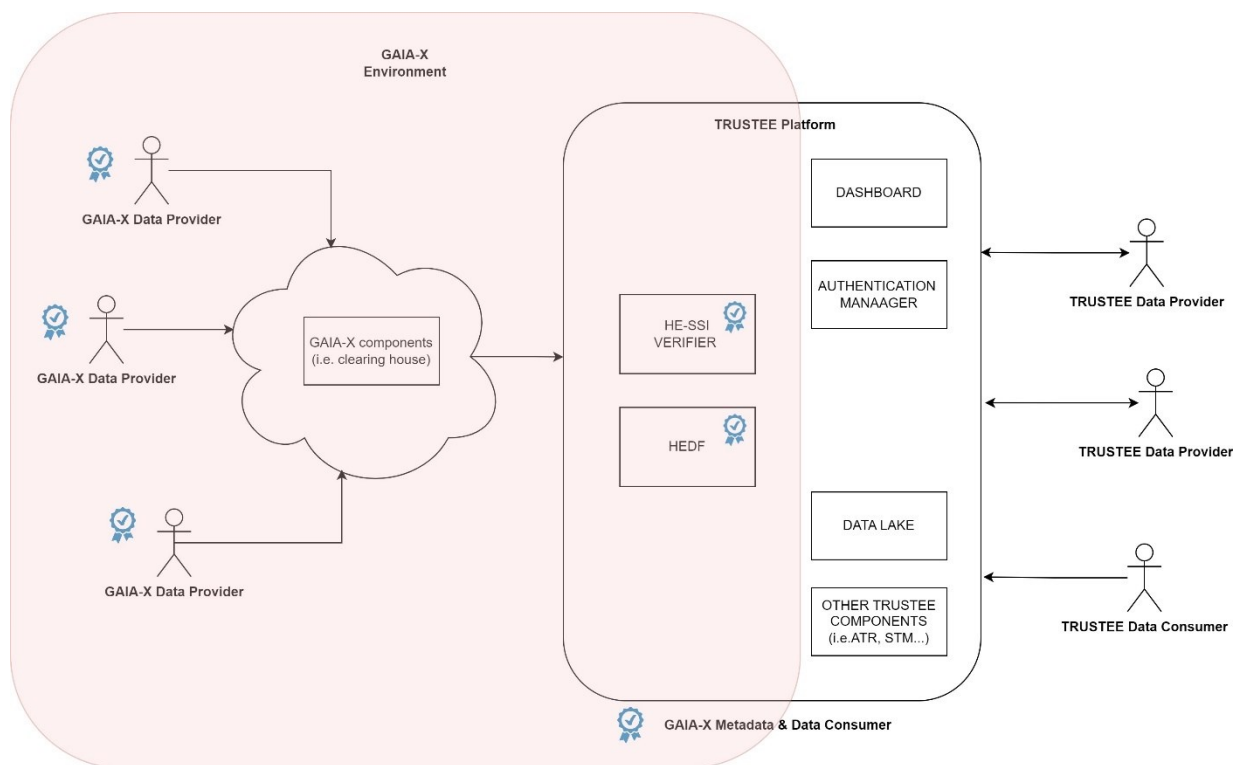


Figure 211: TRUSTEE components involved in the GAIA-X Integration Case 1

Sequence steps

The sequence steps for the first integration case are shown in Figure 212.

The minimum preliminary steps as part of a “registration” process in TRUSTEE are:

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- i. GAIA-X data provider should “register” in TRUSTEE. This registration should happen in the AM component.
- ii. Once registered, the GAIA-X data provider should provide his HE public keys and metadata and data endpoints in the DL ABB.
- iii. GAIA-X data provider should follow the offline steps of the SSI-HE operation as usual in order to have SSI credentials with encrypted attributes to characterize his data set in a private way.

Once the GAIA-X data provider has executed the previous steps, he is also available in TRUSTEE to be considered as a data provider. The process in this case is as follows:

- iv. A TRUSTEE data consumer submits a query for a specific operation over data with specific attributes/features following the traditional TRUSTEE process (DA and AM components are used).
- v. CERTIFIED SSI-HE-VERIFIER receives the attribute to be searched.
- vi. CERTIFIED SSI-HE-VERIFIER obtains the GAIA-X data provider metadata endpoint from the DL.
- vii. SSI-HE flow is executed as usual considering the GAIA-X data provider metadata endpoint. As a result, metadata validation is obtained.
- viii. If positive, the GAIA-X data provider is considered useful for the specific query from the TRUSTEE data consumer. For this reason, the GAIA-X data provider data endpoint is obtained from the DL and shared with the CERTIFIED HEDF component.

Once the GAIA-X data provider is considered suitable for the specific query, the data fusion process can start, applying the specific operation over GAIA-X data provider data (and any other data provider's data that have also fulfilled the query requirements). However, for GAIA-X data providers, in this case, the process is slightly different to the traditional way of operation of the HEDF component, as data needs to be first received in the TRUSTEE platform (as it is the GAIA-X data consumer) and resent to the TRUSTEE data consumer afterwards for the operation execution. TRUSTEE will be like a proxy between GAIA-X data providers and TRUSTEE consumers. For this purpose:

- ix. CERTIFIED HEDF certified requests the GAIA-X data provider's data from the data endpoint. This request will include the HE public key needed for encrypting the data, indicating also the tool that should be used for encryption to be compatible with TRUSTEE.
- x. GAIA-X data provider applies the corresponding encryption process over his data.
- xi. CERTIFIED HEDF obtains the HE-encrypted data and propagates it to the TRUSTEE data consumer where the operation will take place.
- xii. The rest of the HEDF process is as usual.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

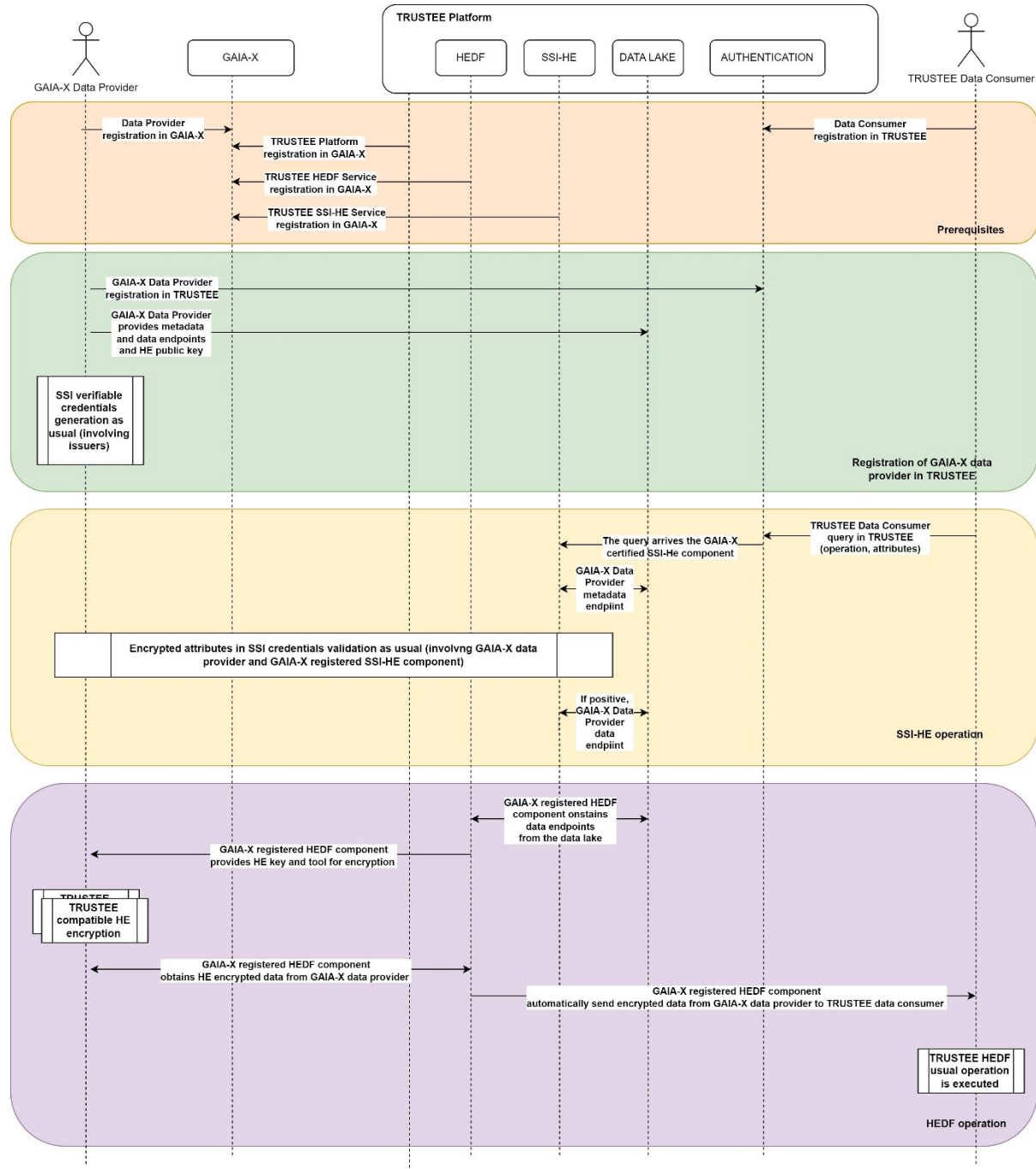


Figure 212: Sequence diagram - GAIA-X Integration Case 1

Analysis

This integration case is the easiest one to be implemented, as just TRUSTEE (as a platform) needs to be part of GAIA-X. In fact, this is the main advantage as TRUSTEE data consumers do not need to be part of GAIA-X to consider GAIA-X data provider's data on their executions. However, the challenge is that it is TRUSTEE who needs to obtain

the HE-encrypted data from GAIA-X data providers and resend it to the TRUSTEE data consumers. The easiest way to resend data is just to act as a proxy, without storing it locally.

INTEGRATION CASE 2 – TRUSTEE DATA CONSUMERS IN GAIA-X

The operation for considering a GAIA-X data provider as a TRUSTEE data provider establishing a communication directly between GAIA-X data provider and TRUSTEE data consumers (avoiding using TRUSTEE as a proxy) should follow a similar process to that explained in integration case 1 but with some differences indicated in bold in the following sections:

Prerequisites

- PRE1. Using the GAIA-X Clearing house, the TRUSTEE Platform should obtain the verifiable credentials to become GAIA-X compliant.
- PRE2. Using the GAIA-X Clearing house, the TRUSTEE Platform should obtain the verifiable credentials for the SSI-HE verifier service needed for receiving metadata from GAIA-X data providers. This service should indicate all the requirements for being compliant with the TRUSTEE platform.
 - CERTIFIED SSI-HE-VERIFIER consumer (metadata). This component will receive and validate SSI proofs with encrypted attributes in GAIA-X.
- PRE3. Using the GAIA-X Clearing house, TRUSTEE data consumers (who want to consider GAIA-X data providers) should obtain the verifiable credentials for the HEDF service needed for receiving data directly from GAIA-X data providers. This service should indicate all the requirements for being compliant with the TRUSTEE platform.
 - CERTIFIED HEDF consumer (data). This component will indicate the public key and the encryption tool needed for the HEDF operation and will receive the encrypted data.
- PRE4. GAIA-X data providers interacting with TRUSTEE must expose two services fulfilling TRUSTEE requirements (for both metadata and data provision)

These prerequisites are supposed to be done in advance. GAIA-X is still under construction, so, these steps are only analyzed from the conceptual point of view by the moment.

Involved components

TRUSTEE components in this integration case with GAIA-X are presented in the Figure 213.

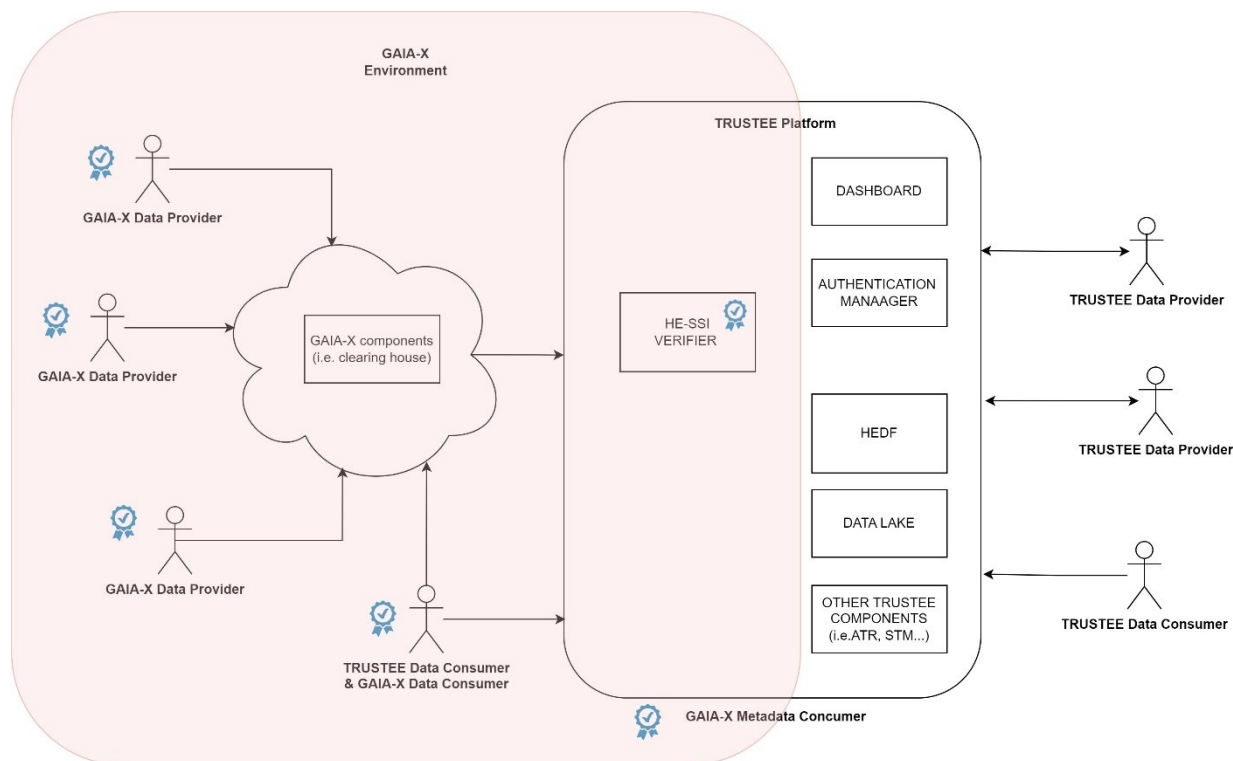


Figure 213: TRUSTEE components involved in the GAIA-X Integration Case 2

Sequence steps

The sequence steps for the second integration case are shown in Figure 214.

The minimum preliminary steps as part of a “registration” process in TRUSTEE are:

- i. GAIA-X data provider should “register” in TRUSTEE. This registration should happen in the AM ABB.
- ii. Once registered, the GAIA-X data provider should provide his HE public keys and metadata and data endpoints in the DL.
- iii. GAIA-X data provider should follow the offline steps of the SSI-HE operation as usual in order to have SSI credentials with encrypted attributes to characterize his data set in a private way.

Once the GAIA-X data provider has executed the previous steps, he is also available in TRUSTEE to be considered as a data provider. The process in this case is as follows:

- iv. A TRUSTEE data consumer submits a query for a specific operation over data with specific attributes/features following the traditional TRUSTEE process (DA and AM components are used).
- v. CERTIFIED SSI-HE-VERIFIER receives the attribute to be searched.
- vi. CERTIFIED SSI-HE-VERIFIER obtains the GAIA-X data provider metadata endpoint from the DL.
- vii. SSI-HE flow is executed as usual considering the GAIA-X data provider metadata endpoint. As a result, the metadata validation is obtained.
- viii. If positive, the GAIA-X data provider is considered useful for the specific query from the TRUSTEE data consumer. For this reason, the GAIA-X data provider data endpoint is obtained from the DL and shared with the CERTIFIED HEDF component.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

Once the GAIA-X data provider is considered suitable for the specific query, the data fusion process can start, applying the specific operation over GAIA-X data provider data (and any other data provider's data that have also fulfilled the query requirements). In the usual HEDF operation, data is directly shared between TRUSTEE data providers and TRUSTEE data consumer; in this case, the process can be equally applied as TRUSTEE data consumers are also GAIA-X data consumers and they are able to receive data from GAIA-X data providers. For this reason, the process is as the usual one inside TRUSTEE:

- ix. CERTIFIED HEDF certified requests the GAIA-X data provider's data from the data endpoint. This request will include the HE public key needed for encrypting the data, also indicating the tool that should be used for encryption to be compatible with TRUSTEE.
- x. GAIA-X data provider applies the corresponding encryption process over his data.
- xi. TRUSTEE data consumer (also GAIA-X data consumer) obtains the HE encrypted data.
- xii. The rest of the HEDF process is as usual.

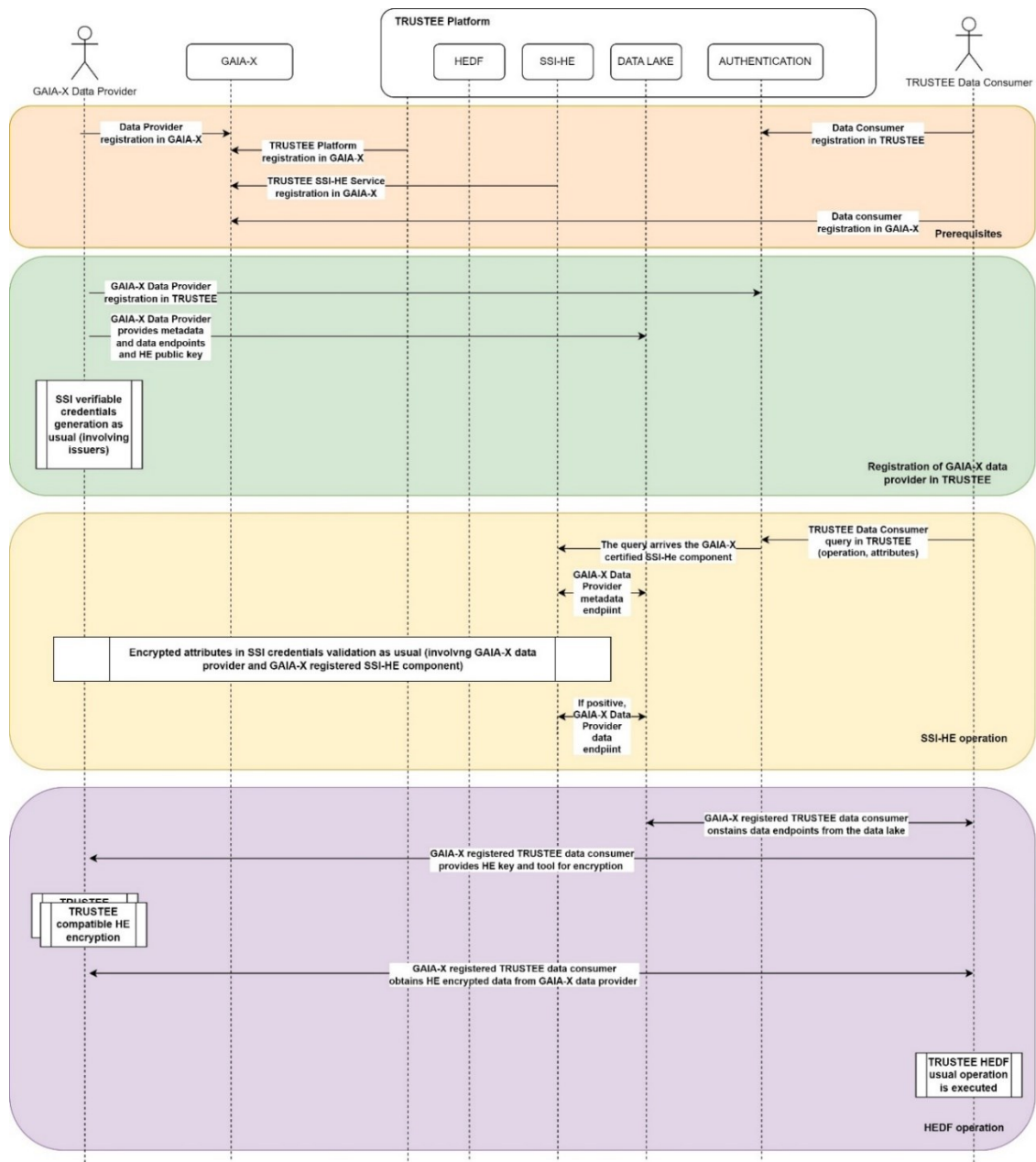


Figure 214: Sequence diagram - GAIA-X Integration Case 2

Analysis

The main advantage of this integration case is that the HEDF operation with GAIA-X data providers remains as usual with TRUSTEE data providers. However, for this purpose, TRUSTEE data consumers who want to consider GAIA-X data providers data on their executions need to be part of GAIA-X. This could be an option for some TRUSTEE data consumers, but not for others, limiting the applicability of TRUSTEE for interacting with GAIA-X.

8.5.2.3. EUROPEAN OPEN SCIENCE CLOUD

EOSC [59] is an initiative by the EU that aims to create an inclusive and collaborative environment for researchers and scientists to access and share research data, services, and tools across disciplines and borders. Its primary goal is to promote open science principles and enable seamless access to a wide range of resources, fostering scientific discovery and innovation. Initiated in 2016, EOSC seeks to address the challenges posed by increasing data volumes and diverse sources by establishing a federated infrastructure. This infrastructure integrates existing and emerging data repositories, services, and infrastructures from various domains, providing researchers with a unified entry point to a diverse pool of data and services.

One of EOSC's main aims is adhering to the principles of open science, emphasizing the importance of making research data FAIR. By promoting open and FAIR data practices and being an environment for hosting and processing research data, EOSC's goal is to foster collaboration among researchers from different scientific fields. A summary of EOSC's services from the user perspective can be seen in Figure 215.

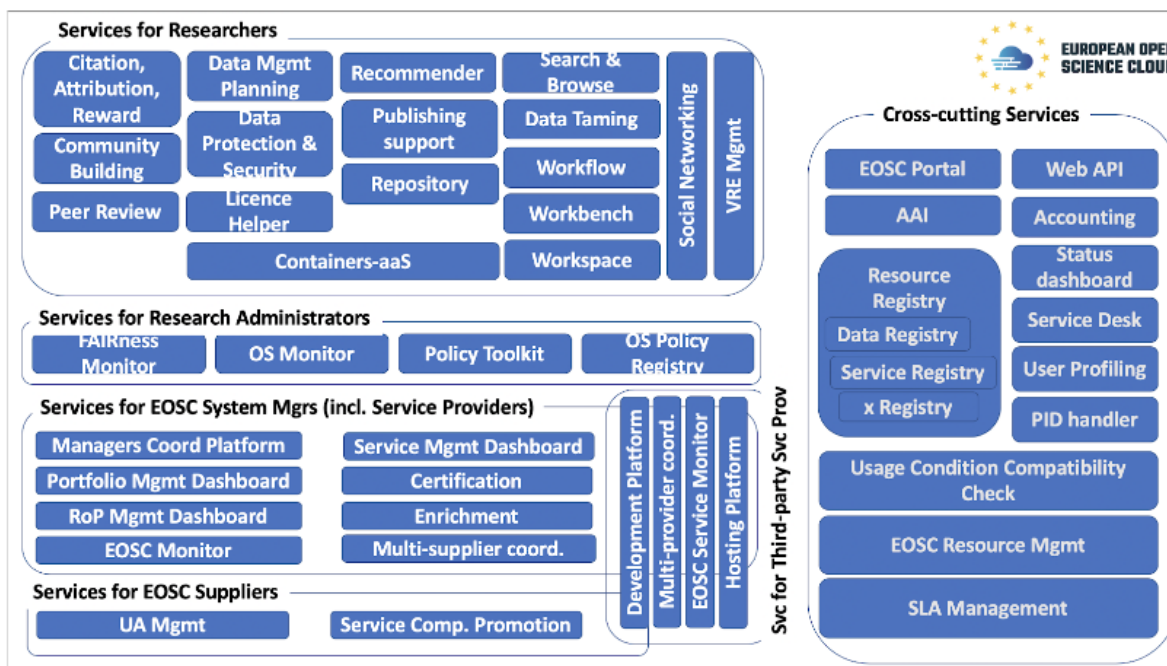


Figure 215: EOSC Services [87]

EOSC's implementation involves collaboration between research communities, infrastructures, service providers, and policymakers. EOSC projects receive funding and support from the European Commission to develop specific components of the EOSC ecosystem. The implementation occurs in phases, with a focus on building a strong foundation, integrating existing infrastructures, and addressing technical and policy challenges.

A possible integration case for TRUSTEE includes connecting to the EOSC catalog and data source via available API⁵⁹, which is going to be further explored in the following months of the project with a conclusion reported in the next version of the deliverable as more details regarding the possible integration will be gathered.

8.5.2.4. COPERNICUS DATA SPACE ECOSYSTEM

The Copernicus Data Space Ecosystem [88] is designed to facilitate access and utilization of Earth observation data, aiming to unlock its full potential for various stakeholders. By providing immediate access to a wide range of open and free Earth observation data from Copernicus Sentinel satellites and other sources, the platform offers scalable interfaces and opportunities to build applications and data products. These developments are expected to contribute to a data economy and promote sustainable practices. Additionally, Copernicus is a reputable and widely used platform and the EU initiative, and hence it was considered as one of the data spaces for integration with TRUSTEE.

One of the primary features of the Ecosystem is its vast collection of Earth observation data, including historical and real-time Sentinel images, commercial datasets, and Copernicus Contributing Missions' data. By consolidating these diverse sources, the platform enables users to explore opportunities across various industries and fields, addressing challenges related to climate change, disaster monitoring, urban planning, and agriculture optimization, among others.

Copernicus Data Space Ecosystem was launched in January 2023, with the goal of expanding the services previously accessed by users through the Copernicus Open Access Hub [89], and it is planned to be fully functional by November 2023, according to the Copernicus Data Space Ecosystem roadmap [90] (Figure 216). Due to the (un)availability of functionalities and APIs, we have explored Copernicus Data Space Ecosystem in the context of documentation and services available at the time of producing this deliverable. As the majority of services, data, and APIs will be available after July 2023, we will report on the practical aspects of integration with TRUSTEE in the following version of this deliverable, namely D4.2. Currently, we focus on the analysis of available APIs that would enable the integration and eventually, developing the TRUSTEE integration case.

