



**Privacy compliant health data as a service for AI development**

*Grant Agreement Number: 101095384*

**D5.3: Data Hub Design and Data Market v1**

<b>Deliverable Identifier:</b>	D5.3
<b>Deliverable Version:</b>	v.1.0
<b>Status</b>	Final (F)
<b>Work Package:</b>	WP5 Health Data Hub
<b>Task:</b>	Task 5.2 Data Hub Design
<b>Author(s) and Organisation:</b>	Geert Machtelinckx (FJBE), Nikolaos Saklampanakis (FJBE), Dirk Pauwels (FJBE), Krzysztof Saja (FJBE)
<b>Peer Reviewer(s):</b>	Matti LESKINEN (VARHA), Arho VIRKKI (VARHA), Artur ROCHA (INESC TEC), Helder OLIVEIRA (INESC TEC), Gonçalo GONCALVES (INESCTEC)
<b>Deliverable Due Date:</b>	2025/01/31
<b>Deliverable Submission Date:</b>	2025/02/07
<b>Dissemination Level:</b>	PU: Public
<b>Funding Authority:</b>	European Commission
<b>Funding Program:</b>	Horizon Europe Health Work Programme 2021 – 2022
<b>Topic:</b>	HORIZON-HLTH-2022-IND-13-02
<b>Rights:</b>	PHASE-IV-AI Consortium

## Document Control History

Version	Date	Edited by	Modification reason
v.0.1	2024/11/15	Geert Machtelinckx	1 <sup>st</sup> draft
v.0.2	2024/12/16	Geert Machtelinckx	2 <sup>nd</sup> draft
v.0.3	2024/12/24	Nikolaos Saklampanakis	Internal review of document
v.0.4	2025/01/09	Dirk Pauwels	Internal review of document
v.1.0	2025/01/10	Geert Machtelinckx	Version for consortium review
v.1.1	2025/01/12	Geert Machtelinckx	Figure reference updates
v.1.2	2025/01/17	Geert Machtelinckx	Integration remarks VARHA - consortium review
v.1.3	2025/01/22	Geert Machtelinckx	Integration of remarks part 2 VARHA – consortium review
v.1.4	2025/01/24	Krzysztof Saja	Internal review, comments and improvements

## Executive Summary

This document is the first of two versions of the design documentation for the Health Data Hub envisioned for PHASE IV AI. The project aims to set up an infrastructure that allows the development of AI Models in the Health Domain, addressing the challenges imposed by legal constraints such as GDPR and AI Act, due to the sensitive nature of medical data.

To overcome these constraints while maintaining scalability for AI model creation, the need for federated and diverse infrastructures among participants has been identified. Multi-Party Computation and Homomorphic Encryption require computation activity to take place. Moreover, data transformation requires storage resources to be accessible on this infrastructure for the generation of AI Modes for medical purposes. The Health Data Hub presented in this document, has been designed while considering these factors.

The Health Data Hub is designed to address the security and privacy challenges related to health data, the computing resource flexibility needs of Federated Learning and Multi-Party Computation workflows that it is expected to meet, as well as the orchestration of the data exchange among partners.

Not only should we implement a secure-by-design architecture, we also should evidence that the result has been obtained using these very high standards. Therefore, we will explore the world of Self-Sovereign Identity (SSI) and Verifiable Credentials (VCs) to certify the identities of organizations as well as their role in the process. SSI and VCs are state of the art when it comes to identity and access management in decentralized architectures and commonly used in dataspace. Additionally, certification of offered solutions and services will be enabled when handling sensitive data, e.g. de-identification, data harmonization, model training.

Health Data Hub should also provide a federated marketplace, enabling the secure exchange of data assets against financial instruments. It's important to stress the federated aspect of what we need to build, as the idea is that all contributing actors participate, all can manage their own data in a sovereign way and have an equal level of influence. There should not be any centralized intermediary or mandatory organization in the middle, as this substantially complexifies the overall process to make work a federated architecture, while simplification is key to make the already complex federated process successful.

Bringing all above capabilities together will add a feature of the hub to make services available that allow the preprocessing of the data locally before it can be used in machine learning and model training.

Finally, we need a feature that provides transparency on all running software activity in an auditable way, however without revealing any sensitive data.

The different requirements lead us in the direction of a 'Decentralised Physical Infrastructure Network', DePIN, because it seems addressing all the requirements expected from the Health Data Hub: federated, secure-by-design, privacy-preserving, supporting financial transactions, supporting all compute activities needed to build AI models.

The current design document contributes to the fulfilment of the project's milestone MS2 'Use Cases Manual and Minimum Viable Product (MVP)', achieving TRL4, having narrowed the possible options in the complete system. It is aligned with D2.7 on Initial Architecture and the "C4 model" implemented in there.

This document is the initial version of the document describing Data Hub Design and Data Market and contains many paths for further explorations. We intend to conclude these explorations on all topics up to month M31, and the results will be presented in deliverable D5.4. The ambition is to achieve TRL6 by then, which is an engineering-scale model.