⁵⁹ <https://providers.eosc-portal.eu/openapi>

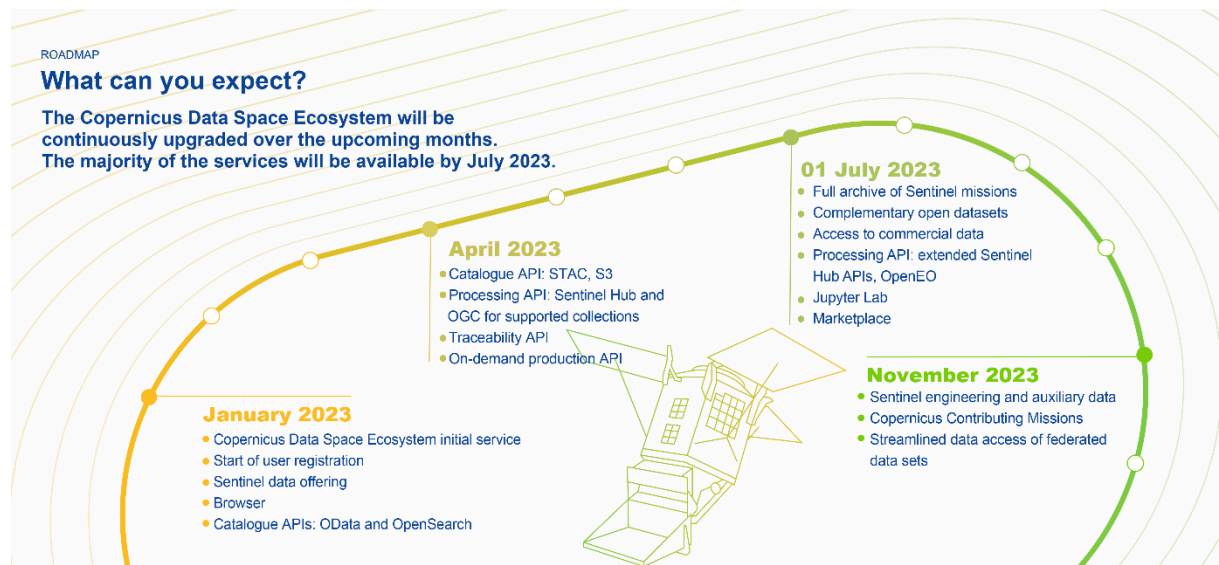


Figure 216: Copernicus Data Space Ecosystem roadmap⁶⁰

The Ecosystem emphasizes data privacy, governance, and intellectual property preservation for the data space participants. This commitment helps build trust among stakeholders and fosters a sense of confidence in sharing and collaborating with others, which is also very much aligned with TRUSTEE principles. The platform also enables the deployment and operation of third-party services, broadening the range of value-added services available to users. As an additional benefit, participants of the Data Space Ecosystem can gain exposure to a diverse user community, increasing the visibility of their services.

For data exploration and analysis, the Ecosystem offers tools for easy discovery, visualization, and download of Earth observation data. One tool that facilitates data exploration is Copernicus Browser which enables users to engage with satellite imagery intuitively. The Ecosystem's data analytics environment provides a set of data tools and applications, such as Jupyter Notebook and Data Workspace enabling users to generate insights from the available data in a user-friendly and straight-forward way, using visual interfaces and streamlining the data processing workflows. The Copernicus Data Space Ecosystem also plans to extend its range of APIs to cover various aspects such as cataloguing, product download, visualization, and processing services.

Currently (July 2023), the Copernicus Data Space Ecosystem offers documentation [91] on the available APIs, which include:

- **Catalogue APIs**
 - Spatio-Temporal Asset Catalog (STAC) product catalog.
 - OData - standard based on HTTP RESTful Application Programming Interfaces which enables the creation and editing of resources via HTTP messages.
 - OpenSearch - enables searching through Copernicus data using a standardized web service.
- **Streamlined data access**
 - Sentinel Hub API - used to retrieve satellite data from the archives.

⁶⁰ <https://dataspace.copernicus.eu/>

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- openEO API - used for sharing code, workflows, and data processing methods across platforms and tools, it also provides programming libraries that enable easy analysis of diverse datasets.
- **Traceability API** - allows user to verify and register traces for user-level data available in the Copernicus Data Space.
- **On-demand Production API** - provides functionalities like creation, update, cancellation, pausing and monitoring of orders, allowing users better control over the workflow execution.
- **S3 API** - one of the main access methods for Earth Observation data, suitable for Third Party applications requiring high performance parallel access and scalability.

The Ecosystem's value proposition will further expand with time, as a range of APIs become available. The APIs cover various aspects, including cataloguing, product download, visualization, and processing services. The platform's gradual extension ensures that stakeholders have access to the relevant tools, thus continuously promoting their ability to leverage available data effectively.

The data available on the platform consist of Copernicus Sentinel Mission data (including Sentinel 1, Sentinel 2, Sentinel 3, and Sentinel 5P missions), containing Earth Observation data documented on the platform website [92]. Other than Sentinel imaging missions, complementary data is available related to the Copernicus Atmosphere Monitoring Service, Copernicus Emergency Management Service, Copernicus Land Monitoring Service, Copernicus Marine Service, Landsat program data, as well as part of the planned very high resolution commercial data.

To register and authenticate on the platform, users have to create a user account via the Copernicus Data Space Ecosystem login on the platform's home page (Figure 217). Also, to download products from Copernicus Data Space Ecosystem using OData and OpenSearch API, users are required to have a Keylock token. The procedure of generating tokens by CURL query or by Python script is available in documentation [93] for both Linux and Windows operating systems.

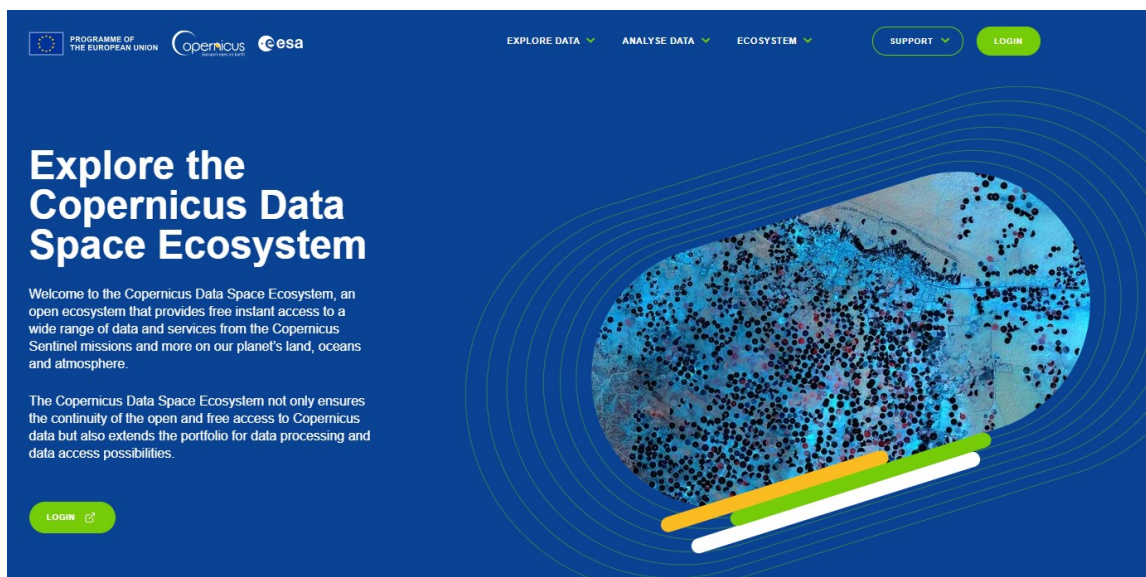


Figure 217: Copernicus Data Space Economy home page

As the platform continues to develop and expand its functionalities, applications, tools, and APIs it is offering, the Copernicus Data Space Ecosystem aims to be a valuable resource for Earth observation data access and utilization. Since most of the platform features are available from July 2023, most of our integration efforts will be reported in the

next version of this deliverable, however, the initial analysis was performed and the most relevant information on the available APIs and plans for future Copernicus was gathered. Based on the performed analysis, we currently envision the integration case in which TRUSTEE could act as a Copernicus Data Space Ecosystem user which will be reported in detail in the next version of this document.

8.5.2.5. EUROPEAN GRID INFRASTRUCTURE

EGI [94] is an international infrastructure cooperation providing solutions that are related to computational services, data storage, and data manipulation. EGI was proposed in 2002 with the demonstration of the first large scale experimental facility for distributed computing, which was presented by the DataGrid [95] project.

There are two entity types in EGI, namely the EGI user and the EGI provider, encompassing different opportunities and options.

The services where an EGI provider can contribute to or benefit from are as follows:

- Act as a federated resource center
- Act as a software provider
- Act as a new provider for an existing service
- Act as a core/central service provider
- Act as a new service provider.

Additionally, the EGI infrastructure provides EGI users with various services such as:

- Computing
- Storage, and data services
- Security and authentication
- Training
- Development environments

In order to access the EGI infrastructure one needs to use the EGI Check-In option. After the authentication process, the user takes access to storage, computing, and development services. There are three available computing categories, in addition to orchestration tools and workload managers, as follows:

- Cloud Compute, which is based on virtual machines.
- Container Compute, which supports running applications with Docker or Kubernetes.
- High Throughput Compute, which is based on shared grid computing nodes.

Regarding storage services, Online Storage includes three major subtypes, which allow users to store, share, and access data. The user can access data through various protocols and replicate them across several storage locations. Data Transfer is another storage service that provides the user with the ability to move data asynchronously from one storage to another. Data Transfer alongside Data Orchestrator makes data manipulation simple, secure, and lossless. The third storage-related service, DataHub, which is based on the Onedata [96] technology, enables simple and scalable access to distributed data for computing and publishing, as well the availability of datasets across federated sites in specific communities or worldwide. All the available data can be accessed through the Onedata web interface or by using the Onedata APIs. In order to be able to access the API, an access token is required, which can be created and managed using the EGI DataHub web interface. Furthermore, there are three different levels of access to the data that comply with applicable regulations, namely:

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- unauthenticated open access
- access after registration, and
- access restricted to members of a Virtual Organization (VO).

Some additional features of the DataHub service include data and metadata sharing management, data import, and data caching based on the file's popularity.

POTENTIAL INTEGRATION CASES

TRUSTEE as an EGI user

TRUSTEE platform could use the EGI infrastructure as an EGI user, in order to retrieve available data available through the DataHub service, by utilizing the provided APIs and an access token, and fuse them with data offered by TRUSTEE Data Providers and registered as part of the Distributed Data Source of TRUSTEE. After the EGI Check-In procedure, the access token is generated using the EGI DataHub web interface, as showcased in Figure 218 and Figure 219.



Figure 218: EGI Check-in Page

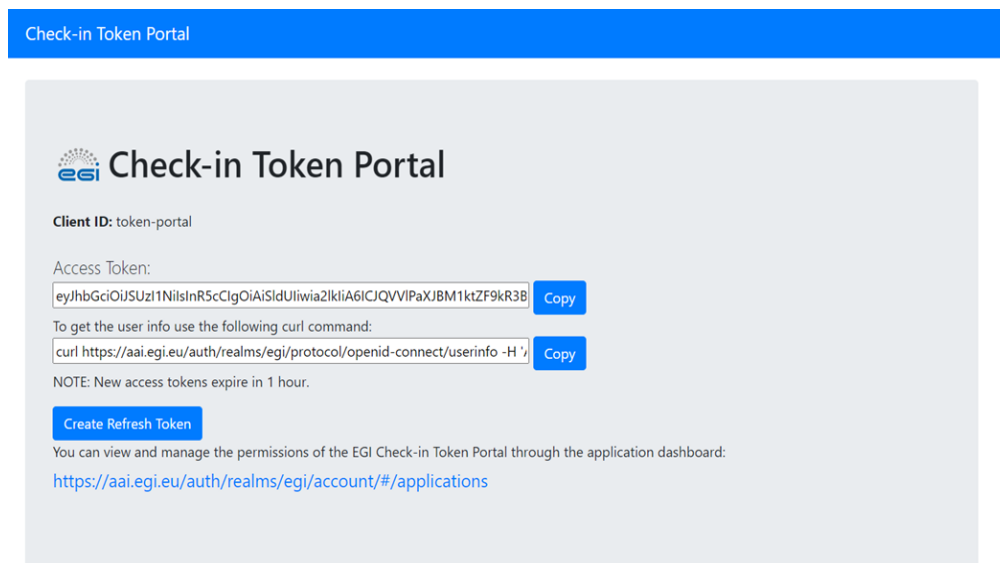


Figure 219: EGI Check-in Token Generation

Following, the generated and provided token can be utilized in the user's API requests, as it is depicted in Figure 220 below, for accessing, downloading, and uploading data to OneData.

```
# Get base folder ID
$ curl -H "X-Auth-Token: $DATAHUB_TOKEN" \
  -X POST "https://$ONEPROVIDER_HOST/api/v3/oneprovider/lookup-file-id/PLAYGROUND"

# Add the folder ID to an environment variable
$ export DIR_ID=<ID_FROM_PREVIOUS_COMMAND>

# List files inside the folder with DIR_ID
$ curl -H "X-Auth-Token: $DATAHUB_TOKEN" \
  -X GET "https://$ONEPROVIDER_HOST/api/v3/oneprovider/data/$DIR_ID/children" \
  | jq .

# Add the ID of the file that you want to download
$ export FILE_ID=<ID_FROM_PREVIOUS_COMMAND>

# Download file with FILE_ID from DataHub to "helloworld.txt" on your computer
$ curl -H "X-Auth-Token: $DATAHUB_TOKEN" \
  -X GET "https://$ONEPROVIDER_HOST/api/v3/oneprovider/data/$FILE_ID/content" \
  -o helloworld.txt

# Upload "helloworld.txt" on your local computer to "uploadtest.txt" on DataHub
$ curl -H "X-Auth-Token: $DATAHUB_TOKEN" \
  -X POST \
  "https://$ONEPROVIDER_HOST/api/v3/oneprovider/data/$DIR_ID/children?name=uploadtest.txt" \
  -H "Content-Type: application/octet-stream" -d "@helloworld.txt"
```

Figure 220: Use of generated EGI token in API requests

Nonetheless, there are several limitations that currently hinder the realization of this use case for exploiting EGI within TRUSTEE. Although the data plan of the EGI project indicates open access to some data and authorized access to privacy-sensitive data, this data plan is currently deprecated and further contact with the EGI SDIS team is required for any updates to this plan. Additionally, up to July 2023, there has been no exploitable data publicly accessible. What

is more, for a user to be able to log in to EGI there is a requirement for an academic or social account, approved by EGI.

TRUSTEE as an EGI provider

By installing the Oneprovider service, adding certain resources, and registering it in Onezone, an entity may become an EGI provider. Providers may choose whether to support the space with empty storage or if existing data should be exposed through the Oneprovider, while Onezone enables communication and data sharing between providers from various zones. TRUSTEE as an EGI Provider could offer certain functionalities of TRUSTEE such as DPIA or the HE toolbox (SSI-HE and HEDF) for other providers to explore them, as part of its incremental strategy, and/or share data with other EGI Providers.

However, this use case faces several challenges as well. TRUSTEE is not a legal entity therefore it is currently not able to join EGI as an organization and become an EGI Provider. In addition to this, there is an annual fee that organizations must commit to if they wish to become EGI Providers, which is proportional to the organization's annual budget.

Further investigation will be conducted regarding the exploitation of EGI within TRUSTEE and will be reported in upcoming versions of this deliverable, namely D4.2 and D4.3, while TRUSTEE partners will continue to monitor whether any updates will be made available regarding the EGI data plan and the offered options for entering the EGI community as EGI Users and/or EGI Providers.

8.5.3. SDK INTERNAL CONNECTORS

SDK internal connectors enable communication between different components or services within a system. In the case of the TRUSTEE platform, internal connectors are facilitating the integration within different building blocks of the TRUSTEE system.

One way of achieving successful integration and communication is by utilizing OpenAPI specifications. The OpenAPI Specification⁶¹ is a specification for machine-readable interface files for describing, producing, consuming, and visualizing RESTful web services. It was formerly (2009-2017) known as the Swagger Specification.

Some benefits of using OpenAPI include:

- Language-agnostic interface for describing RESTful APIs,
- Machine-readable and easily interpretable by humans,
- Design first approach: the whole API with types and examples for every endpoint can be defined before the start of implementation work. With additional tools, mock APIs can be generated for testing until full implementation becomes available, thus speeding up the work and making agreements explicit
- Code generators: most languages are supported by their code-generators

8.6. VALIDATION

⁶¹ <https://spec.openapis.org/oas/latest.html>

Due to the specifics of the OneSS component and the extensive analysis needed for the development and implementation of OneSS external connector subcomponents, validation of the implemented solution for external connectors will be presented in the following versions of this deliverable, namely D4.2 and D4.3.

Since the external connectors of the OneSS are still under analysis and development, it is not possible to provide any validation result. However, the validation of the correct access to the TRUSTEE Developer Handbook have been carried out. It is implemented as Wiki pages and is available via the URL stated in the implementation section. Hence, the validation was executed via accessing the available content and checking the content against the KPIs for the OneSS available in D5.1 [68].

8.7. NEXT STEPS

As described in the previous sections, the main focus of the OneSS development so far was on providing the first version of the OneSS Developer Handbook containing tutorials, best practices, and information on the architecture and building blocks of the TRUSTEE platform. In terms of SDK external connectors, an analysis of several European Data Spaces was performed in terms of the development of OneSS external connectors. Initial integration cases were proposed for some of the connectors, while additional investigation is needed for the rest of them.

The next steps in the development of the OneSS will include:

- expanding the content of the Developer Handbook, providing additional documentation, tutorials, best practices, and updated information on the TRUSTEE ABBs, as the development of the blocks will progress;
- adding new types of materials to the OneSS Handbook, such as legal information and guidelines for the UI;
- developing SDK internal connectors according to the status of development of other TRUSTEE components and integration with them;
- expand research for SDK external connectors that still need to be further investigated;
- implementation of the SDK external connectors towards selected European Data Spaces;
- development of the proposed integration cases, i.e., for the IDS connector and GAIA-X.

8.8. CONCLUSIONS

An overview of the current status of the OneSS building block has been presented. The architecture of the OneSS building block is presented. The system requirements related to user-centred, technical, legal, and socio-ethical aspects that are addressed by the current version of the OneSS are also listed. An initial version of the TRUSTEE Developer Handbook was presented, as well as various European data spaces that are being explored in the context of external connectors implementation and integration with the TRUSTEE platform.

In addition to the IDS and GAIA-X connectors for which integration cases have been proposed, an analysis of EOSC and EGI as well as Copernicus connectors as potential contributors to TRUSTEE have been carried out. However, no final decision was made at the time of finalizing this deliverable so more work will be invested in this direction in upcoming months of the project and will be further discussed in future versions of the deliverable, namely D4.2 and D4.3.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- Tooling: a big ecosystem of tools created under the Swagger brand that can be used to streamline the work (generating API documentation, tests, mock servers), Swagger⁶² refers to a set of SmartBear tools which are among the most popular ones for developing and describing RESTful APIs. Swagger tools help users build, document, test and consume RESTful web services. It can be used with both a top-down and bottom-up API development approach.
- Widely accepted: many developers and organizations use it,
- Stable: known under the name of “Swagger” until 2017. OpenAPI Initiative is part of the Linux Foundation, which increases trustworthiness,
- Reliable single source of truth for APIs: clients and API developers/providers have a good point of reference through the API contract which OpenAPI describes.

As the integration of the TRUSTEE ABBs progresses, the content related to the OneSS internal connectors will be updated and provided in detail in the following versions of this deliverable, namely D4. 2 and D4.3.

Currently the external interfaces of the ABBs are identified in D2.1 [1], and are being extended with the concrete decisions in terms of the data (and data models) that will flow between SBBs (as the implementation of the ABBs) to achieve integration. Technologies, protocols and APIs are being defined after which the concrete “Internal Connectors” will be implemented for the SBBs to realize the integration. In cases where multiple SBBs will act as consumers (client) for external interfaces (APIs or other) such internal connectors will be prioritized for development of the SDKs (and accompanying documentation in Developer Handbook). Partners leading the work of the specific SBBs will act as developers of Internal connectors. Sharing the common practices, governance of the connector related activities will be managed from the OneSS and related task side.

⁶² <https://swagger.io/>

9. CONCLUSIONS

This document presents the initial version of the TRUSTEE ABBs developed under WP4:

- Dashboard (DA), for providing a graphical user-friendly point of entry to the TRUSTEE platform.
- Data Privacy Impact Assessment (DPIA), for identifying and analysing how data privacy is managed inside TRUSTEE.
- Self-Sovereign Identity Homomorphic capable framework (SSI-HE), for allowing confidential characterization and validation of datasets features of TRUSTEE Data Providers.
- Data Lake (DL), for ensuring semantic interoperability and flexible data management for all TRUSTEE ABBs.
- Security and Trust Manager (STM), for creating and managing the agreements between TRUSTEE, providers and consumers.
- One Stop Shop (OneSS), for providing the internal and external interfaces to enable the use of TRUSTEE in other applications or with existing European data spaces.

Each of the above-mentioned ABBs was described in a respective chapter in this document. For each ABB the needs and objectives have been discussed. Some basic concepts needed to understand the ABB design and implementation have also been included as Background. Then, the architecture design and implementation details have been exposed according to each ABB maturity. Finally, some validation tests have been proposed (and presented when available). The next steps for the following versions of each ABB have been presented, as well as the main conclusions of the current implementation for each ABB.

Throughout the implementation process and the technical development activities and investigation performed under WP4, as well as following the project's progress, additional technical requirements may emerge. As explained in the introduction section of this document, WP4 will provide feedback to WP2 under which requirements and architecture details are identified and reported, therefore additional technical requirements stemming from the implementation process will be reported in the upcoming D2.2 – “Live doc conceptualisation, use cases and system architecture V2”. The description of the presented ABBs will be revised, enhanced, and complemented as the project progresses and the maturity of the ABBs improves. The updated details will be gathered in future versions of the deliverable, namely D4.2 and D4.3.

10. REFERENCES

- [1] TRUSTEE consortium, "TRUSTEE D2.1 Live doc conceptualisation, use cases and system architecture," 2023.
- [2] TRUSTEE consortium, "TRUSTEE D3.1 TRUSTEE CORE & FAIR," 2023.
- [3] "AMQP - Advanced Message Quering Protocol," 2023. [Online]. Available: <https://www.amqp.org/>. [Accessed June 2023].
- [4] European Commission, "Rolling plan for ICT standardisation," 2020. [Online]. Available: <https://digital-strategy.ec.europa.eu/en/policies/rolling-plan-ict-standardisation>. [Accessed June 2023].
- [5] DIRECTIVE (EU) 2019/882 OF THE EUROPEAN PARLIAMENT, "The accessibility requirements for products and services (Text with EEA relevance)," 2019.
- [6] J. Juliver, "9 Guidelines & Best Practices for Exceptional Web Design and Usability.," November 2022. [Online]. Available: <https://blog.hubspot.com/blog/tabid/6307/bid/30557/6-guidelines-for-exceptional-website-design-and-usability.aspx>. [Accessed June 2023].
- [7] M. Walburg, "Anatomy of a Website – Key Components and Design Terms," August 2021. [Online]. Available: <https://binarapps.com/anatomy-of-a-website-key-components-and-design-terms/>. [Accessed June 2023].
- [8] I. Hernandez, "Top 6 Basic Elements of Web Design," DreamHost, April 2021. [Online]. Available: <https://www.dreamhost.com/blog/elements-of-webdesign/>. [Accessed June 2023].
- [9] W3C, "WCAG 2 Overview," W3C, July 2023. [Online]. Available: <https://www.w3.org/WAI/standards-guidelines/wcag/>. [Accessed August 2023].
- [10] W3C, "Design and Develop Overview," W3C, May 2022. [Online]. Available: <https://www.w3.org/WAI/design-develop/>. [Accessed June 2023].
- [11] C. Vallaure, "Ultimate Guide to Color in UX/UI Design," Medium, January 2023. [Online]. Available: <https://uxplanet.org/ultimate-guide-to-color-in-ux-ui-design-de8eb104b5d3>. [Accessed June 2023].
- [12] W3C, "Web Content Accessibility Guidelines (WCAG) 2.0," W3C, December 2008. [Online]. Available: <https://www.w3.org/TR/WCAG20/>. [Accessed June 2023].

- [13] A. Rajput, "Website Color Scheme: A Guide To Create Better User Experience," 300Mind, October 2022. [Online]. Available: <https://300mind.studio/blog/blog/website-color-scheme/>. [Accessed June 2023].
- [14] TRUSTEE Consortium, "TRUSTEE D6.3 - Dissemination and Communication Strategy and Plan.," 2022.
- [15] L. Rello and J. P. Bigham, "Good Background Colors for Readers: A Study of People with and without Dyslexia," in *19th International ACM SIGACCESS Conference on Computers and Accessibility*, Baltimore Maryland USA, 2017.
- [16] M. Bernard, C. H. Liao and M. Mills, "The effects of font type and size on the legibility and reading time of online text by older adults," in *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, Seattle Washington, 2001.
- [17] R. H. Hall and P. Hanna, "The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention," in *Behaviour & Information Technology*, vol. 23, no. 3, pp. 183–195, 2004.
- [18] L. Rello, M. Pielot and M. Marcos, "Make It Big!: The Effect of Font Size and Line Spacing on Online Readability," in *2016 CHI Conference on Human Factors in Computing Systems*, pp. 3637–3648, San Jose California, 2016.
- [19] "Font size conversion: pixel-point-em-rem-percent," WebSemantics, January 2021. [Online]. Available: <https://websemantics.uk/tools/font-size-conversion-pixel-point-em-rem-percent/>. [Accessed June 2023].
- [20] "Font sizes," Spectrum, [Online]. Available: <https://spectrum.adobe.com/page/typography/#Font-sizes>. [Accessed June 2023].
- [21] K. Minakata and S. Beier, "The dispute about sans serif versus serif fonts: An interaction between the variables of serif and stroke contrast," in *Acta Psychologica*, vol. 228, p. 103623, 2022.
- [22] "Rules of Typography," amadine.com, January 2023. [Online]. Available: <https://amadine.com/useful-articles/rules-of-typography>. [Accessed June 2023].
- [23] J. Tiao, "7 web typography rules," Medium, February 2023. [Online]. Available: <https://uxdesign.cc/7-web-typography-rules-27de68c60f6>. [Accessed June 2023].
- [24] J. Ling and P. van Schaik, "The influence of font type and line length on visual search and information retrieval in web pages," in *International Journal of Human-Computer Studies*, vol. 64, no. 5, pp. 395–404, 2006.

- [25] L. Moreno, X. Valencia, J. E. Pérez and M. Arrue, “Exploring the Web navigation strategies of people with low vision,” in *Proceedings of the XIX International Conference on Human Computer Interaction*, pp. 1–8., Palma Spain, 2018.
- [26] A. Burrell and A. C. Sodan, “Web Interface Navigation Design: Which Style of Navigation-Link Menus Do Users Prefer?,” in *22nd International Conference on Data Engineering Workshops (ICDEW’06)*, pp. 42–42., 2006.
- [27] G. Bailly, E. Lecolinet and L. Nigay, “Visual Menu Techniques,” in *ACM Comput. Surv.*, vol. 49, no. 4, p. 60:1-60:41, 2016.
- [28] “Luminosity Colour Contrast Ratio Analyser,” Juicy Studio, [Online]. Available: <https://juicystudio.com/services/luminositycontrastratio.php#specify>. [Accessed June 2023].
- [29] E. Canziba, “Hands-On UX Design for Developers,” Packt Publishing Ltd., 2018.
- [30] S. Morson, “Learn Design for iOS Development. Learn Design for IOS Development,” <https://doi.org/10.1007/978-1-4302-6365-4>, 2014.
- [31] European Union, “Communication and visibility rules,” 2022. [Online]. Available: <https://op.europa.eu/en/publication-detail/-/publication/d1d3df9b-03e9-11ed-acce-01aa75ed71a1/language-en>. [Accessed June 2023].
- [32] REGULATION (EU) 2016/679, “Protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation),” vol. 679, p. 2016., 2016.
- [33] DIRECTIVE (EU) 2016/680 OF THE EUROPEAN PARLIAMENT, “The protection of natural persons with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, ...,” 2016.
- [34] ARTICLE 29 DATA PROTECTION WORKING PARTY, “Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is “likely to result in a high risk” for the purposes of Regulation 2016/679,” (WP 248 rev.01 2017), 2017.
- [35] DIRECTIVE 95/46/EC OF THE EUROPEAN PARLIAMENT, “ The protection of individuals with regard to the processing of personal data and on the free movement of such data,” 1995.
- [36] Data Protection Commission (DPC), “Guide to Data Protection Impact Assessments (DPIAs),” October 2019. [Online]. Available: https://www.dataprotection.ie/sites/default/files/uploads/2019-10/Guide%20to%20Data%20Protection%20Impact%20Assessments%20%28DPIAs%29_Oct19_0.pdf. [Accessed June 2023].

- [37] CNIL, "CNIL methodology; PIA, methodology," February 2018. [Online]. Available: <https://www.cnil.fr/sites/cnil/files/atoms/files/cnil-pia-1-en-methodology.pdf>. [Accessed June 2023].
- [38] European Parliament, "Free flow of non-personal data in the European Union," February 2018. [Online]. Available: [https://www.europarl.europa.eu/RegData/etudes/BRIE/2018/615647/EPRS_BRI\(2018\)615647_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2018/615647/EPRS_BRI(2018)615647_EN.pdf). [Accessed June 2023].
- [39] Barclay, Iain and Radha, Swapna and Preece, Alun and Taylor, Ian and Nabrzyski, Jarek, "Certifying provenance of scientific datasets with self-sovereign identity and verifiable credentials," in *arXiv preprint arXiv:2004.02796*, 2020.
- [40] Kulabukhova, NV, "Zero-knowledge proof in self-sovereign identity," in *CEUR Workshop Proceedings*, 2019.
- [41] Benarroch, Daniel and Campanelli, Matteo and Fiore, Dario and Gurkan, Kobi and Kolonelos, Dimitris, "Zero-knowledge proofs for set membership: Efficient, succinct, modular," *Designs, Codes and Cryptography*, pp. 1-69, 2023.
- [42] Yi, Xun and Paulet, Russell and Bertino, Elisa and Yi, Xun and Paulet, Russell and Bertino, Elisa, Homomorphic encryption, Springer, 2014.
- [43] Acar, Abbas and Aksu, Hidayet and Uluagac, A Selcuk and Conti, Mauro, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1-35, 2018.
- [44] Moore, Ciara and O'Neill, Maire and O'Sullivan, Elizabeth and Doroz, Yark and Sunar, Berk, "Practical homomorphic encryption: A survey," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.
- [45] Naeem, Noshaba and Khan, Fawad and Yaqoob, Tahreem and Tahir, Shahzaib, "Privacy-Preserving Computing via Homomorphic Encryption: Performance, Security, and Application Analysis," *Handbook of Research on Cybersecurity Issues and Challenges for Business and FinTech Applications*, 2023.
- [46] Ducas, Leo and Micciancio, Daniele, FHEW: bootstrapping homomorphic encryption in less than a second, Springer, 2015.
- [47] Gentry, Craig and Halevi, Shai and Peikert, Chris and Smart, Nigel P, "Ring switching in BGV-style homomorphic encryption," in *International Conference on Security and Cryptography for Networks*, 2012.

- [48] Boneh, Dan and Gennaro, Rosario and Goldfeder, Steven and Jain, Aayush and Kim, Sam and Rasmussen, Peter MR and Sahai, Amit, "Threshold cryptosystems from threshold fully homomorphic encryption," in *Advances in Cryptology--CRYPTO 2018*, Santa Barbara, 2018.
- [49] Jain, Aayush and Rasmussen, Peter MR and Sahai, Amit, "Threshold fully homomorphic encryption," *Cryptology ePrint Archive*, 2017.
- [50] Preukschat, Alex and Reed, Drummond, *Self-sovereign identity*, Manning Publications, 2021.
- [51] Fedrecheski, Geovane and Rabaey, Jan M and Costa, Laisa CP and Ccori, Pablo C Calcina and Pereira, William T and Zuffo, Marcelo K, "Self-sovereign identity for IoT environments: a perspective," in *2020 Global Internet of Things Summit (GloTS)*, 2020.
- [52] Ishmaev, Georgy and Stokkink, Quinten, "Identity management systems: Singular identities and multiple moral issues," *Frontiers in Blockchain*, vol. 3, no. 15, 2020.
- [53] Bai, Yirui and Lei, Hong and Li, Suozai and Gao, Haoyu and Li, Jun and Li, Leixiao, "Decentralized and self-sovereign identity in the era of blockchain: a survey," in *2022 IEEE International Conference on Blockchain (Blockchain)*, 2022.
- [54] Ferdous, Md Sadek and Chowdhury, Farida and Alassafi, Madini O, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103059-103079, 2019.
- [55] Daep, Ulrich and Gorkin, Pamela and Daep, Ulrich and Gorkin, Pamela, "Fermat's little theorem," *Reading, Writing, and Proving: A Closer Look at Mathematics*, pp. 315-323, 2011.
- [56] Y. Chen , J. Yu and F. Wang, "A semantic interoperability framework for integrating healthcare information systems," *Future Generation Computer Systems*, no. 103, pp. 293-304, 2020.
- [57] L. M. Silva, L. Ferreira Pires and M. & van Sinderen, "Towards data semantic interoperability for IoT systems: A survey.," *Information Systems Frontiers*, no. 21, pp. 853-876., 2019.
- [58] N. Ivezic and H. & Garcia-Molina, "Semantic challenges in distributed data management," in *VLDB Endowment*, 12(12), 2067-2070, 2019.
- [59] EOSC, "European Open Science Cloud," 2023. [Online]. Available: <https://eosc-portal.eu/>. [Accessed June 2023].
- [60] M. T. Horsch and B. & Schembera, "Documentation of epistemic metadata by a mid-level ontology of cognitive processes," in *JOWO 2022: The Joint Ontology Workshops*, 2022.
- [61] E. Robinson, M. Buys, J. Chodacki, K. Garzas, S. Monfort, C. Nancarrow and N. & Davies, "FAIR Island: real-world examples of place-based open science.," *GigaScience*, no. 12, 2023.

- [62] R. da Silva, R. Badia, V. Bala, D. Bard, P. Bremer and et al, "Workflows Community Summit 2022: A Roadmap Revolution," *arXiv preprint*, vol. arXiv:2304.00019, 2022.
- [63] A. Nambiar and D. & Mundra, "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management," *Big Data and Cognitive Computing*, no. 6, p. 132, 2022.
- [64] C. Giebler, C. Gröger, E. Hoos, R. Eichler, H. Schwarz and B. & Mitschang, "The data lake architecture framework: a foundation for building a comprehensive data lake architecture.," *Database Systems for Business, Technology and Web (BTW)*, vol. 70469, 2021.
- [65] A. Kulkarni, P. Bassin, N. S. Parasa, V. E. Venugopal, S. Srinivasa and C. & Ramanathan, "Ontology Augmented Data Lake System for Policy Support," in *In International Conference on Big Data Analytics (pp. 3-16)*, 2022.
- [66] M. Cherradi, A. El Haddadi and H. Routaib, "Data Lake Management Based on DLDS Approach," in *Networking, Intelligent Systems and Security: Proceedings of NISS 2021 (pp. 679-690)*, 2022.
- [67] G. Jin, H. Bian, Y. Chen and X. & Du, "Columnar Storage Optimization and Caching for Data Lakes," *EDBT*, pp. pp. 2-419, 2022.
- [68] TRUSTEE consortium, "TRUSTEE D5.1 Report on TRUSTEE Pilot Campaign Plan and Evaluation Methodology," 2023.
- [69] R. Studer, V. Benjamins and D. Fensel, "Knowledge engineering: Principles and methods," *Data & Knowledge Engineering*, vol. 25, no. 1-2, pp. 161-197, 1998.
- [70] S. C. Rasmusen, M. Penz, S. Widauer, P. Nako, A. Kurteva, A. Roa-Valverde and a. A. Fensel, "Raising consent awareness with gamification and knowledge graphs: An automotive use case," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 18, no. 1, pp. pp. 1-21, 2022.
- [71] S. Kirrane, J. D. Fernández, P. Bonatti, U. Milosevic, A. Polleres and and R. Wenning, "The special-k personal data processing transparency and compliance platform," *arXiv preprint arXiv:2001.09461*, 2020.
- [72] A. Kurteva, T. R. Chhetri, H. J. Pandit and a. A. Fensel, "Consent through the lens of semantics: State of the art survey and best practices," *Semantic Web*, no. Preprint, pp. 1-27, 2021.
- [73] T. R. Chhetri, A. Kurteva, R. J. DeLong, R. Hilscher, K. Korte and a. A. Fensel, "Data Protection by Design Tool for Automated GDPR Compliance Verification Based on Semantically Modeled Informed Consent," *Sensor*, vol. 22, no. 7, p. 2763, 2022.
- [74] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv: Machine Learning*, 2017.

- [75] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, p. 115:1–115:35, 2022.
- [76] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl and B. Yu, "Interpretable machine learning: definitions, methods, and applications," *CoRR*, vol. abs/1901.04592, 2019.
- [77] Stanford Center for Biomedical Informatics Research, "Protégé," [Online]. Available: <https://protege.stanford.edu/>. [Accessed June 2023].
- [78] "Flutter - Build apps for any screen," [Online]. Available: <https://flutter.dev/>. [Accessed June 2023].
- [79] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review* 11(2), 1996.
- [80] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak and B. Mons, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1-9, 2016.
- [81] W3C, "OWL 2 Web Ontology Language," [Online]. Available: <https://www.w3.org/TR/owl2-overview/>. [Accessed June 2023].
- [82] W3C, "PROV-O: The PROV Ontology," [Online]. Available: <https://www.w3.org/TR/2013/REC-prov-o-20130430/>. [Accessed June 2023].
- [83] "The smashHitCore Ontology," [Online]. Available: <https://smashhiteu.github.io/smashHitCore/>. [Accessed June 2023].
- [84] W3C, "Data Privacy Vocabulary (DPV)," [Online]. Available: <https://w3c.github.io/dpv/dpv/>. [Accessed January 2023].
- [85] W3C, "DCAT 2 Vocabulary," [Online]. Available: <https://www.w3.org/ns/dcat>. [Accessed June 2023].
- [86] Schema.org, "Welcome to Schema.org," [Online]. Available: <https://schema.org>. [Accessed June 2023].
- [87] EOSCpilot, "EOSCpilot's Contributions to EOSC Services," European Open Science Cloud for Research Pilot Projects, 2023. [Online]. Available: <https://www.eoscipilot.eu/eoscipilots-contributions-eosc-services>. [Accessed June 2023].
- [88] Copernicus Data Space Ecosystem, "Copernicus Data Space Ecosystem," [Online]. Available: <https://dataspace.copernicus.eu/>. [Accessed June 2023].
- [89] "Open Access Hub," SERCO, [Online]. Available: <https://scihub.copernicus.eu/>. [Accessed June 2023].

- [90] Copernicus, “Service description and evolution roadmap,” [Online]. Available: <https://documentation.dataspace.copernicus.eu/Roadmap.html>. [Accessed June 2023].
- [91] Copernicus, “Documentation – apis,” [Online]. Available: <https://documentation.dataspace.copernicus.eu/APIs.html>. [Accessed June 2023].
- [92] Copernicus, “Documentation - Data,” [Online]. Available: <https://documentation.dataspace.copernicus.eu/Data.html>. [Accessed June 2023].
- [93] Copernicus, “Copernicus Data Space Ecosystem Token Generation,” [Online]. Available: <https://documentation.dataspace.copernicus.eu/APIs/Token.html>. [Accessed June 2023].
- [94] EGI, “EGI - Advanced Computing Services for Research,” [Online]. Available: <https://www.egi.eu/>. [Accessed June 2023].
- [95] European Commission, “Research and Technological Development for an International Data Grid,” [Online]. Available: <https://cordis.europa.eu/project/id/IST-2000-25182E>. [Accessed June 2023].
- [96] Onedata, “Onedata,” [Online]. Available: <https://www.onedata.org/#/home>. [Accessed June 2023].
- [97] V. Driessen, “A successful Git branching model,” January 2010. [Online]. Available: <https://nvie.com/posts/a-successful-git-branching-model/>. [Accessed June 2023].
- [98] Atlassian, “Gitflow Workflow,” [Online]. Available: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>. [Accessed June 2023].
- [99] “Trunk Based Development,” [Online]. Available: <https://trunkbaseddevelopment.com/>. [Accessed June 2023].

11. ANNEX 1: GLOSSARY OF TERMS

This section contains the more important terms for the understanding of the deliverable. More details can be found in D2.1 glossary definition [1].

Architecture Building Block (ABB) - a constituent of the architecture model that describes a single aspect of the overall model. An Architecture Building Block describes generic characteristics and functionalities. Architecture Building Blocks are used to describe reference architectures, solution architecture templates or solution architectures of specific solutions. (TOGAF) (see **Building Block**)

Artificial intelligence (AI) – Artificial intelligence (AI) systems are software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by perceiving their environment through data acquisition, interpreting the collected structured or unstructured data, reasoning on the knowledge, or processing the information, derived from this data and deciding the best action(s) to take to achieve the given goal. The AI system can generate outputs such as content, predictions, recommendations, or decisions influencing the environments they interact with. AI systems can either use symbolic rules or learn a numeric model, and they can also adapt their behaviour by analysing how the environment is affected by their previous actions. As a scientific discipline, AI includes several approaches and techniques, such as machine learning approaches (including supervised, unsupervised and reinforcement learning, using a wide variety of methods including deep learning), machine reasoning (which includes, in particular, planning, scheduling, search, and optimization), logic- and knowledge-based approaches (including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning and expert systems), statistical approaches, Bayesian estimation and robotics (which includes control, perception, sensors and actuators, as well as the integration of all other techniques into cyber- physical systems). [White Paper On Artificial Intelligence - A European approach to excellence and trust. European Commission. COM(2020) 65 final; [HLEG AI, Assessment List for Trustworthy AI \(ALTAI\)](#); Proposal of the EU Artificial Intelligence Act (Brussels, 21.4.2021 COM(2021) 206 final 2021/0106 (COD)), Article 2, point 1.]

Asymmetric cryptography - cryptography that uses two separate keys to exchange data, one to encrypt or digitally sign the data and one for decrypting the data or verifying the digital signature. Also known as public key cryptography. [[NIST SP 800-77 Rev. 1](#) under Asymmetric Cryptography]

Attribute - inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means [ISO/IEC 25000:2014]

Automatic/Automated/Automation - pertaining to a process or system that, under specified conditions, functions without human intervention. [[ISO/IEC DIS 22989\(en\). Terms related to Artificial Intelligence](#)].

Building Block - a (potentially re-usable) component of enterprise capability that can be combined with other building blocks to deliver architectures and solutions. Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached. For instance, at an early stage, a building block can simply consist of a name or an outline description. Later, a building block may be decomposed into multiple supporting building blocks and may be accompanied by a full specification. Building blocks can relate to "architectures" or "solutions". According to The Open Group, building blocks have several generic characteristics: see **Architecture Building Block, Solution Building Block**. According to The Open Group, building blocks have several generic characteristics: A building block is a package of functionality defined to meet the business needs across an organization, A building block has published interfaces to access the functionality, A building block may have multiple implementations but with different inter-dependent “building blocks.” A building block may interoperate with other, interdependent building blocks, A good building block considers implementation and usage and evolves to exploit

technology and standards, It may be assembled from other building blocks, It may be a subassembly of other building blocks, Ideally, a building block is re-usable and replaceable, and well-specified.

Certificate Authority (CA) - a trusted entity that issues and revokes public key certificates. [NIST SP 1800-16B under Certificate Authority from NISTIR 8149]

Ciphertext (CT)— the unintelligible and seeming random form of data that is produced by the cryptographic function of encryption. Ciphertext is produced by a symmetric algorithm when a data set is transformed by the encryption process using a selected key. Ciphertext can be converted back into its original form (i.e. plain text) by performing the decryption process using the same symmetric encryption algorithm and the key used during the encryption process. (Also known as a cryptogram.) [The A to Z of Cybersecurity Glossary by [Simplicity](#)]

Confidentiality - the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. The purpose of the confidentiality service is to protect information from unauthorized disclosure. [ITU-TX.800] **Cross-border processing** – means either processing of personal data which takes place in the context of the activities of an organisation in more than one Member State where that organisation is established in more than one Member State; or processing of personal data which takes place in the context of the activities of an organisation's single establishment but where that processing substantially affects or is likely to substantially affect data subjects in more than one Member State. [GDPR, Article 4(23)]

Consent - consent of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her. [Article 4(11) of the GDPR]

Control - a measure that is modifying risk. Controls include any process, policy, device, practice, or other actions that modify risk. (ISO/IEC 27000: 2018)

Cross-border processing – means either processing of personal data which takes place in the context of the activities of an organisation in more than one Member State where that organisation is established in more than one Member State; or processing of personal data which takes place in the context of the activities of an organisation's single establishment but where that processing substantially affects or is likely to substantially affect data subjects in more than one Member State. [GDPR, Article 4(23)]

Data - a representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means [IEEE Glossary]. A digital representation of acts, facts or information and any compilation of such acts, facts or information, including in the form of sound, visual or audiovisual recording. [DGA, [Article 2\(1\)](#)]

Data Confidentiality - the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. (ISO/IEC PDTR 13335-1). Confidentiality in a general sense refers to the duty not to share information with persons who are not qualified to receive that information (see Article 5(f) of Regulation (EU) 2016/679) [[EDPS Data protection Glossary](#)].

Data holder – legal person, including public sector bodies and international organisations, or a natural person who is not a data subject with respect to the specific data in question, which, in accordance with applicable law, has the right to grant access to or to share certain personal data or non-personal data [Article 2(8)DGA]; means a legal or natural person, who has accessed data from the connected product or has generated data during the provision of a related service and who has the contractually agreed right to use such data, and the obligation, in accordance with this Regulation, applicable Union law or national legislation implementing Union law to make available certain data to the

user or a data recipient [Data Act, Article 2(6), version after amendments of the EP, available at: [Texts adopted - Data Act - Tuesday, 14 March 2023 \(europa.eu\)](#)]

Data management - the activities of defining, creating, storing, maintaining and providing access to data and associated processes in one or more information systems. [b-ISO/IEC TR 10032:2003]

Data processing - the ability to handle data as input and apply different treatments that might modify the data or combine it without modifying it with other data in order to produce an output that is useful for a given application or service in the data lifecycle. In terms of the GDPR: *Any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction* [Article 4(2) of the GDPR]. Related definition – **restriction of processing** (see below).

Data processor - see **Processor**.

Data protection - means the implementation of appropriate administrative, legal, technical or physical means to guard against unauthorized intentional or accidental disclosure, modification, or destruction of data.

Data provider - organization which produces or collects and further shares data or metadata. [OECD]

Data recipient – in the context of personal data protection: means a natural or legal person, public authority, agency or another body, to which the personal data are disclosed, whether a third party or not. However, public authorities which may receive personal data in the framework of a particular inquiry in accordance with Union or Member State law shall not be regarded as recipients; the processing of those data by those public authorities shall be in compliance with the applicable data protection rules according to the purposes of the processing. [Article 4(9) of the GDPR]. In the context of Data Act: means a legal or natural person other than the user of a connected product or related service, to whom a data holder makes available data accessed from a connected product or generated during the provision of a related service following an explicit request by the user or in accordance with a legal obligation under Union law or national legislation implementing Union law [Data Act, Article 2(7), version after amendments of the EP, available at: [Texts adopted - Data Act - Tuesday, 14 March 2023 \(europa.eu\)](#)]

Data trust - includes actions taken to safeguard security, privacy, and quality of data and enhance data trustworthiness.

Data use - after the data visualization and analysis, the data use is the process that application run to either understand the monitored phenomena by analyzing the past and monitor the present or predict the future and build new decisions.

Data user - a natural or legal person who has lawful access to certain personal or non-personal data and has the right, including under the GDPR in the case of personal data, to use that data for commercial or non-commercial purposes. [DGA, Article 2(9)]

Developer - individual or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the system or software life-cycle process [: ISO/IEC 25000:201]

FAIR Data principles - the FAIR Data Principles are a set of guiding principles in order to make data findable, accessible, interoperable and reusable.

Homomorphic Encryption (Homomorphic Cryptography) – encryption schemes allowing certain mathematical operations to be performed directly on ciphertexts, without prior decryption. Homomorphic encryption can be a

powerful tool for leveraging multi-party computations, by enabling a participant to compute functions on values while keeping the values hidden.

Legal Requirement - a responsibility placed on an entity, including statutory or regulatory obligations which an entity is required to fulfil, to ensure legal compliance of its actions.

Metadata - data collected on any activity of a natural or legal person for the purposes of the provision of a data sharing service, including the date, time and geolocation data, duration of activity, connections to other natural or legal persons established by the person who uses the service. [DGA COM (2020) 767 FINAL]

Non-disclosure agreement – a legal contract between two parties or among more parties stating the use and sharing of data among the parties for certain purposes but restricts access to.

Personal data - means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person [Regulation (EU) 2016/679 (General Data Protection Regulation), Article 4 (1)]. Data which relates to a living individual who can be identified: a) from those data; or b) from those data and other information which is in possession of, or is likely to come into the possession of, the data controller, and includes any expression of opinion about the individual and any indication of the intentions of the data controller or any other person in respect of the individual. [b-PAS 185:2017]

Personally Identifiable Information (PII) - personally Identifiable Information is defined as any information about an individual maintained by an [entity], including any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, and any other information that is linked or linkable to an individual([NIST10] p. 2-1). PII is a widely used term for personal data in the domain of information security. Within this document, PII is used in relation to information security

Privacy Impact Assessment (PIA) – in other words Data Protection Impact Assessment (DPIA) – describes a process designed to identify risks arising out of the processing of personal data and to minimise these risks as far and as early as possible. DPIAs are important tools for negating risk, and for demonstrating compliance with the GDPR. Required by Article 35 of the GDPR where data processing “is likely to result in a high risk to the rights and freedoms of natural persons”.

Sensitive data - any information that is protected against unwarranted disclosure. Protection of data may be required for legal or ethical reasons, for issues pertaining to personal privacy, or for proprietary considerations. [CSC]

Sensitive Personal Data / Special category of personal data – Personal data which are, by their nature, particularly sensitive in relation to fundamental rights and freedoms merit specific protection as the context of their processing could create significant risks to the fundamental rights and freedoms. In the GDPR sensitive personal data are considered as “special categories of personal data”. The following personal data is considered ‘special category of personal data’ and is subject to specific processing conditions stemming in particular from the GDPR: personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, genetic data, biometric data processed for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation. [Article 4(13), 4(14), 4(15), 9(1) and Recitals 51 to 56 of the GDPR, https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/legal-grounds-processing-data/sensitive-data/what-personal-data-considered-sensitive_en

12. ANNEX 2: LEGAL REQUIREMENTS

The following sections detail the preliminary analysis of legal requirements presented in the D2.1 [1] related to each component. The ID of the respective legal requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether the ABBs comply with the requirement and how. The approach presented below for addressing each legal requirement may produce additional technical, functional and non-functional, requirements that will be further specified in upcoming versions of this deliverable, namely D4.2 and D4.3, as well as in D2.2.