### *Disclaimer*

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the PHASE-IV-AI consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the PHASE-IV-AI Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the PHASE-IV-AI Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

### *Copyright message*

©PHASE-IV-AI Consortium, 2023-2026. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

## Table of Content

<b>1. Introduction .....</b>	<b>9</b>
1.1 Purpose of the document .....	9
1.2 Insights from other Tasks and Deliverables .....	9
1.3 Reference documents.....	9
1.4 Definitions .....	14
1.5 Structure of the document.....	15
1.6 List of Acronyms .....	15
<b>2. Baseline Concepts .....</b>	<b>17</b>
2.1 Concepts .....	17
2.1.1 Data Associated Technologies.....	17
2.1.2 Centralized, Decentralized, & Federated Data Approaches .....	19
2.2 Challenges related to Health Data .....	21
2.2.1 Technical Challenges.....	22
2.2.2 Business & Organizational Challenge .....	23
2.2.3 Legal Compliance Challenges .....	23
<b>3. Business Process Workflow .....</b>	<b>24</b>
3.1 Onboarding process .....	24
3.2 Register Offered Data and Services.....	25
3.3 Deployment of Virtual Data Center.....	26
3.4 Deployment of Pre-processing Services handling sensitive data .....	26
3.5 Anonymized Data Set Registration on HDH.....	27
3.6 Anonymized data set requested for modeling .....	28
3.7 AI model and services available for consumption.....	28
<b>4. PHASE IV AI Health Data Hub Design .....</b>	<b>29</b>
4.1 Context for Health Data Hub.....	29
4.2 Health Data Hub Container Diagram and Description .....	31
4.3 Identity Access Management System Container Diagram and Description.....	34
4.4 Data Consumer Workflow from Health Data Hub System Perspective .....	35
4.5 Technology Choice and rationale .....	38
4.6 Key Components Description.....	39
4.6.1 DePIN Hardware Infrastructure.....	39
4.6.2 Data markets.....	42
4.6.3 Catalog for Data Assets and Services .....	43
4.6.4 Services deployment technique using flist .....	45
4.6.5 Identity and Access Management.....	45

---

4.6.6	Interconnected Hardware Infrastructure reserved for PHASE IV AI.....	50
4.6.7	User Wallet.....	54
4.6.8	UI Portals.....	54
4.6.9	Infrastructure as Code (IaC) Providing Deployment Services .....	54
4.7	Suggested Approach during the Project .....	55
<b>5.</b>	<b>Conclusions .....</b>	<b>56</b>
<b>6.</b>	<b>Annex A: Reference Architectures and DePIN.....</b>	<b>57</b>
6.1	Significance of Reference Architectures .....	57
6.2	Overview of Reference Architectures for Data Marketplaces and Data Spaces .....	58
6.2.1	Simpl.....	58
6.2.2	Threefold Tech / DePIN .....	59
6.2.3	IDS.....	64
6.2.4	GAIA-X.....	67
<b>7.</b>	<b>Annex B: Private Overlay Network Technologies.....</b>	<b>71</b>
7.1	Private Overlay Network technologies.....	71
7.1.1	Mycelium.....	71
7.1.2	Yggdrasil .....	71
7.1.3	Tailscale.....	72
7.1.4	Netmaker .....	72
7.1.5	Netbird.....	72
7.1.6	Twingate.....	72
7.1.7	Zrok.io .....	73
7.1.8	OpenZiti.....	73
<b>8.</b>	<b>Annex C : Flist creation steps.....</b>	<b>74</b>
<b>9.</b>	<b>Annex D: Bootstrapping a Threefold node .....</b>	<b>80</b>
9.1	An easy way to make hardware available to a secure network .....	80
9.2	Eligible Hardware.....	80
9.3	Bootstrap image.....	81
9.3.1	Download the Zero-OS Bootstrap Image .....	82
9.3.2	Burn the Zero-OS Bootstrap Image.....	84
<b>10.</b>	<b>Annex E: Zero-OS federated operating system.....</b>	<b>85</b>

## Table of Figures

Figure 1 – Health Data Hub concept .....	17
Figure 2 – Relevance of existing data management associated technologies.....	19
Figure 3 – Difference between centralized, decentralized, and federated data approaches .....	20
Figure 4 – HDH Onboardng Process.....	24
Figure 5 – HDH Registration of Data Sets and Services.....	25
Figure 6 – HDH Deployment of Virtual Data Center.....	26
Figure 7 – HDH Deployment of Preprocessing Services .....	27
Figure 8 – HDH Anonymized Data Set Registration .....	27
Figure 9 – HDH Data Set Request Process for modeling.....	28
Figure 10 – HDH AI Model publication .....	28
Figure 11 – System Context C4 Diagram of PHASE IV AI .....	30
Figure 12 – Container C4 Diagram of PHASE IV AI – Part Health Data Hub.....	32
Figure 13 – Container C4 Diagram of Identity Access Management System of PHASE IV AI .....	34
Figure 14 – Data Consumer Workflow in Health Data Hub (Container View Sequence Diagram) .....	36
Figure 15 – Difference between centralized, decentralized, and federated data approaches .....	40
Figure 16 – Topology of a mesh network.....	41
Figure 17 – Data and Services Reservation Process Flow Diagram.....	48
Figure 18 – Sequence diagram of Flist usage.....	49
Figure 19 – Network Wall in a Secure Overlay Network .....	52
Figure 20 – Mesh network topology.....	52
Figure 21 – Threefold key components.....	59
Figure 22 – Threefold Dispersed Storage.....	60
Figure 23 – Threefold Network Wall .....	61
Figure 24 – Threefold Web GW .....	62
Figure 25 – Threefold Smart Contract for IT .....	63
Figure 26 – General structure of IDS Reference Architecture Model.....	65
Figure 27 – Interaction of technical components .....	66
Figure 28 – GXFS supported elements.....	67
Figure 29 – Plug-and-Play Hardware Infrastructure Provisioning .....	80
Figure 30 – Setting up a farm and a bootstrap node .....	82
Figure 31 – Zero-OS Bootstrap .....	82
Figure 32 – Zero-OS Bootstrap set-up interface.....	83
Figure 33 – Zero-OS Bootstrap Image setup – Image format EFI .....	83