12.1. DA

Table 73: DA – Legal requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
LEG-TSD-002	TRUSTEE must have the fundamental rights principles at its core, by ensuring that the privacy of the individuals is respected and protected.	Yes	DA will not be accessible to unauthorized users.
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Yes	DA will employ SSL and TLS for secure communications.
LEG-TSD-007	Security solutions recommended to be considered: - Backups, if possible encrypted and checked regularly should be made, to enable restoring the system. - Limitation of the size of the software stack used, - Installation of critical updates without delay, - Automation of a vulnerability watch, - Application of vulnerability detection tools, - Restriction of physical and software access to diagnostic and remote configuration ports, - Protection of the databases made available online, - Secure database management.	Yes	DA will considerate these security solutions.
LEG-TSD-017	The information provided to the user of the platform, being a natural person, whose credentials/log in details will be processed on TRUSTEE, shall fulfil the below requirements: • concise, transparent, intelligible and easily accessible • a clear and plain language • in writing including by electronic means • recommendation to use layered privacy statements. No information required if it proves impossible or would involve a disproportionate effort to contact the data subject.	Yes	Information on data processing will be presented clearly to end-users through the DA

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design), and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Yes	DA will not be accessible from unauthorized users.
LEG-TSD-023	Logging systems should be used to detect abnormalities or security events.	Yes	TRUSTEE RPA component produce logs for abnormalities or security events detection while the DA ABB logs any user activity in the DA into the ATR ABB
LEG-TSD-024	Penetration testing should be considered and performed.	Yes	DA will consider penetration testing
LEG-TSD-025	Efficient version control system should be set up, to track different versions of the code over the time, with appropriate authentication and permission system and regular backups, as well as efficient development system.	Yes	DA will investigate the employment of GIT version control.
LEG-TSD-026	Code quality metrics tools should be considered.	Yes	DA will consider the use of code quality metrics tools
LEG-TSD-027	Control of the used libraries and SDKs should be applied.	Yes	DA will investigate applying control on the user libraries and SDKs
LEG-TSD-029	When using types of solutions with commercial support, TRUSTEE system providers should contractually ensure maintenance and updates of the code during the whole lifecycle of the project.	Yes	DA will investigate the maintainability of solutions with commercial support.
LEG-TSD-031	Implementation of cryptographic algorithms and protocols by TRUSTEE system developers by their own should be avoided, and choosing cryptographic libraries which are maintained, easy to use and recognised should be encouraged.	Yes	DA will not use any proprietary cryptographic algorithm.
LEG-TSD-032	Appropriate maintenance of libraries and SDKs should be considered, e.g. using dependency management system, managing updates to the dependencies, monitoring the versions of libraries and SDKs and regularly check the open source projects and their status, especially in the context of the change of domain or package ownership.	Yes	DA will investigate and consider the maintainability of libraries and SDKs.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

LEG-TSD-033	Quality and security of the code and its documentation, as well as the documentation of the system architecture should be ensured.	Yes	Documentation and diagrams will be used to achieve this requirement.
LEG-TSD-035	When data security is compromised, TRUSTEE system providers must report the violation to the data controllers (Data providers and Data consumers) without undue delay if it is likely to pose a risk to the rights and freedoms of individuals.	Yes	DA COULD present to the data controllers if any violation occurs, receiving this information from other ABBs.
LEG-TSD-037	TRUSTEE should allow the users to conduct a DPIA prior to any data processing activity (i.e. prior to accessing (encrypted) data and conducting computing activities with the use of data).	Yes	DA SHOULD include a clearly visible DPIA tab leading to the DPIA page.
LEG-TSD-044	The security measures taken by TRUSTEE system developers shall include: (a) detection processes and procedures maintained and tested to ensure timely and adequate awareness of anomalous events; (b) processes and policies on reporting incidents and identifying weaknesses and vulnerabilities in their information systems; (c) a response in accordance with established procedures and reporting the results of the measure taken; (d) an assessment of the incident's severity, documenting knowledge from incident analysis and collection of relevant information which may serve as evidence and support a continuous improvement process.	Partially	DA will consider these security measures.
LEG-TSD-048	TRUSTEE system developers shall implement appropriate incident reporting procedures.	Yes	DA will investigate reporting procedures by visualizing information coming from other ABBs.
LEG-TSD-049	TRUSTEE providers shall disclose the terms and conditions that apply to the services provided by them, as well as shall clearly describe the policies governing their provision.	Yes	DA will display the required terms and conditions.
LEG-TSD-104	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and providers must cooperate with national authorities upon the receipt of an order to act against illegal content or to provide specific information about service recipients.	Yes	DA will investigate the cooperation with national authorities to act against illegal content.
LEG-TSD-105	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and consortium should designate points of contact both for the recipients of their services and for member	Yes	DA will investigate the designation of contact points.

	states' authorities, the European Commission and the European Board for Digital Services		
LEG-TSD-106	If TRUSTEE is considered as an online intermediary service, TRUSTEE should inform their users in TRUSTEE Terms of Use of "any restrictions that they impose in relation to the use of their service in respect of information provided by the recipients of the service" in a "clear, plain, intelligible, user friendly and unambiguous language" and be "in an easily accessible and machine-readable format". Should there be any significant changes to these terms and conditions, the TRUSTEE service providers/ system developers need to notify their data providers.	Partially	DA will display the required terms and conditions.
LEG-TSD-107	TRUSTEE system developers must abstain from designing and operating their interfaces in a manipulative or deceitful manner or in a way hindering service recipients' ability to make independent and informed choices. (This is an additional requirement for providers of online platforms, under the category of hosting services. To the extent that TRUSTEE's activities and service provision falls under this category, this could be applied.)	Yes	DA MUST be designed to avoid manipulation and deception of the users, while also enabling them to make independent and well-informed decisions.
LEG-USR-016	When a data subject exercises its right of erasure and/or restriction to the processing, Data Provider must ensure to remove the data from TRUSTEE / stop sharing the data of data subjects when requested by a data subject, required by the data protection laws and when there is no legal basis for further sharing and processing of the data.	Yes	DA COULD display an option for withdraw of data that have been registered as part of the Distributed Data Sources of TRUSTEE.
LEG-USR-024	Data Provider and Data Consumer must use the technical and organisational measures to protect personal data. Data Provider and Data Consumer should use measures offered by TRUSTEE (Homomorphic encryption, SSI, authentication, federated learning, etc.).	Yes	The DA will allow users to use the mentioned functionalities offered by TRUSTEE by providing the UI with which the user interacts and/or by redirecting the user to the appropriate UI of a respective ABB
LEG-USR-065	Data provider / Model provider competent to grant or refuse access for the re-use of the data are required to make the conditions for allowing such re-use publicly available. The conditions for re-use shall be	Yes	The DA will redirect the user to the UI of the STM ABB, where such

non-discriminatory, transparent, proportionate and objectively justified with regard to the categories of data and the purposes of re-use and the nature of the data for which re-use is allowed.	actions can be displayed
---	--------------------------

12.2. DPIA

Table 74: DPIA – Legal requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
LEG-TSD-001	Should the TRUSTEE system developers and parties maintaining the system at any point handle or process any type of personal data, they should balance their legitimate interest with the fundamental rights and freedoms of the data subjects whose data they will be processing.	Yes	Any data processing will be performed taking into account the balance between legitimate interest and fundamental rights of users/individuals, only the data of the users necessary to conduct the DPIA will be processed by TRUSTEE system providers.
LEG-TSD-002	TRUSTEE must have the fundamental rights principles at its core, by ensuring that the privacy of the individuals is respected and protected.	Yes	The DPIA tool will not be accessible to unauthorized users and will request only the provision of data necessary to conduct a DPIA.
LEG-TSD-003	TRUSTEE must be developed as the system ensuring appropriate and secure processing of personal data. This is relevant especially for: a) the data considered as “special category data”, such as data concerning health or genetic data, as well as b) personal data concerning minors. These data will be processed by the future users of TRUSTEE (Data providers, Data consumers), especially in the HEALTH and EDUCATION domain.	Partially	The DPIA can address the legal requirement by implementing rigorous, user-friendly assessments within its framework, focusing on robust security measures and safeguards specifically tailored for the sensitive nature of special category data and personal data concerning minors, ensuring lawful, compliant, and secure processing by its users in the health and education domains.
LEG-TSD-005	TRUSTEE must be equipped with the technical and governance solutions which will allow for appropriate protection of the data which are or may be considered personal data, to ensure that the rights and freedoms of the data subjects (users or people whose data are accessed between	Partially	This depends on the maturity of the ABBs such as the homomorphic encryption ABB.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	users) are respected, can be exercised and will not be infringed.		
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Partially	This depends on the maturity of development of interfaces and authentication.
LEG-TSD-008	TRUSTEE should have its Terms of Use where it will be clearly indicated what role TRUSTEE system developers and providers assume in the context of the GDPR. The Terms of Use shall be presented to the end users who, if they would like to use tools offered by TRUSTEE to exchange and explore the data which may be considered (encrypted/pseudonymised) personal data, must accept them to be able to do so.	Partially	For DPIA the requirement is partially satisfied as a new user friendly and more informative UI is under development. In the future, the Terms of Use of TRUSTEE should include the information how the DPIA module works.
LEG-TSD-009	In case TRUSTEE system developers decide to engage an external cloud service provider, they shall inform about it the end users and include this information in the Terms of Use of the TRUSTEE system. The data controller (i.e. Data provider / Data consumer), by approving Terms of Use, will have a chance to authorise this sub-processing relationship.	Partially	In case TRUSTEE developers engage an external cloud service they will inform the end users and update the Terms of Use.
LEG-TSD-011	TRUSTEE architecture should allow the data provider to indicate what data can be processed, for how long and for what purpose.	Yes	The DPIA tool gives the data provider control over what data can be processed, for how long and for what purpose. The data provider will describe these aspects in the DPIA and in the STM agreement, asn Consumer will need to adhere to the set provisions.
LEG-TSD-012	With regard to the user data processed on TRUSTEE, TRUSTEE providers should implement solutions, such as technical and organisational measures (such as, in particular, encryption, pseudonymisation, privacy policy, and Terms of Use) which will address the requirements defined in the GDPR principles.	Yes	DPIA assesses the implementation of HE techniques on provided data, as well as other security measures which are made available to the user (such as AM, ATR,
LEG-TSD-013	With regards to the TRUSTEE user's identification data, TRUSTEE providers	Partially	TRUSTEE providers, using the DPIA and CNIL tool, will

	will be controllers and below principles should be addressed by them: • Lawfulness: A legal basis in Article 6 (and where applicable an exception defined in Article 9 (2) GDPR for special categories of data) should be indicated. • Transparency: Information to the data subject (Data provider and Data consumer) about processing of their data by TRUSTEE Developers and maintaining parties should be presented. • Purpose limitation: Specific, explicit and legitimate purposes and process only data for this set purpose shall be defined. • Data Minimisation: Only data which is relevant and strictly necessary to provide the service can be processed. Pseudonymisation and/or anonymisation of data should be considered wherever possible. In the case of TRUSTEE, SSI authentication should allow for protection of users' privacy. Minimisation of the amount of data collected also in the log data and not storing sensitive or critical data must be applied. • Accuracy: Technical and organisational measures shall be put in place to ensure that the data provided by users is up to date and accurate and/or allow the users to correct inaccurate data. • Storage limitation: All data which is no longer necessary for the fulfilment of the purpose shall be deleted or anonymised, unless an exception applies (e.g., scientific research). Retention periods should be associated for each category of data (including logs). • Confidentiality, Integrity and Data Security: Technical and organisation measures, described in the sections below and in the technical requirements shall be implemented to mitigate risks and to ensure a high level of security, confidentiality and integrity of the data		deploy advanced technical solutions like secure transmission methods to ensure confidentiality and integrity of users' identification data. They will integrate user-friendly interfaces to maintain transparency, allowing users to easily access, correct, and understand their data and the processing it undergoes. Advanced data minimization and anonymization techniques will be applied to ensure that only essential data is processed and stored, and rigorous access controls and data lifecycle management systems will be implemented to enforce strict data retention periods and ensure secure, lawful, and transparent processing, thereby maintaining a high level of data security, confidentiality, and integrity in adherence to the GDPR.
LEG-TSD-014	With regards to the data provided onto TRUSTEE by data providers, which may be considered personal, the system should	Partially	CNIL tool has been selected as the candidate technology to perform DPIA of planned

allow to address these principles as well, in particular:

- **Lawfulness:** A legal basis in Article 6 (and where applicable Article 9 GDPR) should be indicated by the data provider as the basis for providing the data onto TRUSTEE and sharing them with the data consumers, and by the Data consumer for further processing for Data consumer's purposes.
- **Transparency:** Data provider and Data consumer should be indicated as responsible for providing data subjects with the information about processing of their data on TRUSTEE and informed that the data consumers are among recipient of the data.
- **Purpose limitation:** Data providers should indicate limitations which are in place in the context of the purposes for which the data provided onto TRUSTEE can be processed.
- **Data Minimisation:** Only data which is relevant to achieve the defined purpose can be processed. Pseudonymisation and/or anonymisation of data should be considered wherever possible. In the case of TRUSTEE, homomorphic encryption is one of the means which should contribute to data providers' and data consumers' compliance with this requirement.
- **Accuracy:** The end users should make sure that the technical and organisational measures shall be put in place on their sides as well as on TRUSTEE allow to keep the data accurate. The users providing the data should confirm whether and how the data provided onto TRUSTEE are accurate, by providing this information in the contract management layer and in the privacy impact assessment.
- **Storage limitation:** All data which is no longer necessary for the fulfilment of the legitimate purpose set out by the Data providers or Data consumers shall be deleted or anonymised, unless an exception applies (e.g., scientific research).
- **Confidentiality, Integrity and Data Security:** Technical and organisation

processing of the data provided via TRUSTEE by Data Providers or accessed via TRUSTEE by Consumers. The CNIL tool enables TRUSTEE to support Data Providers in fulfilling the requirements of "lawfulness", "transparency", and "purpose limitation" by offering structured guidelines and assessments. It aids in identifying and establishing a lawful basis for processing by assessing the necessity and legality of the data processing activities. For "transparency", the tool provides frameworks to clearly communicate the nature, scope, and purpose of data processing to the data subjects, ensuring that they are well-informed about how their data is being used. Regarding "purpose limitation", the CNIL tool assists in defining and documenting the specific, explicit, and legitimate purposes for which the data is being processed and ensures that the processing does not go beyond the set purposes, maintaining strict adherence to GDPR principles.

	measures, described in the sections below and in the technical requirements shall be implemented to mitigate risks to data processing and to ensure a high level of security, confidentiality and integrity of the data.		
LEG-TSD-015	TRUSTEE, as data processor, must allow for responding in a timely manner to data subjects rights by the Data provider (a party sharing the data being personal data with TRUSTEE system providers) to: • Information about the data processing (Article 12-14) • Access (Article 15) • Rectification (Article 16) • Erasure (Article 17) • Restriction on processing (Article 18) • Portability (Article 20) • Object (Article 21) • Not to be subject to a decision based solely on automated processing, including profiling (Article 22)	Partially	DPIA ABB implements a structured framework and processes that enable Data Providers to respond swiftly and accurately to data subjects' requests for information, access, rectification, erasure, restriction on processing, data portability, and objections, and ensures that data subjects are not subjected to decisions based solely on automated processing, maintaining robust compliance with Articles 12-22 of the GDPR.
LEG-TSD-016	TRUSTEE, as data controller of the TRUSTEE user's identification data must respond in a timely manner to data subjects rights to: • Information about the data processing (Article 12-14) • Access (Article 15) • Rectification (Article 16) • Erasure (Article 17) • Restriction on processing (Article 18) • Portability (Article 20) • Object (Article 21) • Not to be subject to a decision based solely on automated processing, including profiling (Article 22) • Lodge a complaint with the supervisory authority (Article 13).	Partially	CNIL tool, can address this requirement by integrating technical solutions such as automated response systems to swiftly manage and respond to data subjects' rights and inquiries, and implementing advanced data management systems that allow for immediate access, rectification, and erasure of data, ensuring seamless compliance with Articles 12-22 and 13 of the GDPR while maintaining transparency and efficiency in processing users' identification data. A
LEG-TSD-017	The information provided to the user of the platform, being a natural person, whose credentials/log in details will be processed on TRUSTEE, shall fulfil the below requirements: • concise, transparent, intelligible and easily accessible • a clear and plain language • in writing including by electronic means • recommendation to	Yes	The DPIA tool follows the CNIL methodology to be accessible and easy to understand.

	use layered privacy statements. No information required if it proves impossible or would involve a disproportionate effort to contact the data subject.		
LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design) and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Yes	The DPIA tool will not be accessible from unauthorized users.
LEG-TSD-021	The measures to be provided and which should be considered in the process of development and maintenance of TRUSTEE are: • Pseudonymisation and encryption of personal data – the TRUSTEE platform will employ homomorphic encryption technique; • Measures to ensure the security of IT employed in the data processing, i.e. ensuring its ongoing confidentiality, integrity, availability and resilience – to answer these requirements, TRUSTEE will employ authentication services, such as SSI and SDK; • Incident response capabilities to restore the availability and access to personal data in a timely manner – due to the federated learning implementation, this responsibility would be rather dependent on the end-users providing datasets via TRUSTEE; • A regular assessment and evaluation of the effectiveness of the implemented technical and organisational security measures. The list is non-exhaustive. The TRUSTEE providers are encouraged to provide additional security measures to ensure appropriate level of data security.	Yes	DPIA is compatible with the HE framework of TRUSTEE and the federated AI platforms for model sharing.
LEG-TSD-022	Appropriate, secure and differential data access management policies must be applied. Authentication should be imposed before any access to personal data.	Partially	Authentication and Access control of DPIA is work in progress and is tightly coupled with the development of the AM ABB. The DPIA has to be conducted before Data provider

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

			provides the data via TRUSTEE and Consumer accesses the data via TRUSTEE.
LEG-TSD-035	When data security is comprised, TRUSTEE system providers must report the violation to the data controllers (Data providers and Data consumers) without undue delay if it is likely to pose a risk to the rights and freedoms of individuals.	Partially	DPIA interfaces with the ATR and STM ABBs to ensure security and trust during the various transactions
LEG-TSD-036	In case of the transfer of the data via TRUSTEE to third (non-EU) countries, appropriate safeguards must be considered and TRUSTEE should enable their implementation by the user.	Partially	Scenario depended. DPIA is developed to assess the risks associated with data sharing to third non-EU countries.
LEG-TSD-037	TRUSTEE should allow the users to conduct a DPIA prior to any data processing activity (i.e. prior to accessing (encrypted) data and conducting computing activities with the use of data).	Yes	The DPIA tool allows the users to perform a DPIA.
LEG-TSD-039	Implement measures into the solutions of TRUSTEE for the protection of personal data in mixed data sets that are to be brought by Data providers.	Yes	DPIA accounts for risks associated with mixed data sets.
LEG-TSD-041	Explore whether any specific addressing of tools/considerations of allowing for the sharing of data/documents held by public sector bodies could be facilitated by TRUSTEE.	Partially	DPIA will be developed to consider risks associated with data held by public sector bodies based on the pilot specific scenarios
LEG-TSD-042	TRUSTEE system developers should identify and take appropriate and proportionate technical and organizational measures to manage risks created by the security systems.	Partially	DPIA's security analysis will be performed after its development and verification of its performance
LEG-TSD-043	The level of security which should be ascertained has to be appropriate to the risks posed, and should take into account the following elements: a. the security of systems and facilities; b. incident handling; c. business continuity management; d. monitoring, auditing and testing; and d. compliance with international standards	Partially	DPIA's security analysis will be performed after its development and verification of its performance
LEG-TSD-045	TRUSTEE system developers must fulfil the obligation of monitoring, auditing and testing, there should be policies implemented which will define	Partially	DPIA interfaces with ATR and STM for monitoring, accountability and auditing.

	supervision, inspection and verification procedures.		
LEG-TSD-046	TRUSTEE providers may be recognised either as essential (as provider of trusted services) or as important (as research institutions) entities, therefore they should adhere, in all phases of the development of TRUSTEE and its further maintenance, to the NIS 2 Directive stemming obligations, which will be further transposed and established at the national level	Partially	User onboarding to DPIA is work in progress tightly coupled with the AM ABB
LEG-TSD-047	TRUSTEE system developers should consider and implement the requirements stemming from the NIS and NIS 2 Directive, to ensure appropriate level of security of information, especially implementation of the means such as: - policies on risk analysis and information system security, - incident handling, - business continuity , - supply chain security, - security in acquisition, development and maintenance of network and information system, - assessment of the effectiveness of implemented measures, - cybersecurity training and basic cyber hygiene practices, - procedures regarding the use of cryptography and, where appropriate, encryption, access control policies, asset management and HR security, - the use of multi-factor authentication or continuous authentication solutions within the entity, where appropriate.	Yes	CNIL tool abides with the NIS 2 Directive
LEG-TSD-048	TRUSTEE system developers shall implement appropriate incident reporting procedures.	Yes	Incident reporting procedures will be implemented.
LEG-TSD-049	TRUSTEE providers shall disclose the terms and conditions that apply to the services provided by them, as well as shall clearly describe the policies governing their provision.	Yes	The DPIA tool will display the required terms and conditions.
LEG-TSD-050	TRUSTEE providers shall thoroughly plan and implement appropriate security measures and constantly monitor, test and audit them, as well as keep the	Partially	DPIA interfaces with ATR and STM for monitoring, accountability and auditing.

	documentation describing those measures in a transparent and clear manner.		
LEG-TSD-103	If TRUSTEE system developers occur to be providers of high risk AI, TRUSTEE system developers must adhere to the requirements for Trustworthy AI: Human Agency and Oversight, Technical Robustness and Safety, Privacy and Data Governance, Transparency, Diversity, Non-discrimination and Fairness, Environmental and Societal Well-Being, Accountability, further described in the legal framework and in socio-ethical requirements section	Partially	Trustworthy AI, explainability, robustness, etc. are relevant risks currently assessed by DPIA, but further development is required based on the availability of other ABBs
LEG-TSD-104	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and providers must cooperate with national authorities upon the receipt of an order to act against illegal content or to provide specific information about service recipients.	Partially	DPIA will provide means of mitigating relevant risks
LEG-TSD-105	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and consortium should designate points of contact both for the recipients of their services and for member states' authorities, the European Commission and the European Board for Digital Services	Yes	The DPIA tool will ensure the designation of contact points.
LEG-TSD-107	TRUSTEE system developers must abstain from designing and operating their interfaces in a manipulative or deceitful manner or in a way hindering service recipients' ability to make independent and informed choices. (This is an additional requirement for providers of online platforms, under the category of hosting services. To the extent that TRUSTEE's activities and service provision falls under this category, this could be applied.)	Partially	The DPIA tool, augmented by the CNIL tool, can implement user-centric design principles and ethical guidelines that ensure the development of transparent, straightforward user interfaces, preventing manipulative or deceitful practices and fostering an environment where service recipients can make independent and informed choices, aligning with the highest standards of ethical conduct and user empowerment.

LEG-TSD-111	The contracts developed by the TRUSTEE system providers should be transparent, taking into account the rights to data and information of all parties involved, respect commercial interests of the parties, ensure undistorted competition, enable data portability as much as possible, define the conditions for access to the data, possibility and limitations concerning re-use of the data, indicate technical measures for data access and sharing, clarify liability towards the parties and indicate parties' rights.	Yes	CNIL assess such aspects and related risks and proposes mitigation actions
--------------------	--	-----	--

12.3. SSI-HE

Table 75: SSI-HE - Legal requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
LEG-USR-016	When a data subject exercises its right of erasure and/or restriction to the processing, Data Provider must ensure to remove the data from TRUSTEE / stop sharing the data of data subjects when requested by a data subject, required by the data protection laws and when there is no legal basis for further sharing and processing of the data.	Partially	Credentials can be removed from the Aries Agent
LEG-USR-024	Data Provider and Consumer must use the technical and organisational measures to protect personal data. Data provider and Consumer should use measures offered by TRUSTEE (Homomorphic encryption, SSI, authentication, federated learning, etc.).	Partially	To support the users in responding to this requirement, this ABB provides technical measures based on HE and SSI solutions which allow for provision only of the necessary metadata and their encryption
LEG-TSD-002	TRUSTEE must have the fundamental rights principles at its core, by ensuring that the privacy of the individuals is respected and protected.	Yes	SSI-HE does not deal with any personal data of the user (only metadata). The only data about the user SSI-HE obtains are Data Provider ID and endpoints information.
LEG-TSD-005	TRUSTEE must be equipped with the technical and governance solutions which will allow for appropriate protection of the data which are or may	Yes	SSI-HE addresses it by the fact that it only deals with meta data and does not allow for access to personal data / the

	be considered personal data, to ensure that the rights and freedoms of the data subjects (users or people whose data are accessed between users) are respected, can be exercised and will not be infringed.		specific information of the data provider.
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Partially	SSI-HE is based on the state of the art Open FHE and Hyperledger, which are well-established information security solutions. The aspects concerning authentication are to be addressed by another component (AM).
LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design) and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Yes	SSI-HE does not deal with personal data provided by Data provider (only metadata). It eliminates the access of the Consumer to data, but provides access only to metadata, following privacy by design and by default principles.
LEG-TSD-021	The measures to be provided and which should be considered in the process of development and maintenance of TRUSTEE are: • Pseudonymisation and encryption of personal data – the TRUSTEE platform will employ homomorphic encryption technique; • Measures to ensure the security of IT employed in the data processing, i.e. ensuring its ongoing confidentiality, integrity, availability and resilience – to answer these requirements, TRUSTEE will employ authentication services, such as SSI and SDK; • Incident response capabilities to restore the availability and access to personal data in a timely manner – due to the federated learning implementation, this responsibility would be rather dependent on the end-users providing datasets via TRUSTEE; • A regular assessment and evaluation of the effectiveness of the implemented technical and organisational security	Partially	SSI-HE provides encryption of the metadata of the data set, and in other terms, with regards to other technical security measures, this component will follow the rules defined in TRUSTEE.

	measures. The list is non-exhaustive. The TRUSTEE providers are encouraged to provide additional security measures to ensure appropriate level of data security.		
LEG-TSD-028	When using free or open source software, TRUSTEE system developers should choose solutions that have an active community, good documentation and are regularly updated.	Yes	OpenFHE & Hyperledger Aries are well-known and established security solutions used in SSI-HE.
LEG-TSD-031	Implementation of cryptographic algorithms and protocols by TRUSTEE system developers by their own should be avoided, and choosing cryptographic libraries which are maintained, easy to use and recognised should be encouraged.	Partially	Search Algorithm has been implemented but using existing OpenFHE framework .
LEG-TSD-034	Applications must be tested (development test, security test). Real/production data should not be used during the development and testing phase, as using personal data for testing purposes may go beyond their initial purpose.	Yes	The test carried out are done with sample data provided by the UCs.
LEG-TSD-056	Access information/material and credentials must be possible to be withdrawn or rotation.	Partially	Credentials can be removed from the Aries Agent.
LEG-TSD-060	Hosting model must be selected (provider-offered service / customer-managed on-premise hosting / provider-managed cloud hosting / customer-managed cloud hosting)	Yes	This is addressed in the following manner: Issuer: TECNALIA on-premise hosting Holder: Data provider-offered service Verifier: TRUSTEE on-premise hosting
LEG-TSD-064	If TRUSTEE is considered data intermediation service, TRUSTEE system providers must further determine whether TRUSTEE will be an intermediation service between data holders and potential data users, including making available the technical or other means to enable such services, that would include bilateral or multilateral exchanges of data or the creation of platform enabling the	Partially	To be further investigated at a later stage. SSI-HE does not allow for access to datasets, only metadata and processes only metadata, and by this contributes to creation of the means for exchange of data and interconnection of the users.

	exchange or joint use of data, as well as the establishment of other specific infrastructure for the interconnection of data holders (Data providers) with data users (Data consumers).		
LEG-TSD-075	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers must apply measures preventing transfer of or access to non-personal data that is unlawful under EU or national law.	Partially	To be further investigated at a later stage. SSI-HE does not allow for access or transfer of the data, only metadata and by this could support addressing this requirement.
LEG-TSD-107	TRUSTEE system developers must abstain from designing and operating their interfaces in a manipulative or deceitful manner or in a way hindering service recipients' ability to make independent and informed choices. (This is an additional requirement for providers of online platforms, under the category of hosting services. To the extent that TRUSTEE's activities and service provision falls under this category, this could be applied.)	Partially	To be further investigated at a later stage. SSI-HE allows the selection of the dataset based on the information about the dataset included in metadata.
LEG-TSD-109	TRUSTEE system developers should use only such blockchain solutions which allow to address GDPR principles, especially data minimisation, storage limitation and where necessary, the ones which will allow deletion of the data, or full anonymisation of the data that are not necessary or which erasure was requested by a data subject or a Data provider.	Partially	Data recorded in the Blockchain is minimised by the SSI Technology – only DID ans schemas are stored.
LEG-TSD-110	When personal data are stored on an immutable blockchain, the TRUSTEE system developers should ensure one of the points a-d: a) the data are either not or will no longer constitute personal data, b) that data subject (user) is considered to be the controllers of their data, c) that the household exemption defined in the GDPR applies or d) a justification for constant storage of data on the blockchain has to exist.	Yes	No personal data is recorded by SSI-HE, only meta data are processed (but not recorded). Therefore, other components should take care of this requirement. The application of this requirement to the SSI-HE is to be reviewed and reflected in D2.2.