---

Figure 34 – Zero-OS Bootstrap Image setup – Image formats ISO and USB.....	84
--	----

## 1. Introduction

### 1.1 Purpose of the document

The objective of this deliverable is to introduce a design for the PHASE IV AI Health Data Hub. It emphasizes the presentation of the structuring principles of the Solution Architecture (as explained in project deliverable D2.7), building on experience, recommendations, guidelines, and design principles from Reference Architectures (RAs) of existing Data Infrastructures (e.g. DePIN), Data Marketplaces (e.g. FAME) and Data Spaces (e.g., IDS, Gaia-X, EHDS and Health-X).

Prior to that, the deliverable aims to present the PHASE IV AI value proposition, its stakeholders, the addressed challenges, and the provided opportunities, plus legal and non-functional requirements (concerning security, data privacy as well as compatibility with existing IT standards). The involved components are thoroughly described and technically specified, providing their added value to the overall Solution Architecture. Hence, the deliverable presents the design of the Health Data Hub aligned with the PHASE IV AI Initial Solution Architecture of PHASE IV AI (D2.7). The design is a more detailed elaboration of the Data Hub component and will drive the technological developments of the HDH implementation, as well as the design and initial integration of the underlying use cases. As part of these development and integration processes, feedback on the relevance and appropriateness of the suggested Data Hub design will be solicited. This feedback will be taken into account in the development of the final version of the PHASE IV AI architecture as part of the deliverable D2.9 – ‘PHASE IV AI Final Architecture’ that will be released at M24 of the project, as well as in the final version of the Data Hub Design document, deliverable D5.4 that will also be released at M31 of the project.

### 1.2 Insights from other Tasks and Deliverables

The PHASE IV AI Health Data Hub Design Document represents an important milestone for the efforts to be done for WP5. Specifically, it provides the structuring principles that will drive the design and development of a Health Data Space, enabling the deployment of AI models in a secure, privacy-preserving and federated way, considering current Data Spaces’ challenges and opportunities, as well as the wider requirements of the health sector.

Other than that, the current deliverable provides insights and technical specifications for Work Packages WP3 and WP4, as well as the use case related to WP6. It must be noted that while the design and development of these other components are independent from each other, the HDH design will also have an impact on these deliverables, as all contribute to an overarching architecture as defined in WP2. It also intends to help the other WPs in achieving the challenging requirements that exist in terms of security, privacy, legal aspects, availability, auditability, and overall applicability.

### 1.3 Reference documents

1. [1 Data Hubs, Data Lakes, and Data Warehouses: How They Are Different and Why They Are Better Together, <https://www.gartner.com/en/documents/3980938>
2. Loukiala, A., Joutsenlahti, J. P., Raatikainen, M., Mikkonen, T., & Lehtonen, T. (2021). Migrating from a centralized data warehouse to a decentralized data platform architecture. In International Conference on Product-Focused Software Process Improvement, pp. 36-48.
3. Janssen, N. E. (2022). The Evolution of Data Storage Architectures: Examining the Value of the Data Lakehouse (master’s thesis, University of Twente).