12.4. DL

Table 76: DL - Legal requirements

Req id	Req. Description	Compliance	How is addressed
LEG-TSD-001	Should the TRUSTEE system developers and parties maintaining the system partners at any point handle or process any type of personal data, they would be expected to balance their legitimate interest with the fundamental rights and freedoms of the data subjects whose data they will be processing.	Yes	No personal data will be handled by DL, as the DL will not have access to any raw data
LEG-TSD-002	TRUSTEE system must have the fundamental rights principles at its core, by ensuring that the privacy of the individuals is respected and protected.	Yes	The DL will not be accessible to unauthorized users.
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Partially	In the future DL implementation phase, the DL will follow comprehensive security measures, encompassing state-of-the-art protocols and technologies
LEG-TSD-007	Security solutions recommended to be considered: - Backups, if possible encrypted and checked regularly should be made, to enable restoring the system. - Limitation of the size of the software stack used, - Installation of critical updates without delay, - Automation of a vulnerability watch, - Application of vulnerability detection tools, - Restriction of physical and software access to diagnostic and remote configuration ports, - Protection of the databases made available online, - Secure database management.	Partially	In the future DL implementation phase, the DL will consider these security recommendations.
LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design), and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Yes	The DL will not be accessible to unauthorized users, and it will not store any raw data.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

LEG-TSD-023	Logging systems should be used to detect anomalies or security events.	Partially	The future implementation of DL aims to generate logs to detect abnormalities or security events, whereas the ATR will store DL transactions.
LEG-TSD-024	Penetration testing should be considered and performed.	Partially	The future implementation of DL aims to perform penetration testing during the APIs communications between ABBs.
LEG-TSD-025	Efficient version control system should be set up, to track different versions of the code over the time, with appropriate authentication and permission system and regular backups, as well as efficient development system.	Yes	The DL is being developed using the GIT version control system.
LEG-TSD-026	Code quality metrics tools should be considered.	Yes	DL will consider the use of code quality metrics tools
LEG-TSD-027	Control of the use libraries and SDKs must be applied.	Yes	DL will investigate applying control on the user libraries and SDKs
LEG-TSD-028	When using free or open source software, TRUSTEE system developers should choose solutions that have an active community, good documentation and are regularly updated.	Yes	DL will investigate the maintainability of solutions with commercial support.
LEG-TSD-032	Appropriate maintenance of libraries and SDKs should be considered, e.g. using dependency management system, managing updates to the dependencies, monitoring the versions of libraries and SDKs and regularly check the open source projects and their status, especially in the context of the change of domain or package ownership.	Yes	DL will investigate and consider the maintainability of libraries and SDKs.
LEG-TSD-033	Quality and security of the code and its documentation, as well as the documentation of the system architecture must be ensured.	Yes	DL will record analytical documentation of the architecture.
LEG-TSD-034	Applications must be tested (development test, security test). Real/ production data should not be used during the development and testing phase, as using personal data for testing purposes may go beyond their initial purpose.	Yes	All testings are carried out using sample / dummy data.
LEG-TSD-035	When data security is compromised, TRUSTEE system providers must report the violation to the data controllers (Data Providers and Data	Partially	DL could report any violation threats to the ATR through their communication.

	Consumers) without undue delay if it is likely to pose a risk to the rights and freedoms of individuals.		
LEG-TSD-043	The level of security which should be ascertained has to be appropriate to the risks posed, and should take into account the following elements: a. the security of systems and facilities; b. incident handling; c. business continuity management; d. monitoring, auditing and testing; and d. compliance with international standards.	Partially	DL's security analysis will be performed after its development and verification of its performance
LEG-TSD-044	The security measures taken by TRUSTEE system developers shall include: (a) detection processes and procedures maintained and tested to ensure timely and adequate awareness of anomalous events; (b) processes and policies on reporting incidents and identifying weaknesses and vulnerabilities in their information systems; (c) a response in accordance with established procedures and reporting the results of the measure taken; (d) an assessment of the incident's severity, documenting knowledge from incident analysis and collection of relevant information which may serve as evidence and support a continuous improvement process.	Partially	DL will consider these security measures during the development phase.
LEG-TSD-048	TRUSTEE system developers shall implement appropriate incident reporting procedures.	Yes	Incident reporting procedures will be implemented within DL.
LEG-TSD-050	TRUSTEE providers shall thoroughly plan and implement appropriate security measures and constantly monitor, test and audit them, as well as keep the documentation describing those measures in a transparent and clear manner.	Partially	Such measures will be investigated during the DL development.
LEG-TSD-073	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers must ensure continuity of the services , and where storage of data is included, allow parties using services to access, transfer or retrieve their data.	Partially	The DL will make available any information / content that is stored within the component to the TRUSTEE users. No raw data is stored within the DL.
LEG-TSD-074	If TRUSTEE happens to be intermediation service according to the	Yes	Interoperability is ensured through the usage of operation

	DGA, TRUSTEE service providers must ensure interoperability by adhering to common open standards in the sectors where service operates.		and commons standards such as semantics and semantic annotations.
LEG-TSD-075	If TRUSTEE happens to be intermediation service according to the DGA , TRUSTEE service providers must apply measures preventing transfer of or access to non-personal data that is unlawful under EU or national law.	Yes	The DL does not store any raw data.
LEG-TSD-077	If TRUSTEE happens to be intermediation service according to the DGA , TRUSTEE service providers must take necessary measures to ensure an appropriate level of security for the (...) processing and transmission of non-personal data and the highest level of security for transmission of competitively sensitive information (e.g. trade secrets).	Yes	The DL does not store any raw data.
LEG-USR-016	When a data subject exercises its right of erasure and/or restriction to the processing, Data Provider must ensure to remove the data from TRUSTEE / stop sharing the data of data subjects when requested by a data subject, required by the data protection laws and when there is no legal basis for further sharing and processing of the data.	Partially	A validity period for new dataset registrations can be provided by the Data Provider, to prevent data processing after the expiration date.
LEG-USR-024	Data Provider and Data Consumer must use the technical and organisational measures to protect personal data. Data provider and data consumer should use measures offered by TRUSTEE (Homomorphic encryption, SSI, authentication, federated learning, etc.).	Yes	No storage of raw data.
LEG-USR-030	Data provider may indicate that the data can be provided only to the Data consumers from the countries with an adequate level of protection, as deemed by the European Commission.	Partially	Data Providers could indicate specific countries as metadata values.

12.5. STM

Table 77: STM - Legal requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
LEG-TSD-001	Should the TRUSTEE system developers and parties maintaining the system at any point handle or process any type of personal data, they should balance their legitimate interest with the fundamental rights and freedoms of the data subjects whose data they will be processing.	Yes	The STM only collects minimal necessary data.
LEG-TSD-003	TRUSTEE must be developed as the system ensuring appropriate and secure processing of personal data. This is relevant especially for: a) the data considered as “special category data”, such as data concerning health or genetic data, as well as b) personal data concerning minors. These data will be processed by the future users of TRUSTEE (Data providers, Data consumers), especially in the HEALTH and EDUCATION domain.	Partially	The STM only enables data sharing and data processing after a Data Provider and a Data Consumer have agreed on the conditions of processing a dataset.
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Yes	The STM is being iteratively developed and improved with the basic state-of-the-art security rules in mind.
LEG-TSD-007	Security solutions recommended to be considered: - Backups, if possible encrypted and checked regularly should be made, to enable restoring the system. - Limitation of the size of the software stack used, - Installation of critical updates without delay, - Automation of a vulnerability watch, - Application of vulnerability detection tools, - Restriction of physical and software access to diagnostic and remote configuration ports,	Partially	The STM’s development currently considers the limitation of the size of software stack, installation of critical updates without delay, and utilizes static application security testing tools.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	- Protection of the databases made available online, - Secure database management.		
LEG-TSD-008	TRUSTEE should have its Terms of Use where it will be clearly indicated what role TRUSTEE system developers and providers assume in the context of the GDPR. The Terms of Use shall be presented to the end users who, if they would like to use tools offered by TRUSTEE to exchange and explore the data which may be considered (encrypted/pseudonymised) personal data, must accept them to be able to do so.	Partially	The STM contains an agreement manager which is being investigated to store the acceptance/denial of the Terms of Use for each user. The STM will be further developed if needed.
LEG-TSD-009	In case TRUSTEE system developers decide to engage an external cloud service provider, they shall inform about it the end users and include this information in the Terms of Use of the TRUSTEE system. The data controller (i.e. Data provider / Data consumer), by approving Terms of Use, will have a chance to authorise this sub-processing relationship.	Partially	The STM and the STM's stored data currently does not engage with external cloud service providers. The storing of a Terms of Use within the STM is being considered.
LEG-TSD-011	TRUSTEE architecture should allow the data provider to indicate what data can be processed, for how long and for what purpose.	Yes	The STM provides an agreement creation functionality for the user to add conditions to the processing of their datasets.
LEG-TSD-012	With regard to the user data processed on TRUSTEE, TRUSTEE providers should implement solutions, such as technical and organisational measures (such as, in particular, encryption, pseudonymisation, privacy policy, and Terms of Use) which will address the requirements defined in the GDPR principles.	Partially	The STM will always encrypt data in transit and it is being considered to provide a Terms of Use when a user launches the STM for the first time. Further encryption of the STM's GraphDB will be investigated.
LEG-TSD-013	With regards to the TRUSTEE user's identification data, TRUSTEE providers will be controllers and below principles should be addressed by them: • Lawfulness: A legal basis in Article 6 (and where applicable an exception defined in Article 9 (2))	Partially	The STM currently satisfies the 'Data Minimisation' and 'Purpose limitation' points. 'Data minimisation' is satisfied by only collecting and processing and storing

	GDPR for special categories of data) should be indicated. • Transparency: Information to the data subject (Data provider and Data consumer) about processing of their data by TRUSTEE Developers and maintaining parties should be presented. • Purpose limitation: Specific, explicit and legitimate purposes and process only data for this set purpose shall be defined. • Data Minimisation: Only data which is relevant and strictly necessary to provide the service can be processed. Pseudonymisation and/or anonymisation of data should be considered wherever possible. In the case of TRUSTEE, SSI authentication should allow for protection of users' privacy. Minimisation of the amount of data collected also in the log data and not storing sensitive or critical data must be applied. • Accuracy: Technical and organisational measures shall be put in place to ensure that the data provided by users is up to date and accurate and/or allow the users to correct inaccurate data. • Storage limitation: All data which is no longer necessary for the fulfilment of the purpose shall be deleted or anonymised, unless an exception applies (e.g., scientific research). Retention periods should be associated for each category of data (including logs). • Confidentiality, Integrity and Data Security: Technical and organisation measures, described in the sections below and in the technical requirements shall be implemented to mitigate risks and to ensure a high level of security, confidentiality and integrity of the data.		the necessary information within the STM. 'Purpose limitation' is achieved by allowing the Data Provider define for what purposes their datasets may be processed for. The STM will be further developed in the next iterations to address the rest of the points.
LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design), and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Partially	The STM is being planned and continuously developed and improved to use as little amount of personal data as possible in order to function correctly and enable users to

			create agreements with each other. No user is able to see another user's personal information without legitimate reason.
LEG-TSD-023	Logging systems should be used to detect anomalies or security events.	Partially	The STM will log all dataset agreement signature events with the ATR after the integration of both ABBs is complete. Currently, the dataset agreement signatures are stored on the STM's GraphDB.
LEG-TSD-025	Efficient version control system should be set up, to track different versions of the code over the time, with appropriate authentication and permission system and regular backups, as well as efficient development system.	Yes	The STM is being developed using the GIT version control system.
LEG-TSD-026	Code quality metrics tools should be considered.	Yes	The development of the STM is supported by static application security testing tools.
LEG-TSD-027	Control of the use libraries and SDKs should be applied.	Partially	All used libraries used for the development of the STM are checked for licensing compliance.
LEG-TSD-028	When using free or open source software, TRUSTEE system developers should choose solutions that have an active community, good documentation and are regularly updated.	Yes	The STM is being developed with open source software, libraries and tools that have active communities (e.g., Flutter and Java).
LEG-TSD-029	When using types of solutions with commercial support, TRUSTEE system providers should contractually ensure maintenance and updates of the code during the whole lifecycle of the project.	Partially	Currently, the STM does not use any solutions with commercial support.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

LEG-TSD-031	Implementation of cryptographic algorithms and protocols by TRUSTEE system developers by their own should be avoided, and choosing cryptographic libraries which are maintained, easy to use and recognised should be encouraged.	Yes	Every encryption is supported by an open source library.
LEG-TSD-032	Appropriate maintenance of libraries and SDKs should be considered, e.g. using dependency management system, managing updates to the dependencies, monitoring the versions of libraries and SDKs and regularly check the open source projects and their status, especially in the context of the change of domain or package ownership.	Yes	The STM uses maven and pub to manage the libraries.
LEG-TSD-033	Quality and security of the code and its documentation, as well as the documentation of the system architecture should be ensured.	Yes	The STM's code is regularly scanned using with static application security testing tools.
LEG-TSD-034	Applications must be tested (development test, security test). Real/ production data should not be used during the development and testing phase, as using personal data for testing purposes may go beyond their initial purpose.	Partially	At this stage, the STM's code includes unit tests for development testing.
LEG-TSD-037	TRUSTEE should allow the users to conduct a DPIA prior to any data processing activity (i.e. prior to accessing (encrypted) data and conducting computing activities with the use of data).	Yes	As soon as integration with the DPIA module is possible, the STM will redirect the user to conduct a DPIA before enabling the processing of a dataset.
LEG-TSD-049	TRUSTEE providers shall disclose the terms and conditions that apply to the services provided by them, as well as shall clearly describe the policies governing their provision.	Partially	The STM will show a Terms of Use to which the user must agree to before continuing.
LEG-TSD-107	TRUSTEE system developers must abstain from designing and operating their interfaces in a manipulative or deceitful manner or in a way hindering service recipients' ability to make independent and informed choices. (This is an	Yes	The STM is not developed, and not designed, in a manipulative or deceitful manner.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	additional requirement for providers of online platforms, under the category of hosting services. To the extent that TRUSTEE's activities and service provision falls under this category, this could be applied.)		
LEG-TSD-112	TRUSTEE should allow for concluding agreements (e.g. confidentiality agreements) where Data provider could indicate limitations for data processing stemming from the intellectual property rights and trade secrets regulations.	Yes	The STM's agreement creation page will allow the user to state their own conditions for the dataset that will be processed by other users.
LEG-TSD-113	TRUSTEE should allow for concluding (licensing) agreements where Model Provider could indicate terms and limitations to the use of AI/ML models provided, on which they assert control stemming from intellectual property rights and/or contracts.	Yes	The STM's agreement creation page will allow the user to state their own conditions for the AI model that will be used by other users.
LEG-USR-001	The Data Provider / Consumer / Model Provider providing the data via TRUSTEE needs to consider whether the data are personal or not and indicate it in the STM and in DPIA phase.	Yes	The STM will enable the providers to indicate in the agreement if a dataset contains personal data and if an AI model may be used with personal data.
LEG-USR-002	The Data Provider / Consumer must indicate appropriate legal basis for processing and sharing the data via TRUSTEE. It is not to be decided by the TRUSTEE system providers.	Yes	The agreements contained in the STM will allow the users to indicate the appropriate legal basis for processing and sharing data.
LEG-USR-005	The Data Provider / Consumer, where considered as necessary, must sign a Data Processing Agreement with TRUSTEE system providers as data processors (when real data are processed).	Partially	The Terms of Use that will be shown on the first launch of the STM could be used as an agreement between the users and the TRUSTEE system providers.
LEG-USR-007	Data Provider must indicate for what purposes data can be processed by the Consumers. Moreover, the Data Provider and Consumer should include in the	Yes	The agreement creation page in the STM allows the Data Provider to define their

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	data sharing agreement the following information: a) the parties identification and information on the participants of data sharing, b) the purpose of data sharing defined in a clear manner, i.e. description of the aims for which data are shared, why sharing is necessary to achieve those goals and what benefits to individuals and/or to society the parties hope to bring, c) what data are going to be shared d) lawful basis for data sharing (both parties need to clarify it for themselves – Data Providers the basis for making data available, and Consumers for accessing the data) e) information if there is any type of special categories of data that would be shared f) procedures allowing for achieving compliance with the rights of individuals (data subjects) g) governance arrangements between the parties		conditions and add all necessary information to each dataset separately.
LEG- USR-027	All actors should be able to access only the data they actually need.	Yes	The STM is developed using the Privacy-by-Design methodology, giving access to only necessary information.
LEG- USR-028	Data Provider and Consumer should always take into account the risk of re-identification of a data subject and should not attempt to re-identify a data subject when it is not necessary or forbidden by the Data Provider. Appropriate statements referring to that should be included in the agreements.	Partially	In future iterations, the Data Provider may use the agreement creator from the STM to fulfil this requirement.
LEG- USR-029	Consumer should adhere to top confidentiality standards required by Data Provider.	Partially	In future iterations, the Data Provider may use the agreement creator from the STM to fulfil this requirement.
LEG- USR-030	Data Provider may indicate to what countries the data can be made available.	Partially	In future iterations, the Data Provider may use the agreement creator from the STM to fulfil this requirement.

LEG-USR-031	Data Provider may indicate that the data can be provided only to the Consumers from the countries with an adequate level of protection, as deemed by the European Commission.	Partially	In future iterations, the Data Provider may use the agreement creator from the STM to fulfil this requirement.
LEG-USR-052	Data Provider and Consumer must keep the record of the DPIA.	Partially	After integration with the DPIA ABB, the STM will have a unique ID of the users DPIA. This unique ID could then be used in combination with the agreement ID to link the agreement and the DPIA within the ATR (and possibly the STM).

12.6. ONESS

Table 78: OneSS – Legal requirements

Req id	Req. Description	Compliance	How is addressed
LEG-TSD-001	Should the TRUSTEE system developers and parties maintaining the system at any point handle or process any type of personal data, they should balance their legitimate interest with the fundamental rights and freedoms of the data subjects whose data they will be processing.	Yes	OneSS will only handle the data necessary for the component's functionalities. No data will be processed without legitimate interest. Any potential transfer of personal data via external connectors will be examined in the future as the connectors will develop.
LEG-TSD-002	TRUSTEE must have the fundamental rights principles at its core, by ensuring that the privacy of the individuals is respected and protected.	Yes	OneSS will not be accessible to unauthorised users.
LEG-TSD-006	Any website, application or server must incorporate basic state-of-the-art security rules, not only on network communications but also on authentication and infrastructure.	Yes	OneSS will ensure secure communications, authorisation and authentication mechanisms.
LEG-TSD-007	Security solutions recommended to be considered: - Backups, if possible encrypted and checked regularly should be made, to enable restoring the	Yes	Stated requirement will be ensured by OneSS. Proper technologies and best practices

	system. - Limitation of the size of the software stack used, - Installation of critical updates without delay, - Automation of a vulnerability watch, - Application of vulnerability detection tools, - Restriction of physical and software access to diagnostic and remote configuration ports, - Protection of the databases made available online, - Secure database management.		will be used in design and implementation of the OneSS.
LEG-TSD-018	TRUSTEE system developers should represent and describe how the products work, with diagrams presenting data flows and description of processes carried out, in a manner understandable for the users.	Yes	The relevant information on the data flows and description of processes is available via TRUSTEE Developer Handbook. An initial version of the content reflects the maturity of the available results and will be updated and improved throughout the project. By updating and expanding content according to the user feedback, it will additionally be assured that the content is presented in a high quality way and in a manner understandable for users.
LEG-TSD-019	TRUSTEE must incorporate the solutions compliant with data protection requirements already in planning and development phases (privacy by design), and ensure that only personal data necessary for the purpose of providing TRUSTEE to the users are processed (privacy by default).	Yes	OneSS will be compliant with data protection requirements.
LEG-TSD-021	The measures to be provided and which should be considered in the process of development and maintenance of TRUSTEE are: • Pseudonymisation and encryption of personal data – the TRUSTEE platform will employ homomorphic encryption technique; • Measures to ensure the security of IT employed in the data processing, i.e. ensuring its	Partially	OneSS will employ all security measures needed to ensure appropriate level of data security. More information will be provided as the development will progress. OneSS development will comply with project-level development processes.

	ongoing confidentiality, integrity, availability and resilience – to answer these requirements, TRUSTEE will employ authentication services, such as SSI and SDK; • Incident response capabilities to restore the availability and access to personal data in a timely manner – due to the federated learning implementation, this responsibility would be rather dependent on the end-users providing datasets via TRUSTEE; • A regular assessment and evaluation of the effectiveness of the implemented technical and organisational security measures. The list is non-exhaustive. The TRUSTEE providers are encouraged to provide additional security measures to ensure appropriate level of data security.		
LEG-TSD-023	Logging systems should be used to detect anomalies or security events.	Partially	The OneSS will have logging enabled where applicable to detect relevant events including security.
LEG-TSD-024	Penetration testing should be considered and performed.	Partially	This requirement will be implemented in accordance with the general procedures of TRUSTEE platform development and testing.
LEG-TSD-025	Efficient version control system should be set up, to track different versions of the code over the time, with appropriate authentication and permission system and regular backups, as well as efficient development system.	Yes	OneSS uses the version control system in accordance with the general procedures of TRUSTEE platform development.
LEG-TSD-026	Code quality metrics tools should be considered.	Yes	Stated tools will be considered and used where applicable in the development of the OneSS.
LEG-TSD-027	Control of the use libraries and SDKs should be applied.	Yes	Stated control will be used where applicable in the development of the OneSS.
LEG-TSD-028	When using free or open source software, TRUSTEE system developers should choose solutions that have an active community, good	Yes	OneSS chooses solutions that have an active community, good documentation and are regularly updated.

	documentation and are regularly updated.		
LEG-TSD-029	When using types of solutions with commercial support, TRUSTEE system providers should contractually ensure maintenance and updates of the code during the whole lifecycle of the project.	Yes	No solutions with commercial support are planned to be used for OneSS development.
LEG-TSD-030	Change of the default configurations, audit, mapping of the dependencies of the SDKs and third party libraries should be considered.	Partially	OneSS has considered the stated in the initial phase of the development, but as the development of the SDK and external connector will progress, more information will be provided in the next versions of this deliverable. OneSS development will comply with the project-wide processes.
LEG-TSD-031	Implementation of cryptographic algorithms and protocols by TRUSTEE system developers by their own should be avoided, and choosing cryptographic libraries which are maintained, easy to use and recognised should be encouraged.	Yes	Cryptographic libraries are not used for the development of the OneSS.
LEG-TSD-032	Appropriate maintenance of libraries and SDKs should be considered, e.g. using dependency management system, managing updates to the dependencies, monitoring the versions of libraries and SDKs and regularly check the open source projects and their status, especially in the context of the change of domain or package ownership.	Yes	OneSS development will be compliant with the general procedures of TRUSTEE platform development and terms of usage.
LEG-TSD-033	Quality and security of the code and its documentation, as well as the documentation of the system architecture should be ensured.	Yes	The documentation of the OneSS will be available via the Handbook and project deliverables. Quality and security of the code is taken into account in the development process.
LEG-TSD-034	Applications must be tested (development test, security test). Real/production data should not be used during the development and testing	Yes	The tests will be conducted without using the personal data.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	phase, as using personal data for testing purposes may go beyond their initial purpose.		
LEG-TSD-035	When data security is comprised, TRUSTEE system providers must report the violation to the data controllers (Data providers and Data consumers) without undue delay if it is likely to pose a risk to the rights and freedoms of individuals.	Yes	OneSS will comply with the project-wide procedures in terms of reporting violations.
LEG-TSD-036	In case of the transfer of the data via TRUSTEE to third (non-EU) countries, appropriate safeguards must be considered and TRUSTEE should enable their implementation by the user.	Yes	OneSS will comply with this requirement and apply the appropriate safeguards where applicable.
LEG-TSD-041	Explore whether any specific addressing of tools/considerations of allowing for the sharing of data/documents held by public sector bodies could be facilitated by TRUSTEE.	Yes	OneSS will be developed taking into account risk management and support the sharing of data held by public sector bodies.
LEG-TSD-042	TRUSTEE system developers should identify and take appropriate and proportionate technical and organizational measures to manage risks created by the security systems.	Yes	OneSS will be assessed in terms of security and taking into account risk management, as the development of OneSS will progress.
LEG-TSD-043	The level of security which should be ascertained has to be appropriate to the risks posed, and should take into account the following elements: a. the security of systems and facilities; b. incident handling; c. business continuity management; d. monitoring, auditing and testing; and d. compliance with international standards.	Yes	OneSS will be assessed in terms of security and taking into account risk management, as the development of OneSS will progress.
LEG-TSD-044	The security measures taken by TRUSTEE system developers shall include: (a) detection processes and procedures maintained and tested to ensure timely and adequate awareness of anomalous events; (b) processes and policies on reporting incidents and identifying weaknesses and vulnerabilities in their information	Yes	OneSS, as a part of the TRUSTEE platform, will comply with the security measures taken by TRUSTEE system developers.