4. European Data Strategy: Making the EU a role model for a society empowered by data, [https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy\\_en](https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy_en)
5. Centralized vs. Federated: State Approaches to P-20W Data Systems, [https://nces.ed.gov/programs/slds/pdf/federated\\_centralized\\_print.pdf](https://nces.ed.gov/programs/slds/pdf/federated_centralized_print.pdf)
6. Why Data Mesh is a Must for Federated Data Management, <https://www.k2view.com/blog/federated-data-management>
7. SGIC Webinar Part 3: The Operational Reality of Statewide Data Ambitions: Federated Models, <https://www.appgeo.com/part-3-the-operational-reality-of-statewide-data-ambitions-centralized-and-decentralized-models/>
8. Data Federation: Foolproof Guide on Data Management, <https://www.g2.com/articles/data-federation>
9. Understanding the Value of Reference Architectures, <https://doveltech.com/innovation/understanding-the-value-of-reference-architectures>
10. Bass, L., Clements, P., & Kazman, R. (2003). Software architecture in practice. Addison-Wesley Professional.
11. Nakagawa, E. Y., Oliveira Antonino, P., & Becker, M. (2011). Reference architecture and product line architecture: A subtle but critical difference. In European conference on software architecture, pp. 207-211.
12. Martínez-Fernández, S., Ayala, C. P., Franch, X., & Nakagawa, E. Y. (2015). A Survey on the Benefits and Drawbacks of AUTOSAR. In Proceedings of the First International Workshop on Automotive Software Architecture, pp. 19-26.
13. Valle, P. H. D., Garcés, L., Volpato, T., Martínez-Fernández, S., & Nakagawa, E. Y. (2021). Towards suitable description of reference architectures. PeerJ Computer Science, 7, e392.
14. Kruchten, P. B. (1995). The 4+ 1 view model of architecture. IEEE software, 12(6), pp. 42-50.
15. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. Present and ulterior software engineering, pp. 195-216.
16. European Cluster for Securing Critical Infrastructures, <https://www.finsec-project.eu/ecsci>
17. EU-CIP, <https://www.eucip.eu/>
18. SecureIoT, <https://secureiot.eu>
19. PolicyCloud, <https://policycloud.eu/>
20. International Data Spaces Associations, <https://internationaldataspaces.org/>
21. International Data Spaces, <https://www.fraunhofer.de/en/research/lighthouse-projects-fraunhofer-initiatives/international-data-spaces.html>
22. IDS Business Layer, <https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3-1-business-layer>
23. IDS Functional Layer, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_2\\_functionallayer](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_2_functionallayer)
24. IDS Information Layer, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_3\\_informationlayer](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_3_informationlayer)
25. IDS Process Layer, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_4\\_process\\_layer](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_4_process_layer)
26. IDS System Layer, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_5\\_0\\_system\\_layer](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer)
27. IDS Security Perspective, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4\\_perspectives/4\\_1\\_security\\_perspective](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4_perspectives/4_1_security_perspective)
28. IDS Certification Perspective, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4\\_perspectives/4\\_2\\_certification\\_perspective](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4_perspectives/4_2_certification_perspective)
29. IDS Data Governance Perspective, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4\\_perspectives/4\\_3\\_governance\\_perspective](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4_perspectives/4_3_governance_perspective)

30. IDS Connector, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_5\\_0\\_system\\_layer/3\\_5\\_2\\_ids\\_connector](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer/3_5_2_ids_connector)
31. IDS Data Connector Report, [https://internationaldataspaces.org/wp-content/uploads/dlm\\_uploads/IDSA-Data-Connector-Report-7\\_May-2023-2.pdf](https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-7_May-2023-2.pdf)
32. IDS Functional Requirements, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/idsa-rulebook/idsa-rulebook/3\\_functional\\_requirements](https://docs.internationaldataspaces.org/ids-knowledgebase/v/idsa-rulebook/idsa-rulebook/3_functional_requirements)
33. IDS Usage Control, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4\\_perspectives/4\\_1\\_security\\_perspective/4\\_1\\_6\\_usage\\_control](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/perspectives-of-the-reference-architecture-model/4_perspectives/4_1_security_perspective/4_1_6_usage_control)
34. IDS Usage Control Policies, <https://international-data-spaces-association.github.io/DataspaceConnector/Documentation/v5/UsageControl>
35. IDS Dataspace Protocol, <https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol>
36. Is X-Road a Data Space Technology? <https://www.niis.org/blog/2023/6/21/is-x-road-a-data-space-technology>
37. IDS Metadata Broker, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_5\\_0\\_system\\_layer/3\\_5\\_4\\_metadata\\_broker](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer/3_5_4_metadata_broker)
38. IDS Vocabulary Hub, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_5\\_0\\_system\\_layer/3\\_5\\_6\\_vocabulary\\_hub](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer/3_5_6_vocabulary_hub)
39. IDSA Information Model, <https://github.com/International-Data-Spaces-Association/InformationModel>
40. IDS Clearing House, [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3\\_5\\_0\\_system\\_layer/3\\_5\\_5\\_clearing\\_house](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer/3_5_5_clearing_house)
41. IDSA IDS-G, <https://github.com/International-Data-Spaces-Association/IDS-G>
42. IDS Meta Data Broker, <https://github.com/International-Data-Spaces-Association/IDS-G/blob/main/Components/MetaDataBroker/README.md>
43. IDS Clearing House (IDS-CH), <https://github.com/International-Data-Spaces-Association/IDS-G/blob/main/Components/ClearingHouse/README.md>
44. IDSA GitHub, <https://github.com/International-Data-Spaces-Association/IDS-G-pre/pull/26>
45. The International Data Spaces (IDS) Information Model, <https://github.com/International-Data-Spaces-Association/InformationModel/blob/develop/README.md>
46. IDS Deployment Scenarios, <https://github.com/International-Data-Spaces-Association/IDS-Deployment-Scenarios>
47. IDS Reference Testbed, <https://docs.internationaldataspaces.org/knowledge-base/ids-testbed-1>
48. GAIA-X, <https://gaia-x.eu/>
49. GAIA-X Federation Services, [www.gxfs.de](http://www.gxfs.de)
50. GXFS Authentication Authorization, <https://www.gxfs.eu/authentication-authorisation>
51. GXFS Organizational Credential Manager, <https://www.gxfs.eu/organizational-credential-manager/>
52. GXFS Personal Credential Manager, <https://www.gxfs.eu/personal-credential-manager/>
53. GXFS Trust Services API, <https://www.gxfs.eu/trust-services-api/>
54. GXFS Data Contract Service, <https://www.gxfs.eu/data-contract-service/>
55. GXFS Data Exchange Logging Service, <https://www.gxfs.eu/data-exchange-logging-service/>
56. GXFS Core Catalogue Features, <https://www.gxfs.eu/core-catalogue-features/>
57. GXFS Continuous Automated Monitoring, <https://www.gxfs.eu/continuous-automated-monitoring/>
58. GXFS Onboarding Accreditation Workflows, <https://www.gxfs.eu/onboarding-accreditation-workflows/>
59. GXFS Notarization API, <https://www.gxfs.eu/notarization-api/>
60. GXFS Tutorial, <https://www.gxfs.eu/portal/>