	systems; (c) a response in accordance with established procedures and reporting the results of the measure taken; (d) an assessment of the incident's severity, documenting knowledge from incident analysis and collection of relevant information which may serve as evidence and support a continuous improvement process.		
LEG-TSD-045	TRUSTEE system developers must fulfil the obligation of monitoring, auditing and testing, there should be policies implemented which will define supervision, inspection and verification procedures.	Yes	OneSS will comply with the project-wide procedures and processes.
LEG-TSD-046	TRUSTEE providers may be recognised either as essential (as provider of trusted services) or as important (as research institutions) entities, therefore they should adhere, in all phases of the development of TRUSTEE and its further maintenance, to the NIS 2 Directive stemming obligations, which will be further transposed and established at the national level.	Partially	OneSS will implement certain security measure, however, it is yet to be analysed if it is fully compliant with the NIS and NIS 2 Directives.
LEG-TSD-047	TRUSTEE system developers should consider and implement the requirements stemming from the NIS and NIS 2 Directive, to ensure appropriate level of security of information, especially implementation of the means such as: - policies on risk analysis and information system security, - incident handling, - business continuity , - supply chain security, - security in acquisition, development and maintenance of network and information system, - assessment of the effectiveness of implemented measures, - cybersecurity training and basic cyber hygiene practices, - procedures regarding the use of cryptography and, where appropriate,	Partially	OneSS will implement certain security measure, however, it is yet to be analysed if it is fully compliant with the NIS and NIS 2 Directives.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

	encryption, access control policies, asset management and HR security, - the use of multi-factor authentication or continuous authentication solutions within the entity, where appropriate.		
LEG-TSD-048	TRUSTEE system developers shall implement appropriate incident reporting procedures	Yes	OneSS will investigate reporting procedures related to functionalities covered by this component.
LEG-TSD-049	TRUSTEE providers shall disclose the terms and conditions that apply to the services provided by them, as well as shall clearly describe the policies governing their provision.	Yes	OneSS will comply with the project-wide procedures.
LEG-TSD-050	TRUSTEE providers shall thoroughly plan and implement appropriate security measures and constantly monitor, test and audit them, as well as keep the documentation describing those measures in a transparent and clear manner.	Yes	OneSS plans and will implement appropriate security measures, as well as provide related documentation.
LEG-TSD-067	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers must not make the use of TRUSTEE dependent on the use of their other services.	Yes	OneSS will not make the use of TRUSTEE dependent on the use of their other services.
LEG-TSD-070	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers may offer additional services to data holders or data subjects that facilitate the exchange of data, such as temporary storage, curation, conversion, anonymisation and pseudonymisation; such tools should be used only at the explicit request or approval of the Data Provider (data holder) or data subject and third-party tools offered in that context should not be used for other purposes.	Yes	OneSS will facilitate the exchange of data.
LEG-TSD-076	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers must apply measures	Partially	OneSS, as a part of the TRUSTEE platform, will indirectly support such measures.

	preventing transfer of or access to non-personal data that is unlawful under EU or national law.		
LEG-TSD-077	If TRUSTEE happens to be intermediation service according to the DGA, TRUSTEE service providers must take necessary measures to ensure an appropriate level of security for the (...) processing and transmission of non-personal data and the highest level of security for transmission of competitively sensitive information (e.g. trade secrets).	Partially	OneSS, as a part of the TRUSTEE platform, will indirectly support such measures.
LEG-TSD-083	If TRUSTEE becomes a data processing service that is not free of charge and does not operate on a trial basis, TRUSTEE shall make open interfaces publicly available and free of charge and ensure compatibility with open interoperability specifications of European standards for interoperability. If such standards do not exist, the data should be provided in a structured, commonly used and machine-readable format.	Yes	In case of stated development, OneSS will adhere to open interoperability specifications.
LEG-TSD-085	TRUSTEE system providers (as parties offering data services to other participants) shall ensure interoperability, present at the level of data, data sharing mechanisms and services (including the technical means to do so): a) the dataset content, use restrictions, licences, data collection methodology, data quality and uncertainty shall be sufficiently described in a machine-readable format to allow the recipient to find, access and use the data; b) the data structures, data formats, vocabularies, classification schemes, taxonomies and code lists shall be described in a publicly available and consistent manner; c) the technical means to access the data, such as application programming	Partially	OneSS will provide the means to access the data, such as APIs, and this will be further addressed in the following period with the progress of connectors development.

	interfaces, and their terms of use and quality of service shall be sufficiently described to enable automatic access and transmission of data between parties, including continuously or in real-time in a machine-readable format; d) the means to enable the interoperability of contracts for data sharing within their services and activities shall be provided.		
LEG-TSD-097	If TRUSTEE system developers occur to be providers of high risk AI, a technical documentation of the high risk AI system must be provided before it is placed on the market or put into service, and regularly updated.	Yes	OneSS supports the compliance with this requirement since it can provide the instructions of procedure for developers in case of high risk AI.
LEG-TSD-099	If TRUSTEE system developers occur to be providers of high risk AI, shall ensure appropriate level of accuracy, robustness, cybersecurity. The levels of accuracy and accuracy metrics shall be declared in the instructions of use. The AI systems must be adaptable against errors, faults and inconsistencies that may occur within the system or the environment in which it operates and withstand any attempts by unauthorised parties to alter the use or performance of the AI system.	Partially	OneSS supports the compliance with this requirement since it can provide the instructions of use declaring stated metrics in case of high risk AI.
LEG-TSD-104	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and providers must cooperate with national authorities upon the receipt of an order to act against illegal content or to provide specific information about service recipients.	Yes	In case of stated development, OneSS, as a part of TRUSTEE platform, will comply with the project-wide processes.
LEG-TSD-105	If TRUSTEE is considered as an online intermediary service, TRUSTEE system developers and consortium should designate points of contact both for the recipients of their services and for member states' authorities, the European Commission	Yes	In case of stated development, OneSS, as a part of TRUSTEE platform, will comply with the project-wide processes.

	and the European Board for Digital Services		
LEG-USR-016	When a data subject exercises its right of erasure and/or restriction to the processing, Data Provider must ensure to remove the data from TRUSTEE / stop sharing the data of data subjects when requested by a data subject, required by the data protection laws and when there is no legal basis for further sharing and processing of the data.	Yes	OneSS will not store any personal data.
LEG-USR-018	Data Provider and Consumer must answer the data subject's rights: • Information about the data processing (Article 12-14) • Access (Article 15) • Rectification (Article 16) • Erasure (Article 17) • Restriction on processing (Article 18) • Portability (Article 20) • Object (Article 21) • Not to be subject to a decision based solely on automated processing, including profiling (Article 22) • Lodge a complaint with the supervisory authority (Article 13).	Yes	OneSS will not store or process any personal data.
LEG-USR-024	Data Provider and Data Consumer must use the technical and organisational measures to protect personal data. Data provider and data consumer should use measures offered by TRUSTEE (Homomorphic encryption, SSI, authentication, federated learning, etc.).	Yes	OneSS will comply and use such measures, in accordance with the project-wide procedures and development processes.
LEG-USR-054	Data Provider / Consumer / Model Provider should be able to easily port the data when switching to other data service providers, i.e. to other platforms similar to TRUSTEE.	Yes	OneSS will not store any data that needs to be ported in stated situation.
LEG-USR-068	If the Data Provider / Model Provider is going to be a "data holder" in the meaning of the Data Act (see legal framework), they should fulfil the obligations listed in Article 4 Data Act concerning, inter alia, way of making the data available to the user, the form in which the data should be made	Yes	OneSS will not store or process any personal data.

	accessible, contractual agreement which may be concluded between user of a connected device/related service and Data provider, liability clauses, enabling identification through eID wallets, collecting minimum data about the user, protection of trade secrets (including construction of model contractual terms, confidential agreements, strict access protocols, technical standards), sharing of personal data and non-personal data does not apply to information derived or inferred from the data by complex algorithms combining output of multiple sensors.		
LEG-USR-079	"Data Provider / Model Provider / Developer being a ""participant of data space offering data or data services to other participants"" shall comply with the essential requirements to facilitate interoperability of data, data sharing mechanisms and services, defined in Article 28 Data Act: (a) the dataset content, use restrictions, licences, data collection methodology, data quality and uncertainty shall be sufficiently described in a machine-readable format to allow the recipient (Consumer) to find, access and use the data; (b) the data structures, data formats, vocabularies, classification schemes, taxonomies and code lists shall be described in a publicly available and consistent manner; (c) the technical means to access the data, such as application programming interfaces, and their terms of use and quality of service shall be sufficiently described to enable automatic access and transmission of data between parties, including continuously or in real-time in a machine-readable format where that is technically	Partially	OneSS can ensure parts of this requirement - that the necessary information is documented through the OneSS Developer Handbook, and the means to access the data from the external sources via external connectors.

feasible and does not hamper the good functioning of the product;
(d) the means to enable the interoperability of contracts for data sharing within their services and activities shall be provided."

13. ANNEX 3: SOCIO-ETHICAL REQUIREMENTS

The following sections detail the preliminary analysis of socio-ethical requirements presented in the D2.1 [1] related to each component. The ID of the respective socio-ethical requirement is used as a reference to D2.1, while the requirement's description is also provided alongside whether the ABBs comply with the requirement and how. The approach presented below for addressing each socio-ethical requirement may produce additional technical, functional and non-functional requirements that will be further specified in upcoming versions of this deliverable, namely D4.2 and D4.3, as well as in D2.2.

13.1. DA

Table 79: DA - Socio-ethical requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
SOC-ETH-TSD-004	Provide safeguards against overconfidence and overreliance on any of the data manipulation AI provisions on TRUSTEE.	Yes	DA SHOULD inform the user not to over rely on the data manipulation results produced by the TRUSTEE platform.
SOC-ETH-TSD-007	Ensure universal design for accessibility, explicability and explainability specifically where non-technical end-users engage with open data platforms such as TRUSTEE. Their experience and understanding of AI and data manipulation, their age, abilities, and other characteristics reflecting how use and interpretation of the produced results from TRUSTEE are accepted should be considered by TRUSTEE's developers.	Yes	DA MUST follow the UI/UX Design guidelines and recommendations.
SOC-ETH-TSD-022	Provide opportunity for stakeholder participation and feedback collection to maintain TRUSTEE's trustworthiness. Such mechanisms should be easy to spot and access and ensure it elicits regular feedback from all stakeholders.	Yes	DA SHOULD have a section/page that will allow users to give feedback.
SOC-ETH-TSD-023	Provide and update codes of conduct that technical and non-technical stakeholders of TRUSTEE such as developers and data providers are clear about and adhere to.	Partially	DA will investigate the presentation of codes of conduct
SOC-ETH-TSD-037	Clearly communicate which audiences the platform and the data manipulation and AI functionalities are intended for.	Partially	DA COULD display a disclaimer on how the data and AI functionalities will be used

SOC-ETH-USR-001	Data consumer should be made aware and clearly informed that any data or data manipulation and AI use should respect human dignity (i.e., treat humans as 'moral subjects' not as 'objects' that can be scored, manipulated etc.	Yes	DA will display the appropriate messages.
SOC-ETH-USR-003	Data consumer should be made aware and clearly informed that the system (e.g., TRUSTEE) is designed to respect democracy, justice, and the rule of law, plurality of choice, values, and decision-making of individuals.	Yes	DA will display the appropriate disclaimers.
SOC-ETH-USR-004	Data consumer should be made aware that the platform respects equality, non-discrimination, and solidarity whereby datasets used to train AI models strive to consider diversity and be inclusive of diverse populations.	Yes	DA will display the appropriate disclaimers.
SOC-ETH-USR-006	Data consumer should be made aware that the platform is designed to ensure that any AI and data manipulation deployed is done for the purpose* of enhancing and benefiting human and societal wellbeing and human capabilities.	Yes	DA will display the appropriate disclaimers.
SOC-ETH-USR-007	Data consumer should be made aware that the platform aims to provide safeguards against overconfidence and overreliance on any of the data manipulation AI provisions on TRUSTEE.	Yes	DA will display the appropriate disclaimers.
SOC-ETH-USR-008	Data consumer should be made aware that the platform aims to prioritise sustainable development, personal data rights, and agency over digital identity; and ensure legal frameworks for accountability are met.	Yes	DA will display the appropriate disclaimers.
SOC-ETH-USR-013	Provide transparency around the AI and data manipulation are functions with capabilities as well as limitations and they do not act as replacement of or on human individuals' behalf. That such functions are part of open data platforms such as TRUSTEE must be clearly communicated across to ensure that end users identify the AI functionalities as such.	Yes	DA will display the appropriate disclaimers.

13.2. DPIA

Table 80: DPIA – Socio-ethical Requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
SOC-ETH-TSD-002	Respect citizens' rights (regardless of circumstances of individuals, e.g., illegal immigrants) in deploying AI onto open data markets such as TRUSTEE	Yes	The DPIA tool respects citizens' rights. DPIA ABB implements comprehensive assessment protocols and

			ethical guidelines. These protocols ensure that AI technologies are developed and deployed with a strict adherence to fundamental human rights, including privacy and non-discrimination.
SOC-ETH-TSD-007	Ensure universal design for accessibility, explicability and explainability specifically where non-technical end-users engage with open data platforms such as TRUSTEE. Their experience and understanding of AI and data manipulation, their age, abilities, and other characteristics reflecting how use and interpretation of the produced results from TRUSTEE are accepted should be considered by TRUSTEE's developers.	Yes	The DPIA tool MUST follow the UI/UX design guidelines and recommendations.
SOC-ETH-TSD-011	Ensure that data privacy and protection are guaranteed. Any AI and data manipulation should prior to their work have guaranteed privacy, security and protection throughout the functionalities' and the platform's entire lifecycle.	Yes	The DPIA tool follows the CNIL methodology to ensure privacy, security and protection
SOC-ETH-TSD-014	Prevent harm through technological security and safety of the systems that handle data and data processing; provide sufficient information around data manipulation and AI (AI that is transparent and understandable to end users on how manipulation and AI arrive at specific results). Particular attention must be paid in relation to vulnerable groups such as children, persons with disabilities, historical disadvantages, and any other individuals or collectives who are at risk of exclusion.	Yes	The DPIA tool will not be accessible from unauthorized users.
SOC-ETH-TSD-024	Ensure TRUSTEE system developers are clear about the regulatory obligations with regards to data manipulation and the deployment of AI functionalities	Yes	The legal partners provide information and guidance for the DPIA tool developers to align with regulatory obligations and DPIA tool developers consider and adhere to them.
SOC-ETH-TSD-025	Ensure that TRUSTEE system developers adhere to clear design and data standards	Yes	The DPIA tool follows the CNIL methodology.

SOC-ETH-TSD-034	Ensure societal and environmental wellbeing through explicability, and transparency around choice and alternative ways of achieving the same or similar results as the ones promised through data manipulation and AI use in open data platforms such as TRUSTEE	Yes	The purpose of data processing as well as legitimate interest will be transparent.
SOC-ETH-TSD-035	Ensure protocols around access to data that is made available on open data platforms such as TRUSTEE are clear and put in place.	Yes	The DPIA tool follows the CNIL methodology with regards to the access solutions. Unauthorised and not authenticated parties will not have access to DPIA.
SOC-ETH-TSD-041	Ensure that prior to ethical and trustworthy assessment (i.e., the implementation of all of the above requirements), the TRUSTEE System Developers also meet the legally binding requirements (where relevant)	Yes	The DPIA tool developers have been informed about legal and ethical requirements in the context of WP2 and take them into account in the development and implementation process.

13.3. SSI-HE

Table 81: SSI-HE - Socio-ethical requirements

Req id (WP2)	Req. Description	Compliance	How is addressed
SOC-ETH-TSD-003	Ensure that any AI and data manipulation deployed is done for the purpose* of enhancing and benefiting human and societal wellbeing and human capabilities.	Yes	SSI-HE is based on the state of the art Open FHE and Hyperledger, which are well established information security solutions.
SOC-ETH-TSD-004	Provide safeguards against overconfidence and overreliance on any of the data manipulation AI provisions on TRUSTEE.	Yes	SSI-HE is based on the state of the art Open FHE and Hyperledger, which are well established information security solutions.
SOC-ETH-TSD-005	Prioritise sustainable development, personal data rights, and agency over digital identity; and ensure legal frameworks for accountability are met.	Yes	SSI-HE addresses it by the fact that it only deals with meta data and does not allow for access to personal data / the specific information of the data provider
SOC-ETH-TSD-011	Ensure that data privacy and protection are guaranteed. Any AI and data manipulation should prior to their work have guaranteed privacy, security and	Yes	SSI-HE is based on the state of the art Open FHE and Hyperledger, which are well

protection throughout the functionalities' and the platform's entire lifecycle.	established information security solutions.
---	---

13.4. DL

Table 82: DL - Socio-ethical requirements

Req id	Req. Description	Compliance	How is addressed
SOC-ETH-TSD-025	Ensure that TRUSTEE system developers and other stakeholders (e.g., data providers) adhere to clear design and data standards.	Yes	Establish agile feedback channels with different implementation phases. Ensure data standards used (e.g. semantic ontologies) comply with legal and regulatory requirements, such as data protection laws (e.g., GDPR)

13.5. STM

Table 83: STM - Social-ethical requirements.

Req id (WP2)	Req. Description	Compliance	How it is addressed
SOC-ETH-TSD-025	Ensure that TRUSTEE system developers adhere to clear design and data standards.	Yes	The development of the STM includes well-known design patterns.
SOC-ETH-TSD-037	Clearly communicate which audiences the platform and the data manipulation and AI functionalities are intended for.	Partially	The STM enables the Data Provider to indicate from which domain the dataset originates from, what actions should be applied and for what purposes their dataset may be processed for.
SOC-ETH-USR-006	Data consumer should be made aware that the platform is designed to ensure that any AI and data manipulation deployed is done for the purpose* of enhancing and benefiting human and societal wellbeing and human capabilities.	Partially	Consumers can see the data processing purposes for each dataset within the corresponding agreements.

13.6. ONESS

Table 84: OneSS – Socio-ethical requirements

Req id	Req. Description	Compliance	How is addressed
SOC-ETH-TSD-022	Provide opportunity for stakeholder participation and feedback collection to maintain TRUSTEE's trustworthiness. Such mechanisms should be easy to spot and access and ensure it elicits regular feedback from all stakeholders.	Yes	The users will be able to provide feedback via TRUSTEE GitHub, but also user feedback will be gathered throughout the project and TRUSTEE pilot campaign
SOC-ETH-TSD-024	Ensure TRUSTEE system developers are clear about the regulatory obligations with regards to data manipulation and the deployment of AI functionalities	Yes	Information about regulatory obligations is provided by TRUSTEE legal partners and has been made available to TRUSTEE system developers will be available to users.
SOC-ETH-TSD-031	Provide a clear 'whitelist' of rules, policies and procedures that TRUSTEE developers are guided by (and ensure transparency for end user's consumption) and provide a clear 'black list' restrictions and behaviours (AI and data manipulation that is prohibited or used in certain circumstances) which can be used for audit and compliance by outside third parties and for the maintenance of the platform's trustworthiness.	Yes	Information about the development process and onboarding information are available to users via Developer Handbook, and as such they will contain a list of procedures and rules for TRUSTEE developers. As such, TRUSTEE Handbook enables compliance with this requirement, although the content is yet to be populated and will be expanded and improved during the project.
SOC-ETH-TSD-042	Provide regular assessment of socio-ethical requirements throughout the lifecycle of the project to ensure that this list is updated, that education and awareness among key stakeholders (and especially amount TRUSTEE developers) grows and improves, and that the mandatory and subsequently desirable requirements are met.	Yes	In addition to planned regular assessment of socio-ethical requirements related to the OneSS throughout the project lifecycle, OneSS will also cover the education of the key stakeholders by providing the relevant information in the Handbook.

14. ANNEX 4: DPIA ENDPOINTS

Table 85 details the available endpoints of the DPIA ABB API

Table 85: Endpoints of the RESTful API of the DPIA ABB

ID #01	
Description	Returns all DPIAs.
Endpoint	/pias
Authentication	
Method	GET
Input parameters	
Request Body	
Response Body	<pre>[{ "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "", "structure_data": null, "category": "string", "is_archive": 0 }]</pre>
ID #02	
Description	Creates a new DPIA.

Endpoint	/pias
Authentication	
Method	POST
Input parameters	body:object*
Request Body	<pre>{ "name": "string", "status": 0, "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_option": "string", "dpos_name": "string", "people_name": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string" }</pre>
Response Body	<pre>{ "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "" }</pre>

	<pre> "structure_data": null, "category": "string", "is_archive": 0 } </pre>
ID #03	
Description	Returns a DPIA.
Endpoint	/pias/{id}
Authentication	
Method	GET
Input parameters	id:integer*
Request Body	
Response Body	<pre> { "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "", "structure_data": null, "category": "string", "is_archive": 0 } </pre>
ID #04	
Description	Updates a DPIA.
Endpoint	/pias/{id}
Authentication	
Method	PATCH

Input parameters	id:integer*, body:object*
Request Body	<pre>{ "name": "string", "status": 0, "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_option": "string", "dpos_name": "string", "people_name": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string" }</pre>
Response Body	<pre>{ "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "", "structure_data": null, "category": "string", "is_archive": 0 }</pre>

ID #05	
Description	Deletes a single DPIA.
Endpoint	/pias/{id}
Authentication	
Method	DELETE
Input parameters	id:integer*
Request Body	
Response Body	
ID #06	
Description	Duplicates a DPIA.
Endpoint	/pias/{id}/duplicate
Authentication	
Method	POST
Input parameters	id:integer*
Request Body	
Response Body	<pre>{ "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "", "structure_data": null, "category": "string", "is_archive": 0 }</pre>

ID #07	
Description	Imports a DPIA from a json export.
Endpoint	/pias/import
Authentication	
Method	POST
Input parameters	data:file
Request Body	
Response Body	<pre>{ "id": 1, "status": 0, "name": "string", "author_name": "string", "evaluator_name": "string", "validator_name": "string", "dpo_status": 0, "dpo_opinion": "string", "dpos_names": "string", "people_names": "string", "concerned_people_opinion": "string", "concerned_people_status": 0, "rejection_reason": "string", "applied_adjustments": "string", "is_example": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z", "concerned_people_searched_opinion": true, "concerned_people_searched_content": "string", "structure_id": null, "structure_name": "", "structure_sector_name": "", "structure_data": null, "category": "string", "is_archive": 0 }</pre>
ID #08	
Description	Returns all answers.
Endpoint	/pias/{pia_id}/answers
Authentication	
Method	GET
Input parameters	pia_id:integer*
Request Body	
Response Body	[

	<pre>{ "id": 1, "pia_id": 1, "reference_to": "string", "data": { "text": "string", "gauge": "" }, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #09	
Description	Creates a new answer.
Endpoint	/pias/{pia_id}/answers
Authentication	
Method	POST
Input parameters	pia_id:integer*, body:object*
Request Body	<pre>{ "reference_to": "string", "data": { "text": "string", "gauge": "" } }</pre>
Response Body	<pre>{ "id": 1, "pia_id": 1, "reference_to": "string", "data": { "text": "string", "gauge": "" }, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #10	
Description	Returns an answer.
Endpoint	/pias/{pia_id}/answers/{id}
Authentication	
Method	GET
Input parameters	pia_id:integer*, id:integer*

Request Body	
Response Body	<pre>{ "id": 1, "pia_id": 1, "reference_to": "string", "data": { "text": "string", "gauge": "" }, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #11	
Description	Updates an answer.
Endpoint	/pias/{pia_id}/answers/{id}
Authentication	
Method	PATCH
Input parameters	pia_id:integer*, id:integer*, body:object*
Request Body	<pre>{ "reference_to": "string", "data": { "text": "string", "gauge": "" } }</pre>
Response Body	<pre>{ "id": 1, "pia_id": 1, "reference_to": "string", "data": { "text": "string", "gauge": "" }, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #12	
Description	Deletes a single answer.
Endpoint	/pias/{pia_id}/answers/{id}
Authentication	
Method	DELETE
Input parameters	pia_id:integer*, id:integer*

Request Body	
Response Body	
ID #13	
Description	Returns all comments.
Endpoint	/pias/{pia_id}/comments
Authentication	
Method	GET
Input parameters	pia_id:integer*
Request Body	
Response Body	<pre>[{ "id": 1, "pia_id": 1, "description": "string", "reference_to": "string", "for_measure": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }]</pre>
ID #14	
Description	Creates a new comment.
Endpoint	/pias/{pia_id}/comments
Authentication	
Method	POST
Input parameters	pia_id:integer*, body:object*
Request Body	<pre>{ "description": "string", "reference_to": "string", "for_measure": false }</pre>
Response Body	<pre>{ "id": 1, "pia_id": 1, "description": "string", "reference_to": "string", "for_measure": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #15	

Description	Returns a comment.
Endpoint	/pias/{pia_id}/comments/{id}
Authentication	
Method	GET
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	<pre>{ "id": 1, "pia_id": 1, "description": "string", "reference_to": "string", "for_measure": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #16	
Description	Updates a comment.
Endpoint	/pias/{pia_id}/comments/{id}
Authentication	
Method	PATCH
Input parameters	pia_id:integer*, id:integer*, body:object*
Request Body	<pre>{ "description": "string", "reference_to": "string", "for_measure": false }</pre>
Response Body	<pre>{ "id": 1, "pia_id": 1, "description": "string", "reference_to": "string", "for_measure": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #17	
Description	Deletes a single comment.
Endpoint	/pias/{pia_id}/comments/{id}
Authentication	
Method	DELETE
Input parameters	pia_id:integer*, id:integer*

Request Body	
Response Body	
ID #18	
Description	Returns all evaluations.
Endpoint	/pias/{pia_id}/evaluations
Authentication	
Method	GET
Input parameters	pia_id:integer*
Request Body	
Response Body	<pre>[{ "id": 1, "pia_id": 1, "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": null, "gauges": {}, "estimated_implementation_date": null, "person_in_charge": "", "global_status": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }]</pre>
ID #19	
Description	Creates a new evaluation.
Endpoint	/pias/{pia_id}/evaluations
Authentication	
Method	POST
Input parameters	pia_id:integer*, body:object*
Request Body	<pre>{ "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": "string", "gauges": {}, "estimated_implementation_date": "string", "person_in_charge": "string", }</pre>

	"global_status": 0 }
Response Body	{ "id": 1, "pia_id": 1, "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": null, "gauges": {}, "estimated_implementation_date": null, "person_in_charge": "", "global_status": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }
ID #20	
Description	Returns an evaluation.
Endpoint	/pias/{pia_id}/evaluations/{id}
Authentication	
Method	GET
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	{ "id": 1, "pia_id": 1, "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": null, "gauges": {}, "estimated_implementation_date": null, "person_in_charge": "", "global_status": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }
ID #21	
Description	Updates an evaluation.
Endpoint	/pias/{pia_id}/evaluations/{id}
Authentication	

Method	PATCH
Input parameters	pia_id:integer*, id:integer*, body:object*
Request Body	<pre>{ "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": "string", "gauges": {}, "estimated_implementation_date": "string", "person_in_charge": "string", "global_status": 0 }</pre>
Response Body	<pre>{ "id": 1, "pia_id": 1, "status": 0, "reference_to": "string", "action_plan_comment": "string", "evaluation_comment": "string", "evaluation_date": null, "gauges": {}, "estimated_implementation_date": null, "person_in_charge": "", "global_status": 0, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #22	
Description	Deletes a single evaluation.
Endpoint	/pias/{pia_id}/evaluations/{id}
Authentication	
Method	DELETE
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	
ID #23	
Description	Returns all measures.
Endpoint	/pias/{pia_id}/measures
Authentication	
Method	GET

Input parameters	pia_id:integer*
Request Body	
Response Body	[<pre> { "id": 1, "pia_id": 1, "title": "string", "content": "string", "placeholder": "string" }] </pre>
ID #24	
Description	Creates a new measure.
Endpoint	/pias/{pia_id}/measures
Authentication	
Method	POST
Input parameters	pia_id:integer*, body:object*
Request Body	<pre> { "title": "string", "content": "string", "placeholder": "string" } </pre>
Response Body	<pre> { "id": 1, "pia_id": 1, "title": "string", "content": "string", "placeholder": "string" } </pre>
ID #25	
Description	Returns a measure.
Endpoint	/pias/{pia_id}/measures/{id}
Authentication	
Method	GET
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	<pre> { "id": 1, "pia_id": 1, "title": "string", "content": "string", "placeholder": "string" } </pre>

	}
ID #26	
Description	Updates a measure.
Endpoint	/pias/{pia_id}/measures/{id}
Authentication	
Method	PATCH
Input parameters	pia_id:integer*, id:integer*, body:object*
Request Body	{ "title": "string", "content": "string", "placeholder": "string" }
Response Body	{ "id": 1, "pia_id": 1, "title": "string", "content": "string", "placeholder": "string" }
ID #27	
Description	Deletes a single measure.
Endpoint	/pias/{pia_id}/measures/{id}
Authentication	
Method	DELETE
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	
ID #28	
Description	Returns all attachments.
Endpoint	/pias/{pia_id}/attachments
Authentication	
Method	GET
Input parameters	pia_id:integer*
Request Body	
Response Body	[{ "id": 1, "pia_id": 1, "comment": "" }

	<pre> "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z ", "file": "", "name": "string", "mime_type": "string" }]</pre>
ID #29	
Description	Creates a new attachment.
Endpoint	/pias/{pia_id}/attachments
Authentication	
Method	POST
Input parameters	pia_id:integer*, body:object*
Request Body	<pre> { "comment": "", "file": "", "name": "string", "mime_type": "string" }</pre>
Response Body	<pre> { "id": 1, "pia_id": 1, "comment": "", "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z ", "file": "", "name": "string", "mime_type": "string" }</pre>
ID #30	
Description	Returns an attachment.
Endpoint	/pias/{pia_id}/attachments/{id}
Authentication	
Method	GET
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	<pre> { "id": 1, "pia_id": 1, "comment": "", "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z ", }</pre>

	<pre>"file": "", "name": "string", "mime_type": "string" }</pre>
ID #31	
Description	Deletes a single attachment.
Endpoint	/pias/{pia_id}/attachments/{id}
Authentication	
Method	DELETE
Input parameters	pia_id:integer*, id:integer*
Request Body	
Response Body	
ID #32	
Description	Returns all STRUCTURES.
Endpoint	/structures
Authentication	
Method	GET
Input parameters	sorting[column]:string, sorting[direction]:string
Request Body	
Response Body	<pre>[{ "id": 1, "name": "string", "sector_name": "string", "data": {}, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }]</pre>
ID #33	
Description	Creates a new STRUCTURE.
Endpoint	/structures
Authentication	
Method	POST
Input parameters	body:object*
Request Body	<pre>{ "name": "string", "sector_name": "string", "data": {}, }</pre>

Response Body	{ "id": 1, "name": "string", "sector_name": "string", "data": {}, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }
ID #34	
Description	Returns a STRUCTURE.
Endpoint	/structures/{id}
Authentication	
Method	GET
Input parameters	id:integer*
Request Body	
Response Body	{ "id": 1, "name": "string", "sector_name": "string", "data": {}, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }
ID #35	
Description	Updates a STRUCTURE.
Endpoint	/structures/{id}
Authentication	
Method	PATCH
Input parameters	id:integer*, body:object*
Request Body	{ "structure": "name": "string", "sector_name": "string", "data": "string" }
Response Body	{ "name": "string", "sector_name": "string", "data": "string" }
ID #36	

Description	Deletes a single STRUCTURE.
Endpoint	/structures/{id}
Authentication	
Method	DELETE
Input parameters	id:integer*
Request Body	
Response Body	
ID #37	
Description	Returns all knowledge bases
Endpoint	/knowledge_bases
Authentication	
Method	GET
Input parameters	
Request Body	
Response Body	[{ "id": 1, "name": "string", "author": "string", "contributors": "string", "is_example": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }]
ID #38	
Description	Create a new knowledge base
Endpoint	/knowledge_bases
Authentication	
Method	POST
Input parameters	body:object*
Request Body	{ "name": "string", "author": "string", "contributors": "string" }
Response Body	{ "id": 1, "name": "string", "author": "string", }

	<pre>"contributors": "string", "is_example": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #39	
Description	Returns a knowledge base
Endpoint	/knowledge_bases/{id}
Authentication	
Method	GET
Input parameters	id:integer*
Request Body	
Response Body	<pre>{ "id": 1, "name": "string", "author": "string", "contributors": "string", "is_example": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #38	
Description	Updates a knowledge base
Endpoint	/knowledge_bases/{id}
Authentication	
Method	PATCH
Input parameters	id:integer*, body:object*
Request Body	<pre>{ "name": "string", "author": "string", "contributors": "string" }</pre>
Response Body	<pre>{ "id": 1, "name": "string", "author": "string", "contributors": "string", "is_example": false, "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #39	

Description	Deletes a knowledge base
Endpoint	/knowledge_bases/{id}
Authentication	
Method	DELETE
Input parameters	id:integer*
Request Body	
Response Body	
ID #40	
Description	Returns all knowledges from a knowledge base
Endpoint	/knowledge_bases/{base_id}/knowledges
Authentication	
Method	GET
Input parameters	base_id:integer*
Request Body	
Response Body	<pre>[{ "id": 1, "name": "string", "slug": "string", "filters": "string", "category": "string", "placeholder": null, "description": "string", "items": [], "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }]</pre>
ID #41	
Description	Creates a new knowledge for a knowledge base
Endpoint	/knowledge_bases/{base_id}/knowledges
Authentication	
Method	POST
Input parameters	base_id:integer*, body:object*
Request Body	<pre>{ "name": "string", "slug": "string", "filters": "string", "category": "string", "placeholder": null,</pre>

	<pre>"description": "string", "items": [] }</pre>
Response Body	<pre>{ "id": 1, "name": "string", "slug": "string", "filters": "string", "category": "string", "placeholder": null, "description": "string", "items": [], "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #42	
Description	Returns a knowledge from a knowledge base
Endpoint	/knowledge_bases/{base_id}/knowledges{id}
Authentication	
Method	GET
Input parameters	base_id:integer*, id:integer*
Request Body	
Response Body	<pre>{ "id": 1, "name": "string", "slug": "string", "filters": "string", "category": "string", "placeholder": null, "description": "string", "items": [], "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" }</pre>
ID #43	
Description	Updates a knowledge of a knowledge base
Endpoint	/knowledge_bases/{base_id}/knowledges{id}
Authentication	
Method	PATCH
Input parameters	base_id:integer*, id:integer*, body:object*
Request Body	<pre>{ "name": "string",</pre>

	<pre> "slug": "string", "filters": "string", "category": "string", "description": "string", "items": [] } </pre>
Response Body	<pre> { "id": 1, "name": "string", "slug": "string", "filters": "string", "category": "string", "placeholder": null, "description": "string", "items": [], "created_at": "0000-00-00T00:00:00.000Z", "updated_at": "0000-00-00T00:00:00.000Z" } </pre>
ID #44	
Description	Deletes a knowledge from a knowledge base
Endpoint	/knowledge_bases/{base_id}/knowledges{id}
Authentication	
Method	DELETE
Input parameters	base_id:integer*, id:integer*
Request Body	
Response Body	

15. ANNEX 6: STM ENDPOINTS

The endpoint design is used for representing the different operations and are represented by the different HTTP methods as explained in the previous subsection (Operations). Table 86 and Table 87 represent the current state of the STM's internal REST API, providing endpoints required to be able to register, log in, view datasets, view agreements, sign agreements, and view users.

Table 86: The first table of the STM's REST API endpoints.

Endpoint	Operation	Input	Output	Description
/api/v1/users	GET		JSON LIST of User Objects	Get all users
/api/v1/users/{id}	GET	STRING ID	JSON User Object	Get a single user by their ID
/api/v1/auth/register	POST	JSON User Object	JSON User Object + JWT	Create a new user and return a Java Web Token
/api/v1/auth/login	POST	STRING email, STRING password	JSON User Object + JWT	Authenticate a user and return a Java Web Token
/api/v1/metadatas	GET		JSON LIST of Metadata Objects	Get all metadatas
/api/v1/metadatas/{id}	GET	STRING ID	JSON Metadata Object	Get a single metadata by its ID
/api/v1/metadatas	POST	JSON Metadata Object	JSON Metadata Object with ID	Creates a new metadata object
/api/v1/metadatas/{id}	PUT	JSON Metadata Object with ID	JSON Metadata Object with ID	Updates a metadata to the given input metadata
/api/v1/metadatas/{id}	DELETE	STRING ID	HTTP Status Code 202	Deletes the specified metadata object
/api/v1/datasets	GET		JSON LIST of Dataset Objects	Get all datasets
/api/v1/datasets/{id}	GET	STRING ID	JSON Dataset Object	Get a single dataset by its ID
/api/v1/datasets/ownerID/{id}	GET	STRING ID	JSON LIST of Dataset Objects	Get all datasets created by the given user's ID
/api/v1/datasets	POST	JSON Dataset Object	JSON Dataset Object with ID	Creates a new dataset object
/api/v1/datasets/{id}	DELETE	STRING ID	HTTP Status Code 202	Deletes the specified dataset object
/api/v1/organizations	GET		JSON LIST of Organization Objects	Get all organizations
/api/v1/organizations/{id}	GET	STRING ID	JSON Organization Object	Get a single organization by its ID

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

/api/v1/organizations	POST	JSON Organization Object	JSON Organization Object with ID	Creates a new organization object
/api/v1/organizations/{id}	PUT	JSON Organization Object	HTTP Status Code 202	Updates an organization to the given input organization
/api/v1/organizations/{id}	DELETE	STRING ID	HTTP Status Code 202	Deletes the specified organization object

Table 87: The second table of the STM's REST API endpoints.

Endpoint	Operation	Input	Output	Description
/api/v1/actions	GET		JSON LIST of Action Objects	Get all actions
/api/v1/actions/{id}	GET	STRING ID	JSON Action Object	Get a single action by its ID
/api/v1/actions	POST	JSON Action Object	JSON Action Object with ID	Creates a new action object
/api/v1/actions/{id}	PUT	JSON Action Object	JSON Action Object with ID	Updates an action to the given input action
/api/v1/actions/{id}	DELETE	STRING ID	HTTP Status Code 202	Deletes the specified action object
/api/v1/purposes	GET		JSON LIST of Purpose Objects	Get all purposes
/api/v1/purposes/{id}	GET	STRING ID	JSON Purpose Object	Get a single purpose by its ID
/api/v1/purposes	POST	JSON Purpose Object	JSON Purpose Object with ID	Creates a new purpose object
/api/v1/purposes/{id}	PUT	JSON Purpose Object	JSON Purpose Object with ID	Updates a purpose to the given input purpose

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

/api/v1/purposes/{id}	DELETE	STRING ID	HTTP Code 202	Status	Deletes the specified purpose object
/api/v1/agreements	GET				Get all agreements
/api/v1/agreements/{id}	GET	STRING ID			Get a single agreement by its ID
/api/v1/agreements/signerID/{id}	GET	STRING ID	JSON LIST of Agreement Objects		Get all agreements that are associated to the given user ID
/api/v1/agreements	POST	JSON Agreement Object	JSON Agreement Object with ID		Creates a new agreement object
/api/v1/agreements/{id}/sign	POST	STRING ID, STRING status	HTTP Code 202	Status	A user signs an agreement
/api/v1/agreements/{id}	DELETE	STRING ID	HTTP Code 202	Status	Deletes the specified agreement object

Copyright © 2022. All rights reserved.

16. ANNEX 5: DEVOPS AND CI/CD METHODOLOGY

DevOps is a general collection of flexible software creation and delivery practices that looks to close the gap between software development and IT operations, which are two critical but often misaligned efforts. It's not a specific technology, but a tactical approach. By working together, development and operations teams can eliminate roadblocks and focus on improving the creation, deployment, and continuous monitoring of software.

This comprehensive approach enables businesses to achieve faster delivery of high quality software products, adapt to evolving market demands, and ultimately enhance customer satisfaction.

Before proceeding, it is necessary to introduce some general concepts, including:

- **Agile:** Managing changing requirements and minimizing the costs of misunderstandings. Short iterations allow for quick feedback and progressive improvements. Faster time-to-market by starting with a Minimum Viable Product.
- **DevOps:** Supporting rapid iterations through the automation of the CI/CD pipeline. Reducing change risks and increasing security with a reproducible process. Eliminating bottlenecks by fostering collaboration between developers and operations.
- **Containers:** Simplifying deployment with self-contained images that include dependencies. Lighter images and optimized resource usage by eliminating the hypervisor. Immutable infrastructure, autoscaling, self-healing.
- **Cloud Native:** Breaking down monoliths into smaller, more manageable units with microservices. Team and technology independence for each microservice (right tool for the job). Elasticity for deployment and automatic scaling with containers and the 12 factors.

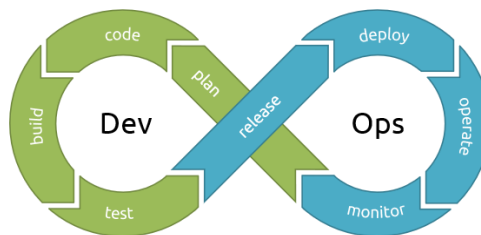


Figure 221: DevOps - Stages

16.1. SCOPE

The following guidelines outline how to adopt the DevOps model in the development process by automating build/test/release/deploy operations through Continuous Integration/Continuous Delivery pipelines.

Figure 222 presents a possible implementation of a platform for DevOps development:

The platform consists of a set of tools that support the execution of various stages of the DevOps process, from activity planning to production deployment, enabling key features such as team collaboration and Continuous Integration and Continuous Delivery processes. In particular, one of the primary objectives of the tools is the automation of build/test/deploy cycles.

Below are some of the possible tools that can be used and the programming languages currently supported.

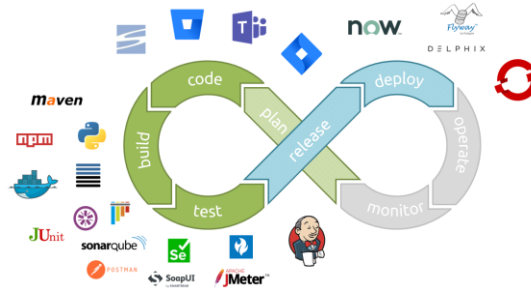


Figure 222: DevOps - Tools

Table 88: DevOps – Stages and tools

Stage	Scope	Tool
Plan	Issue tracking, project mgmt	Jira Software Server
Plan	Collaboration	Teams
Code	Source code management	Bitbucket Server, Subversion
Build	Build automation	Maven, Npm, Pip
Build	Container engine	Docker
Build, Test, Release, Deploy	Automation server	Jenkins
Build	Artifact repository	Nexus Repository OSS
Build	Container registry	OpenShift Container Registry
Test	Unit testing	JUnit, Jasmine, Pytest (altri possibili; vedi descrizione)
Test	API testing	PostMan, SoapUI
Test	End-to-end testing	Selenium
Test	Code Static Analysis	SonarQube Community Edition
Test	Security Dinamic Analysis	AppScan Enterprise
Test	Performance testing	JMeter
Release	IT Service Management	ServiceNow
Deploy	Container platform	OpenShift Container Platform
Deploy	Database migration	Flyway
Deploy	Data refresh	Delphix

16.2. DEVOPS PROCESS

The CI/CD pipeline is a fundamental feature of the DevOps methodology. It is a series of steps that start from code build and take a new version of the application through various stages of testing and verification in the intended environments, culminating in the release to production.

The stages of the pipeline are automated, with the exception of certain steps that require manual intervention, such as user testing or approval for release. Jenkins orchestrates the pipeline, ensuring the smooth flow of tasks and activities.

Implementing the CI/CD pipeline ensures that every new version of the application goes through a consistent and reproducible process of build, test, and deployment. This increases the quality of the software, speeds up release cycles, and supports the application of company policies, such as adherence to security criteria.

By automating and standardizing the release process, the CI/CD pipeline promotes efficiency, reliability, and adherence to best practices. It enables faster feedback loops, early bug detection, and easier rollbacks, ultimately leading to improved software delivery and customer satisfaction.

16.2.1. PIPELINE CI/CD – DIAGRAMS

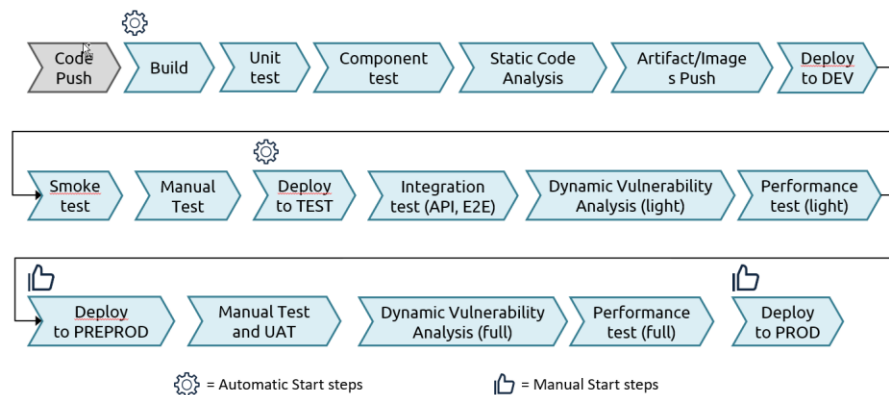


Figure 223: Pipeline CI/CD – Diagrams

16.2.2. PIPELINE CI/CD – TOOLS

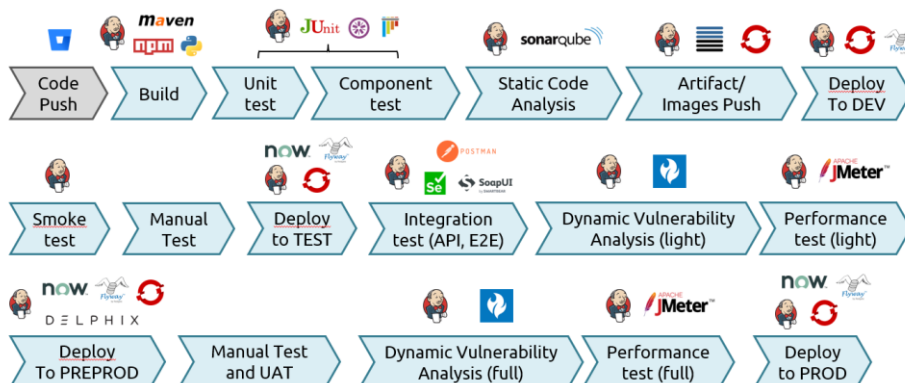


Figure 224: Pipeline CI/CD – Tools

16.2.3. PIPELINE CI/CD – STAGES

Code Push It is the trigger that initiates a new execution of the pipeline whenever a developer pushes new code to the repository.

Build Compilation (for compiled languages) or possible validation of sources (for interpreted languages).

Unit tests and Component tests Automated tests that does not require deployment in an external environment. The execution can be included in the Build step, for example, using tools like Maven. Unit tests verify individual classes in isolation, using mocks for all external dependencies.

Static code analysis Static analysis of the code to identify poor programming practices, potential bugs, and security vulnerabilities. Downstream from static analysis, the pipeline may include a quality gate that fails the build or marks it as unstable if the code does not meet the criteria defined by the quality gate, such as the absence of critical vulnerabilities or an adequate level of code coverage in automated tests.

Artifact/Image Push Packaging artifacts (e.g., JAR files), building container images, and publishing them to artifact repositories and container registries. The packaging of artifacts can be included in the Build step, for example, using tools like Maven. In a container-based application, the publication of artifacts such as JAR and WAR files is redundant and unnecessary as only the container images will be deployed. The publication of artifacts is still necessary for shared libraries that will be reused in different pipelines and for legacy applications not based on containers that will be deployed in a traditional manner (e.g., WAR/EAR files).

Deploy to DEV Automated deployment of the application to the DEV environment.

Smoke test Quick verification of the correct deployment on the DEV environment. For example, running a script that simulates HTTP calls to the main page and user login, and verifies the response status codes.

Manual Testing on DEV The DEV environment hosts a complete deployment of the application and can be utilized by developers to perform an initial set of manual integration tests on implemented functionalities.

Deploy to TEST Automated deployment of the application to the TEST environment. Before performing the deployment, the automated process must open a Change Request in ServiceNow, solely for tracking purposes and without specific approval requirements.

Integration test (API, E2E) Execution of a suite of automated functional tests on the application deployed in the TEST environment. These tests may include testing the APIs exposed by the backend and end-to-end tests that simulate the interaction of a real user with the UI and services. Regression testing is also included.

Light Dynamic Vulnerability Analysis Execution of automated penetration testing (DAST) on the application deployed in the TEST environment. The scanning should be performed with a light profile to verify the main vulnerability scenarios within a reasonably short execution time. This helps to contain the total execution time of the entire test suite (functional, security, and performance) and allows for its execution once or multiple times per day.

Light Performance Test Execution of automated performance tests on the application deployed in the TEST environment. Similarly, these tests should have a reasonably short duration with the purpose of monitoring performance trends following recent code changes and identifying any major issues. This phase is not intended to measure the absolute performance of the application or perform stress testing (both of these activities occur in PREPROD).

Deploy to PREPROD Deployment of the application to the PREPROD environment. Unlike deployments to DEV and TEST, which are triggered automatically, the deployment to PREPROD is always triggered manually through an integrated process with Jira and ServiceNow, involving the opening of a typically pre-approved Change Request. It is important to note that while the trigger is manual, the deployment process itself is fully automated.

Manual Testing and UAT Manual testing of the application deployed in PREPROD by the Quality Assurance (QA) team. If applicable, user acceptance testing is performed by end users.

Full Dynamic Vulnerability Analysis Similar to the light penetration testing phase in the TEST environment, but in this case, in PREPROD, the scanning should be conducted with a full profile, which is more extensive, to perform a more thorough vulnerability assessment before release. The scanning is manually triggered to avoid interference with manual testing/UAT activities. Once triggered, the scanning is executed automatically.

Full Performance Test Similar to the light performance testing phase in the TEST environment, but in this case, the tests can be more demanding and may include stress testing. Being in the PREPROD environment allows for realistic measurement of application response times. The suite of tests is manually triggered to avoid interference with manual testing/UAT activities. Once initiated, the test suite is executed automatically.

Deploy to PROD Deployment of the application to the PROD environment. Similar to PREPROD, the deployment is always manually triggered through an integrated process with Jira and ServiceNow, involving the opening and active approval of a Change Request in ServiceNow. It is important to note that while the trigger is manual, the deployment process itself is fully automated.

16.2.4. PIPELINE CI/CD – ENVIRONMENTS

Below are some considerations regarding the use of different environments within the CI/CD pipeline.

16.2.4.1. ALL ENVIRONMENTS

In all cases, the application deployment to various environments is performed by Jenkins using an automated procedure. The procedure is triggered automatically for DEV and TEST environments and manually for PREPROD and PROD environments.

The automated procedure ensures that the deployment is reproducible from one environment to another based on the artifacts produced by the build and the versioned manifests/configurations in Git.

16.2.4.2. DEVELOPMENT (DEV)

DEV is the first environment with the integrated deployment of the entire application (e.g., frontend and all backend microservices). It is continuously updated by the Continuous Integration processes, which automatically deploy the updated components after each modification in the main development branch.

DEV is used by developers as:

- A support environment during development (e.g., a local service calling other services, frontend relying on the DEV backend).
- An environment for manual integration testing of the latest build (e.g., verifications that cannot be executed locally during development, bug report checks).

- Developers have full access to the environment, including console access to containers, log access, DB access, etc.

16.2.4.3. TEST (TEST)

Within the CI/CD pipeline, the TEST environment is intended as a disposable environment used for running automated integration tests.