61. GXFS Orchestration, <https://www.gxfs.eu/orchestration/>
62. C4 Architecture Model, <https://c4model.com/>
63. Unified Modelling Language (UML), <https://www.uml.org/>
64. Kruchten, P. B. (1995). The 4+ 1 view model of architecture. IEEE software, 12(6), pp. 42-50.
65. C4 model for system architecture design, <https://icepanel.io/blog/2022-10-03-c4-model-for-system-architecture-design>
66. C4 Model for Software Architecture, <https://www.infoq.com/articles/C4-architecture-model/>
67. OpenDEI – Design Principles for Data Spaces, <https://h2020-demeter.eu/wp-content/uploads/2021/05/Position-paper-design-principles-for-data-spaces.pdf>
68. Data Spaces Business Alliance: Unleashing the Data Economy, [https://data-spaces-business-alliance.eu/wp-content/uploads/dlm\\_uploads/Data-Spaces-Business-Alliance-Technical-Convergence-V2.pdf](https://data-spaces-business-alliance.eu/wp-content/uploads/dlm_uploads/Data-Spaces-Business-Alliance-Technical-Convergence-V2.pdf)
69. Accelerating Business transformation in the Data Economy, <https://data-spaces-business-alliance.eu/>
70. Performant and modular APIs for Verifiable Data and SSI, <https://veramo.io/>
71. Verifiable Credentials Data Model v1.1, <https://www.w3.org/TR/vc-data-model/>
72. Decentralized Identifiers (DIDs) v1.0, <https://www.w3.org/TR/did-core/>
73. Yuan, E., & Tong, J. (2005). Attributed based access control (ABAC) for web services. In IEEE International Conference on Web Services (ICWS'05).
74. Spring Boot, <https://spring.io/projects/spring-boot>
75. Postgre SQL, <https://www.postgresql.org/>
76. Casbin, <https://casbin.org/>
77. Joinup Licensing Assistant, <https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant/jla-find-and-compare-software-licenses>
78. Solidity, <https://soliditylang.org/>
79. ExpressJS, <https://expressjs.com/>
80. EU Vocabularies, <https://op.europa.eu/en/web/eu-vocabularies/dcat-ap>
81. Natural Language Processing with Python's NLTK Package, <https://realpython.com/nltk-nlp-python/>
82. Elasticsearch Python Client, <https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/index.html>
83. TensorFlow, <https://www.tensorflow.org/>
84. PyTorch, <https://pytorch.org/>
85. SciKit Learn, <https://scikit-learn.org/stable/>
86. Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering, 16(9), pp. 1128-1142.
87. Van Der Aalst, W., & van der Aalst, W. (2016). Data science in action, pp. 3-23.
88. Proton, <https://github.com/ishkin/Proton/>
89. Shimizu, S. (2022). Statistical Causal Discovery: LiNGAM Approach.
90. ChatGPT, <https://chat.openai.com/chat>
91. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135-1144.
92. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.
93. Jimenez-Peris, R., Burgos-Sancho, D., Ballesteros, F., Patiño-Martinez, M., & Valduriez, P. (2022). Elastic scalable transaction processing in LeanXcale. Information Systems, 108, 102043.
94. JDBC Drivers, <https://www.geeksforgeeks.org/jdbc-drivers/>
95. Flower: A Friendly Federated Learning Framework, <https://flower.dev/>
96. ThreeFold QSFS, [https://threefoldfoundation.github.io/www\\_threefold\\_io/developer/qsfs/](https://threefoldfoundation.github.io/www_threefold_io/developer/qsfs/)

97. Simpl, [https://linktr.ee/ec\\_simpl?utm\\_source=linktree\\_profile\\_share&tsid=b2552ba7-1a00-41c3-aa52-3b429f3d4d02](https://linktr.ee/ec_simpl?utm_source=linktree_profile_share&tsid=b2552ba7-1a00-41c3-aa52-3b429f3d4d02)
98. Simpl Open-Source framework for data spaces, <https://code.europa.eu/simpl/simpl-open/development>
99. Decentralized Identity Foundation (DIF), <https://identity.foundation/did-traits/#abstract>
100. W3C did:web method, <https://w3c-ccg.github.io/did-method-web/>

## 1.4 Definitions

This Section introduces some key terms that are used across the document and the PHASE IV AI project in general. These key terms refer to the following:

- **Data Space:** A decentralized infrastructure for transparent and trustworthy data sharing and exchange in data ecosystems within a certain application domain, based on commonly agreed principles and capabilities, consisting of data platform(s), data marketplace(s), and data sovereignty.
- **Data Hub:** A data hub takes care of the *flow of data between source/target systems and users*. The goal is to indicate exactly what actions need to be performed with the underlying data, where systems can distribute data through the data hub. The data hub concept brings some structure in the integration between peers with whom data needs to be shared.
- **Data Platform:** An environment that facilitates the exchange of value between two or more parties, with the multiple parties interacting through the platform.
- **Data Marketplace:** A multi-sided place (i.e., platform) where data providers and data consumers can find each other to stimulate data exchange or access.
- **Data Sovereignty:** The capability of a person or organization to make all data-related decisions on their own.
- **DePIN:** Decentralised Physical Infrastructure Network, the decentralised technology aiming at creating decentralisation in the setup of a physical hardware network.
- **Data Hub:** An instantiated infrastructural component using data platform around a certain topic (in our case Health).
- **Federation:** A group of participants (e.g. multiple data producers and consumers, model producers and consumers) interconnected with agreed governance, access, and security rules.
- **Federated Security Management:** The capability of having individual Data Spaces security management associated with a federated collaboration on a global security management, considering access control, usage control, trust, and identity management.
- **(Data) Asset:** A data resource or artefact (e.g., system, application output file, document, database, web page, dataset, service, algorithm, AI/ML model, data insights, data visualizations, software, publications) that carries data which is relevant for the value chain of an organization or institution, or which has strategic or operative value to generate revenue through exchanging it.
- **(Asset) Offering:** An instance of a data asset, where the latter may contain one or multiple offerings simultaneously.
- **Asset Metadata:** Information about a data asset that helps to describe, structure, or administer that data asset, providing a structured reference that helps to sort and identify attributes of the information it describes.
- **Self-Sovereign Identity (SSI)** is an approach to digital identity that gives individuals control over the information they use to prove who they are to websites, services, and applications across the web.
- **Decentralised Identifiers (DIDs)** are a type of globally unique identifier that enables an entity to be identified in a manner that is verifiable, persistent (for as long as the DID controller desires) and does not require the use of a centralized registry. DIDs enable a new model of decentralized digital identity that is often referred to as a self-sovereign identity. They are an important component of decentralized web-applications.
- **Verifiable Credentials:** are digital credentials which follow the relevant [World Wide Web Consortium open standards](#). They can represent information found in physical credentials, such as a passport or license, as well as new things that have no physical equivalent, such as ownership of a bank account. They have numerous advantages over physical credentials, most notably the fact that

they're digitally signed, which makes them tamper-resistant and instantaneously verifiable. Verifiable credentials can be issued by anyone, about anything, and can be presented to and verified by everyone. The entity that generates the credential is called the *Issuer*. The credential is then given to the *Holder* who stores it for later use. The Holder can then prove something about themselves by presenting their credentials to a *Verifier*.

## 1.5 Structure of the document

This document is organized into several key sections to provide a comprehensive overview of the PHASE IV AI Data Hub Design. The structure is as follows:

1. **Baseline Concepts:** This section outlines the overall concept described in the document, some context around approaches of governing data, challenges of technical, legal, business and organisational nature.
2. **Business Process Workflow:** This sections explains high-level the different business process steps that we see to come to a trained AI model on health data, respecting all security and privacy measures as required by regulation.
3. **Technical Design Overview:** This section provides a detailed description of the technical design, specifying the Health Data Hub subsystem that has been described high-level in the D2.7 Initial Architecture deliverable (D2.7), including system context diagram and container diagrams. It also contains a data consumer workflow diagram from the subsystem perspective. Furthermore a rationale is explained about technology choices. The key components and technologies are described, mainly DePIN and Self-Sovereign Identity (SSI). We make a start in the elaboration on a data catalogue, and describe the service deployment technique that DePIN offers. We explain about where an Identity Management system using SSI can provide value to certify participants in their role and authenticity, software on their authenticity and behaviour and IoT devices on their authenticity. We also stress the importance to include the orchestration of hardware infrastructure in the framework to guarantee end-to-end security and privacy-preservation by design, and elaborate on the tools available in DePIN to fulfil this objective.  
We end the chapter with a suggestion about a realistic path towards a compliant realisation.
4. **Conclusions:** This final section summarizes the key points discussed in the document and outlines the next steps for the project.

## 1.6 List of Acronyms

List of Acronyms	
AI	Artificial Intelligence
AML	Anti-Money Laundering
BIOS	Basic Input/Output System
B2B	Business to Business
DaaS	Data as a Service
DePIN	Decentralized Physical Infrastructure Network
DID	Decentralized Identifier
CPU	Compute Processing Unit

List of Acronyms	
DID	Decentralized Identifier
DIH	Digital Innovation Hub
EHDS	European Health Data Space
FHE	Full Homomorphic Encryption
GDPR	General Data Protection Regulation
GPU	Graphical Processing Unit
HDD	Hard Disk Drive
HDH	Health Data Hub
HW	Hardware
ICT	Information and Communication Technology
IoT	Internet of Things
IP	Internet Protocol
KYC	Know Your Customer
MaaS	Model as a Service
ML	Machine Learning
MPC	Multi-Party Computation
OS	Operating System
PaaS	Platform as a Service
RAM	Random-Access Memory
RDMA	Remote Direct Memory Access
PPP	Public-Private Partnership
SaaS	Software as a Service
SSD	Solid State Drive
SSI	Self-Sovereign Identity
TFT	ThreeFold Token
UEFI	Unified Extensible Firmware Interface
USB	Universal Serial Bus
VC	Verifiable Credential
VDC	Virtual Data Centre
VP	Verifiable Presentation
VPN	Virtual Private Network
3Node	Hardware infrastructure with Zero-OS running as operating system on it