To ensure that the automated tests start from a known initial state, the environment is recreated from scratch by Jenkins for each deployment, including any state reset operations (e.g., DB) and loading initial data in preparation for the tests.

16.2.4.4. PRE-PRODUCTION (PREPROD)

PREPROD is the environment for manual testing by the QA team and user acceptance testing (UAT) by end users.

The environment should closely resemble the production environment in terms of configuration and data, enabling realistic testing, including testing of the automated deployment and upgrade process.

Thanks to the data refresh process using Delphix, it is possible to update the data in the PREPROD DBs with an anonymized copy of the data from PROD upon request, facilitating more accurate testing.

All deployments in PREPROD are initiated with a manual request to ensure the stability of the environment during testing. Deployment requests are made through an integrated process with Jira and ServiceNow, involving the opening of a Change Request, typically pre-approved.

16.2.4.5. PRODUCTION (PROD)

It is the production environment of the application. Once again, deployment requests, managed through an integrated process with Jira and ServiceNow, require the opening of a Change Request that needs to be approved by project stakeholders.

16.2.5. ADOPTION IN PHASES

The CI/CD pipeline constitutes a comprehensive process that, through various automated checks, enables achieving a high level of software quality in terms of functionality, security, and performance.

However, the implementation of the different steps requires effort from the application team, as they need to automate deployment procedures and write automated test suites. Additionally, the reports produced by automated analysis tools (e.g., static analysis, dynamic analysis) enhance the quality only to the extent that they are regularly reviewed by the development team, which must commit to analyzing the identified issues and addressing them.

For this reason, it is common to adopt DevOps in phases.

In the initial phase, the steps of automatic build and deployment on various environments will be implemented, while end-to-end functional tests will remain manual. It is recommended, however, to introduce unit tests from the start, as they should always be part of the code development process.

In the second phase, static analysis, dynamic analysis, and an initial set of automated end-to-end functional tests will be introduced.

Subsequently, the suite of end-to-end tests will be gradually expanded. Applications with specific performance requirements will also implement a suite of automated performance tests.

The following bullet point list revisits the previously described DevOps process, going through it step by step, indicating the various actions and tools involved, using a Java-based microservices application that utilizes a relational database as an example.

16.2.5.1. DEVOPS PROCESS – WALKTHROUGH

- Planning and tracking development activities (features/bugs/other) on Jira.
- Developers write code using approved languages/frameworks (e.g., Java).
- All code is versioned on Bitbucket (Git) for existing codebases.
- Each developer writes unit/component automated tests to verify the code.
- The code includes descriptors for automated builds with execution of unit/component tests (e.g., Maven and JUnit).
- The developer tests locally (Docker).
- Push code changes to Bitbucket.
- Upon each code push, Bitbucket triggers a webhook on Jenkins, which automatically initiates the Continuous Integration build for the modified component.
- Jenkins performs the build, unit tests, and component tests (Maven, JUnit).
- Jenkins performs static code analysis (SonarQube).
- Jenkins produces artifacts and Docker images, publishing them to respective repositories (Nexus, OCR NoProd).
- Each build is a potential release if it passes all stages of the pipeline.
- Jenkins automatically deploys to the DEV environment (Flyway, OCP), followed by smoke testing.
- Developers conduct integration tests on the DEV environment.
- At scheduled times established by the team, Jenkins automatically deploys to the TEST environment, which is recreated from scratch each time (Flyway, OCP). The deployment job tracks the operation by opening a pre-approved Change Request on ServiceNow.
- Jenkins initiates automated tests (functional with Selenium, Postman, and/or SoapUI, security with AppScan, performance with JMeter) on the TEST environment.
- To release to PREPROD, the System Integrator opens a Deploy Request on Jira, triggering the opening of a Change Request on ServiceNow.
- The Change Request is typically pre-approved.
- At the deployment time specified in the request, ServiceNow invokes Jenkins, which performs the automated deployment job on PREPROD (Flyway, OCP). If requested, the process can perform data refresh on PREPROD (Delphix).
- The QA team and/or users conduct testing on the PREPROD environment.
- The System Integrator initiates a full AppScan scan on PREPROD through a Jenkins job.
- If applicable, the System Integrator launches JMeter performance tests on PREPROD through a Jenkins job.
- When the new release is ready for production, the System Integrator opens a Deploy Request on Jira, triggering the opening of a Change Request on ServiceNow.
- The project manager approves the Change Request.

- At the deployment time specified in the request, ServiceNow invokes Jenkins, which performs the automated deployment job on PROD (Flyway, OCP).

16.2.5.2. ENVIRONMENT VARIANTS

The standard environments for applications are **development, test, pre-production, and production**, as described earlier.

However, some applications may use fewer or more environments depending on specific needs.

A use case for fewer environments may be an application that, for specific reasons, does not require automated end-to-end testing and needs to be tested in an integrated environment. In this case, the application can utilize the development, pre-production, and production environments without the testing environment.

Use cases for multiple environments are applications that require separate spaces for testing and evaluation by different user groups. Examples of such cases are additional QA environments, a training environment for end users, and specific environments for testing production hotfixes.

16.2.5.3. INTERMEDIATE QA ENVIRONMENT

Introducing an additional intermediate QA environment between TEST and PREPROD can be a solution when there is a need to separate project QA team activities from end user User Acceptance Testing (UAT) activities.

This allows PREPROD to remain stable while the team tests new versions in the QA environment at a faster pace.

To be used in this manner, the QA environment should still have the characteristics of a pre-production environment, including integration with external systems.

It is recommended to introduce additional environments like QA only if necessary for project QA/UAT processes. Adding environments entails increased resource usage and management overhead. In normal cases, it is advised to use a single PREPROD environment for internal QA and UAT.

Figure 225 illustrates how the QA environment would be inserted into the CI/CD pipeline.



Figure 225: CI/CD pipeline with QA Environment

16.2.5.4. HOTFIX ENVIRONMENTS

Hotfix environments are optional environments that can be requested in addition to the standard environments in cases where an application has long release cycles (e.g., monthly or longer). During the time between one feature release and the next, there may be a need to release only bug fixes without introducing any other changes besides the bug fixes.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

In such cases, the hotfix environment allows for a separate and dedicated space to exclusively handle hotfixes, without interfering with new developments for the upcoming feature release, which continue in parallel on the main environment chain.

On the other hand, it is not necessary to introduce hotfix environments when the application follows frequent releases that include both features and bug fixes, with a single release chain (and therefore a single set of environments).

Even in cases where hotfix environments are utilized, they serve their purpose only during the period when a hotfix is being developed and tested, while they will not be used for most of the time. Therefore, they should be managed as on-demand environments, instantiated automatically by the CI/CD pipelines when needed and terminated at the end of the activities. This allows for optimization of infrastructure resource usage. Within Jenkins, hotfix environments will be associated with specific pipelines linked to hotfix branches. Figure 226 illustrates an example with hotfix environments for development and testing.

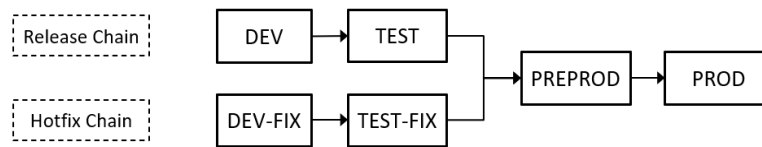


Figure 226: Hotfix environments - Example

16.2.5.5. DEVOPS ENGINEER ROLE

For the successful implementation of the DevOps process, it is strongly recommended that the project team includes one or more members who will take on the role of a DevOps Engineer. The DevOps Engineer is a role that combines development and system administration skills, which are utilized to automate build, deployment, and testing processes. Within the team, the DevOps Engineer is responsible for managing Jenkins, the CI/CD pipeline, and deployment manifests for automation. They also assist the team in making design and tool choices related to DevOps. The DevOps Engineer should adhere to the company's guidelines regarding acceptance criteria to be implemented in the pipeline steps (e.g., no critical vulnerabilities found, minimum required performance level, etc.). Additionally, the DevOps Engineer should align with the infrastructure team on topics such as deployment guidelines, environments, and infrastructure features relevant to the application.

Table 89: Division of responsibilities between Developers, QA, and DevOps Engineers

Activity	Owner
Source Code	Developer
Unit/component automatic test	Developer
Deployment Manifest	DevOps Engineers [+ Developer]
Jenkins and pipelines CI/CD	DevOps Engineers
Integration automatic test e2e	Testers (team QA)
Test book/manual test	Testers (team QA)

16.3. DEVELOPMENT GUIDELINES

16.3.1. MULTI-REPOSITORY STRUCTURE

- A single Bitbucket project per application with multiple repositories.
- Separate repositories for each microservice.
- Dedicated repositories for shared libraries used across different microservices.
- A repository for common CI/CD pipelines and integrated deployment of the application (Includes OpenShift manifests, configuration files, and any supporting scripts)

16.3.1.1. WHAT TO VERSION AND WHAT NOT TO VERSION:

OK to version: everything needed for building and deploying the application, text-based documentation (e.g., Markdown). Source code, build scripts, dependency manifests, binary assets like icons and UI images, deployment manifests, commonly shared support scripts, README.md.

Not OK to version: everything produced by the build, local configurations, binary documents. Application binaries, dependency binaries, IDE project files, developer-specific configurations with local paths and addresses, Word and PowerPoint documents that should reside in a document repository.

Note: modern build systems download dependencies from external repositories like the Maven Central Repository. In the (rare) case of using libraries not available in public repositories, upload them to internal Nexus to manage them without committing them to Git.

16.3.1.2. COMMITTING

Always reference the associated Jira issue in commit messages. Links the change to the issue, including direct Jira links to verify impact. Split complex changes into multiple smaller commits. Helps easily understand the purpose of each individual change. With Git, commits are only published after pushing.

16.3.1.3. README

Use Markdown files (README.md) to provide developer-specific information about the project. Describe what the component does and provide basic information on how to build and configure it. Optionally, include deployment information for different environments and references to other documentation sources (wikis, etc.).

16.4. CLOUD NATIVE DESIGN

Cloud Native Design embodies a cutting-edge and adaptable strategy for designing, developing and delivering software applications and services within the cloud computing ecosystem. Cloud Native Design enables organizations to adopt an agile, robust and highly flexible development model. Basically, It revolves around several core principles:

16.4.1. MICROSERVICES AS INDEPENDENT PROCESSES

1 microservice = 1 process. Independent process with an embedded server (e.g., Spring Boot with embedded Tomcat). Avoid using application servers with multiple WAR files (e.g., traditional JBoss or WebLogic).

16.4.2. ONE PROCESS PER CONTAINER

Exceptions may exist for support processes, but they are rare.

16.4.3. STATELESS MICROSERVICES

Store state in external services such as databases, queues, and key-value stores. Avoid saving data to local ephemeral storage unless it's temporary. Each instance of a microservice can scale horizontally (add or remove instances) or be moved to another node (deletion and recreation).

16.4.4. EXTERNALIZED CONFIGURATION

Make applications configurable through environment variables or configuration files. Configurable through ConfigMaps and Secrets (i.e. in OpenShift). Applications should include reasonable default values wherever possible to simplify configuration. Never commit credentials to the repository. Passwords, certificates, etc., should be configured as environment variables or files through OpenShift Secrets.

16.4.5. LOGGING

Write all logs to stdout instead of files. Allows automatic collection with EFK (Elasticsearch, Fluentd, Kibana). Use a logging library that supports logging levels and externalized configuration - Examples: SLF4J/Logback/Log4j2 for Java.

Define and use a parsable logging format to extract various fields. Include relevant information depending on the application. Mandatory fields: timestamp, log level, and message. Add tags to log lines (e.g., "[component1]") if necessary to differentiate different flows. EFK can parse logs to extract tags and separate visualization in dashboards.

16.4.6. READINESS AND LIVENESS PROBES

Always define Readiness and Liveness Probes for your deployments (i.e. in OpenShift). Allows (OpenShift) to determine when a pod is ready to respond to requests (readiness) and when it is functioning correctly (liveness). OpenShift automatically restarts pods that repeatedly fail the liveness probe, ensuring fast service recovery in case of crashes.

16.4.7. RESOURCE REQUESTS AND LIMITS

Always define Requests and Limits with the necessary resources for your deployments. Proper definition of requests and limits enables efficient coexistence between different applications. The scheduler determines the optimal placement of pods on cluster nodes based on declared resource usage with requests and limits. Properly defining requests and limits improves the stability and performance of both the application and the entire cluster⁶³.

16.5. BRANCHING MODELS

Choosing the right branching model for code management within repositories is crucial to ensure orderly collaboration among team developers and to meet the project's quality goals and release practices. The following text describe some

⁶³ <https://12factor.net/>

well-known branching models that are compatible with the DevOps methodology. The development team will need to choose which model to adopt based on the project's needs.

16.5.1. REQUIREMENTS FOR RELEASE FREQUENCY

i. Infrequent releases (monthly or longer)

- The main development branch can be in an “un-releasable” state between releases.
- It should be possible to make hotfixes in production without releasing features still in development.

ii. Frequent releases (daily or weekly => Continuous Delivery)

- The main development branch must always be stable and releasable.
- Hotfixes can be released directly from the main development branch.

iii. Infrequent releases (monthly or longer)

- Gitflow
- Trunk-Based Development with release branches

iv. Frequent releases (daily or weekly => Continuous Delivery)

- Trunk-Based Development with releases from the trunk

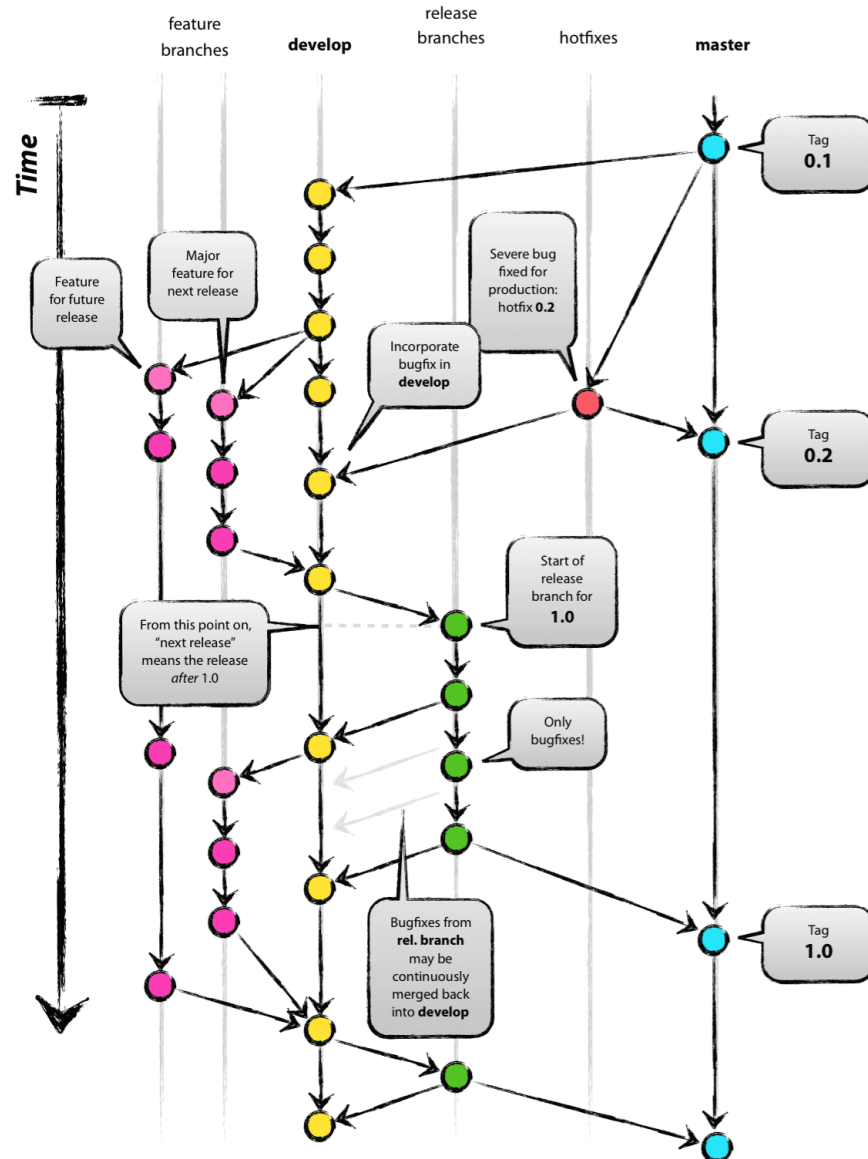


Figure 227: Git Branching Model [97]

16.5.1.1. GITFLOW - FEATURES

Master is not used for development and always contains the latest released version.

- Develop is the main development branch where changes are integrated.
- All changes happen on feature branches associated with their respective Jira issue.
 - When the feature is ready, the developer opens a pull request (PR).
 - Once the PR is accepted, the branch is merged into develop and then deleted.
- When develop contains all the features for the next release in a stable state, a release preparation branch is created.
 - This branch is used for any last-minute bug fixes and version number updates.

TRUSTEE D4.1 TRUSTEE Automation, OPEN CONTINUUM & SSI Homomorphic capable framework

- Fixes are made on a separate bugfix branch with a PR for merging into release and subsequent deletion of the bugfix branch.
- The release branch allows work to continue develop without impacting the release.
- Every fix made on the release branch must be brought back into develop through a merge.
- During the release, the following actions are performed: merging the release branch into master and develop, version tagging on master, and branch deletion.
- Critical fixes for production are handled by creating a hotfix branch from master.
 - The version number for the hotfix release is modified on the branch.
 - Once the fix is ready, a PR is opened.
 - When the PR is accepted, the branch is merged into master and develop, version tagging is done on master, and the branch is deleted [97] [98].

Note: Within these guidelines, the term "master" is commonly referred to as the main development branch. If using GitFlow, the reference should be understood as the "develop" branch.

GITFLOW - PROS

- Well-known and well-defined model
- Feature branches allow for choosing when to integrate various developments into develop
- Feature branches isolate develop from instabilities caused by major changes
- Use of release branches avoids or reduces code freezes on develop during the pre-release period
- Suitable for managing long release cycles with new developments vs. production fixes

GITFLOW - CONS

- Requires initial adaptation for developers to handle various branches and the new role of the develop and master branches (compared to the classic master as the default branch)
- Long-lived feature branches (e.g., 1 week or more) delay integration with other developments and have a higher likelihood of generating conflicts when merging into develop, with a risk of release delays
- Mitigate by breaking down the work into smaller, faster-to-complete sub-features
- Merge from develop to import changes in case of long-lived branches.

16.5.1.2. TBD WITH RELEASE BRANCHES – FEATURES

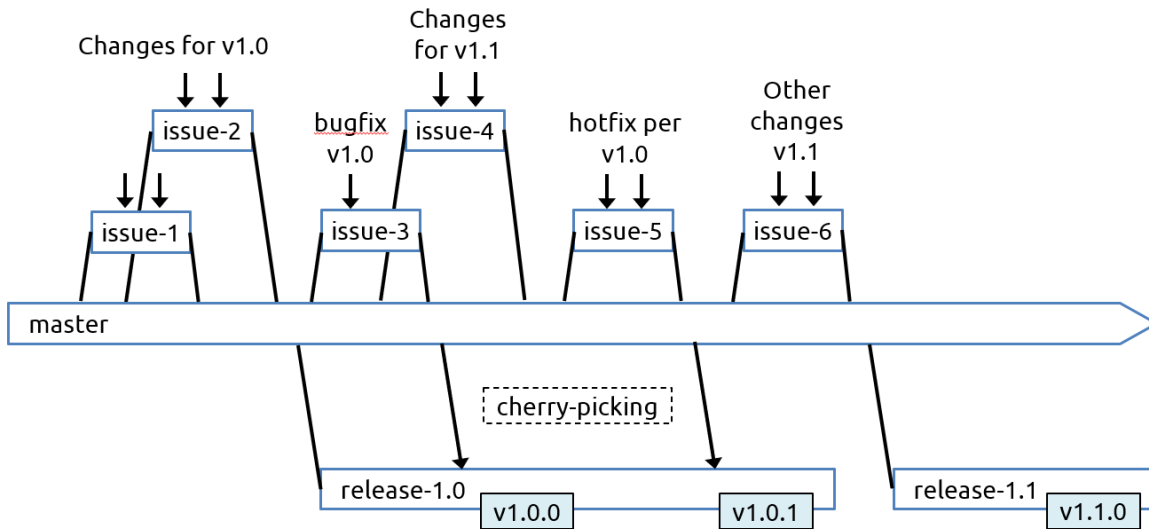


Figure 228: TBD with release branches

Master is the main development and bug fixing branch, but it is modified only through PRs. The developer works on short-lived topic branches (max 2 days) related to their respective Jira issue. After completing the work, the developer integrates the latest changes from master (pull or rebase) and opens a PR. The PR should be promptly reviewed (within a few hours) to avoid slowing down the work. Once the PR is accepted, it is merged into master and the topic branch is removed. Multiple topic branches (and PRs) can be opened for a single issue if it is complex and needs to be addressed in sub-tasks. Release branches are maintenance branches created before releases. They are used only for pre-release bug fixes and hotfixes for production with point releases, no new features. Create release branches as late as possible when master is feature-complete and stable to minimize commits to be brought from master to release. After branching, work on master can continue without impacting the release. All fixes are made on master and brought into the release branch through cherry-picking (no merge). Direct modifications on release branches are allowed only when fixing on master is not possible. For example, divergent branches or bugs do not present on master due to subsequent changes.

The bug should always be fixed on master as well if present (cherry-pick or independent fix). Only lead developers/release managers can bring fixes to release branches. Each active release branch has its own CI/CD pipeline and related test environments. Old release branches can be deleted when transitioning to subsequent releases [99].

TBD WITH RELEASE BRANCHES - PROS

- Simple to use as developers always work on branches based on master.
- Short-lived topic branches facilitate continuous integration on master and avoid merge conflicts that can occur when using long-lived feature branches.
- Since fixes are made on master, it is impossible to have regressions due to forgetting to port hotfixes from release to master.
- Use of release branches avoids or reduces code freeze on master during the pre-release period and allows for managing bug fixes on releases spaced over time.

TBD WITH RELEASE BRANCHES - CONS

- Requires discipline in organizing work into short and self-contained tasks that can be immediately integrated into master without breaking the build.

16.5.1.3. TBD WITH TRUNK-BASED RELEASES – FEATURES

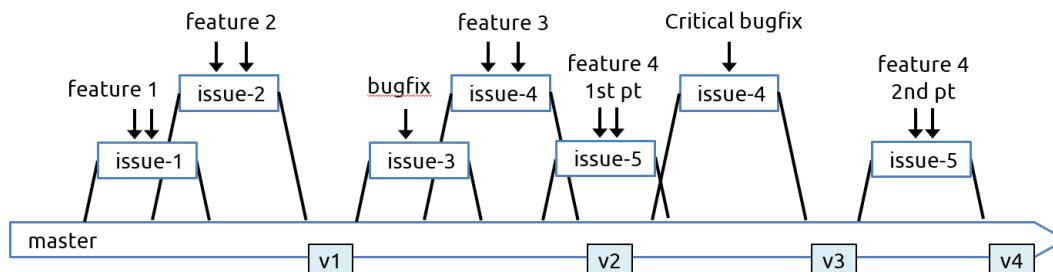


Figure 229: TBD with Trunk-Based Releases

- Master is the only branch used.
- Master is always releasable at any time.
- Hotfix releases also include all new features integrated in the meantime.
- Every change is made through a topic branch and its corresponding PR for merging into master.
- PRs with code review and essential automated tests ensure the stability of master.
- Long and complex changes should be approached progressively, dividing them into smaller and non-disruptive interventions, possibly distributed over different releases, and enabling them and making them visible to users only at the end of the work.
 - Techniques like feature flags and branch by abstraction are used.
- This model allows true Continuous Delivery with daily releases.
- Release numbering is based on incrementals, timestamps, or commit hashes.
 - Semantic versioning loses its meaning because there is no longer a distinction between major/minor/patch.
- When using PRs, developers cannot directly push to the main development branch; instead, they push to a dedicated branch with the changes for the PR.
- Through the PR, the developer indicates that a task is complete and requests its review and integration into the main development branch.
- The use of PR allows for:
 - Code verification with automated CI builds (including unit tests and other checks like static analysis)
 - Code review by one or more other developers (e.g., lead/senior dev)
 - Possible iterations before merging if the reviewer requests changes
- The merge into master only occurs if the CI build is successful and the PR has been approved by one or more reviewers.
- The code review process should be fast to avoid introducing bottlenecks in the workflow.

TBD WITH TRUNK-BASED RELEASES - PROS

- Improved code quality thanks to code review.
- Code review facilitates quick onboarding of junior and new developers in general.

- PRs serve as a learning method through feedback from seniors.
- Master branch is always stable due to prior PR builds.
- Avoids costly errors and preserves quality in mature projects.

TBD WITH TRUNK-BASED RELEASES - CONS

- PRs with approval can slow down work when very rapid iterations are needed.
- For example, when creating a new product from scratch with a tight schedule.
- Possible compromise: automatic merge without review if CI build passes.
- PRs may be redundant with small teams and/or experienced developers.
- The use of PRs is a best practice due to the benefits of code review.
- Projects with tight deadlines or small teams and/or experienced developers may choose not to use them to avoid overhead.
- It is always recommended to use them in large teams with many junior developers and/or high turnover.
- Using PRs is crucial in the trunk-based branching model with continuous releases directly from trunk, as it ensures the stability of the master branch.

16.6. RELEASE MANAGEMENT - PRINCIPLES

In Continuous Delivery, every build is a potential release. Every artifact or image must be identified by a unique version number or tag when it is built at the beginning of the pipeline. This requirement does not apply to artifacts that are never published independently but only within container images; in that case, the version of the image is considered. Never recompile (e.g., for version changes or compilation settings) within later stages of the pipeline; only by keeping the artifact/image built at the beginning unchanged can we ensure that it has passed all the tests in the various stages. If it passes all the tests in the pipeline, the build can be released. All environment changes required for a new release (configuration, DB schema changes) should be automated as much as possible and versioned. Credentials and secrets (passwords, private keys, etc.) are exceptions and cannot be versioned, but they must be documented and configured by the responsible individuals for each environment. Upgrade steps are automatically validated by deploying to various environments in the pipeline.