## 2. Baseline Concepts

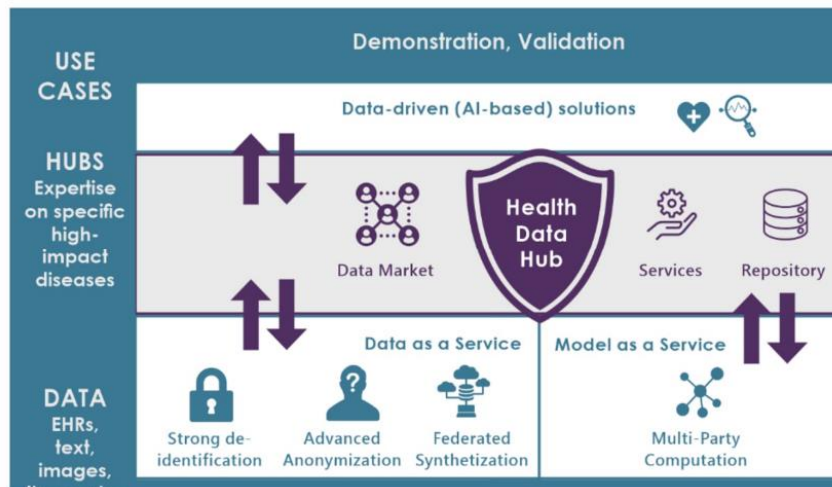


Figure 1 – Health Data Hub concept

### 2.1 Concepts

#### 2.1.1 Data Associated Technologies

Data and information are the most important assets for most organizations. While organizations can use their data to improve their businesses, their data can also have significant value beyond their business. At the same time, nowadays organisations are aware of both the competitive edge that they can gain by also incorporating external data into their data strategy and business models, trying to optimize their overall information processes and insights. It is undeniable that in such a data-driven business climate, data is playing a key role in capturing market intelligence and actionable insights to augment business operations. Hence, in today's data-driven world, organisations are constantly seeking innovative approaches to manage and harness the power of data. For this purpose, data management platforms, tools, and associated technologies (i.e., Databases, Data Warehouses, Data Lakes, Data Meshes, Data Hubs, Data Catalogues, Data Platforms, Data Ecosystems, Data Marketplaces, Data Spaces) are increasingly getting a global focus, having created a plethora of diverse options that lead to controversies and hot debates. Such fact is also verified by a Gartner study that has shown that more than 25% of customers thought that a Data Hub was a Data Lake solution [2]. Gartner's research illustrated how much confusion there is about what the different concepts entail, which is also intensively spotted in the research community [3][4]. While all the existing concepts aim to improve data capabilities within organisations, they differ in their fundamental principles, architectural design, and organisational impact.

The following paragraphs provide more clarity on the meaning of these terms and show how their concepts differ from each other, making clear their general scope and usage.

- **Database:** A database stores information from a *single data source* for one particular function of an organization, being able to process many simple queries upon the data quickly.
- **Data Warehouse:** A data warehouse stores large amounts of *structured data from multiple sources in a centralized place*. The goal of using a data warehouse is to combine disparate data sources (inside or outside of an organization) to analyse the data, look for insights, and create Business Intelligence (BI) in the form of reports and dashboards.

- **Data Lake:** A data lake stores data from *disparate sources that is stored in its original, raw format* (either structured, semi-structured or unstructured data format). The basic scope is to store raw data from all sources without the need to process or transform it at that time, allowing organizations to ingest and manage large volumes of data in an aggregated storage solution when they might not be entirely sure how they will use that data in the future (whether for business intelligence or data products).
- **Data Fabric:** A data fabric is a *centralized storage approach* to data management architecture, using automated/intelligent systems to connect *data stored in multiple places and in multiple formats*. In essence, a data fabric expands on the architecture of a data warehouse, including building blocks such as data access, discovery, transformation, integration, security, governance, lineage, and orchestration. The goal is to advocate for setting up a unified data layer to provide a single source of truth for data, ensuring data quality, consistency, and security, while allowing different end-users to access and manage data easily.
- **Data Mesh:** A data mesh is a *decentralised and domain-oriented storage approach* that enables collection, integration, and analysis of data from disconnected systems concurrently, so there is no need to pull in data from disparate systems into a single location and preprocess them for analysis. In essence a data mesh decentralizes data from a single source so that it can be readily available to multiple users. It is ideally targeted for a business environment where data needs to be integrated from many disintegrated systems or processes for fast analysis.
- **Data Catalogue:** A data catalogue primarily serves as a *unified inventory or directory of an organization's data* (much like a library catalogue for books), *maintaining metadata* about the data including their location, format, schema, usage, relationships, ownership, and quality among others. The goal is to act as an inventory or a metadata management system, enabling users to quickly find and access the data they need for their tasks.
- **Data Platform:** A data platform serves as the *ultimate storehouse for data coming from diverse sources* and *allowing the scaling of complex processing and analytics operations* that turn data into valuable insights. In essence, a data platform is a *software framework or environment* that provides a foundation for developing and running software applications. It can be thought of as a set of tools, libraries, and services that are used to create, deploy, and manage the built applications. Thus, it can be viewed as an integrated solution that encompasses the features of data lakes, data warehouses, data hubs, etc. towards the realization of the needed applications.
- **Data Ecosystem:** A data ecosystem refers to the *combination of enterprise infrastructure and applications* that is utilized to aggregate and analyse information. In essence it refers to all the programming languages, algorithms, applications, and general infrastructure that is used for *collecting, analysing, and storing data*. The goal is to enable organizations to better understand their data and take the proper decisions.
- **Data Marketplace:** A data marketplace is a system where *data* (e.g., raw data, processed data, analytics-ready data products) can be *bought, sold, or exchanged* between organizations and/or individuals.
- **Data Space:** A data space is a *decentralised infrastructure for trustworthy data sharing and exchange* in data ecosystems, based on *commonly agreed principles*. Into this context, data is not stored centrally but rather at the source, and thus only transferred as necessary, supporting a decentralised nature that allows actors to keep the sovereignty on the data. In essence, a data space brings together relevant data infrastructures and governance frameworks to facilitate data pooling and sharing. The goal is for the data to stay with the providers and made available via secure peer-to-peer communication with common semantics and data sovereignty.
- **Data Hub:** A data infrastructure allowing participants to be interconnected and to structurally exchange information around a certain topic (in our case Health). A data hub takes care of the *flow of data between source/target systems and users*. The goal is to indicate exactly what actions need to be

performed with the underlying data, where systems can distribute data through the data hub. The data hub concept brings some structure in the integration between peers with whom data needs to be shared.

As relevance of data grows every year, the European Commission chose for the development of Data Spaces, envisioned as being of strategic importance for the growth of the European data economy [5]. The aim is to enable and stimulate the development of Data Value Chains, keeping sovereignty and trustworthiness under European premises and values. Figure 1 depicts the relationship and the relevance among all the abovementioned concepts, highlighting the positioning of Data Spaces that is further investigated and adopted in the context of PHASE IV AI.

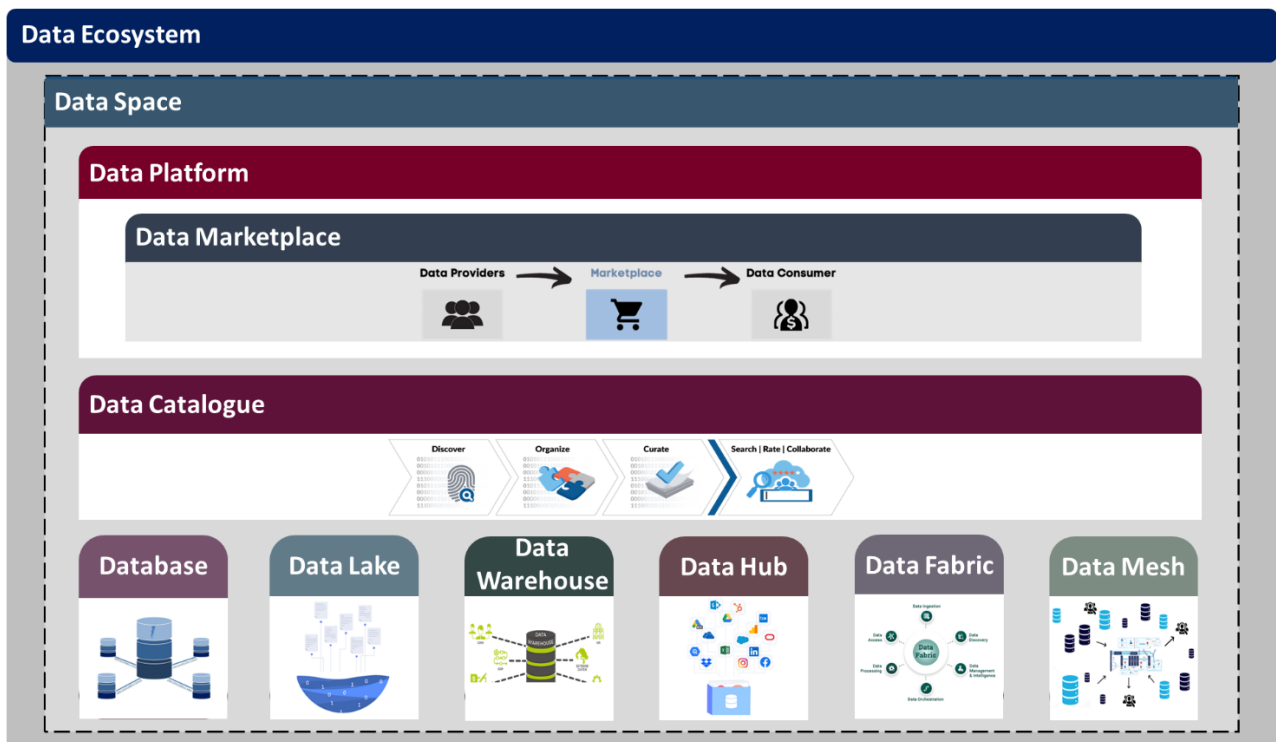


Figure 2 – Relevance of existing data management associated technologies

### 2.1.2 Centralized, Decentralized, & Federated Data Approaches

In today’s data-driven world, organizations face critical decisions regarding the storage and management of their valuable data. One fundamental choice for performing such tasks is to select the application of a centralized, decentralized or federated data architecture. Each approach has its merits, and understanding the advantages and disadvantages can help organizations to make informed decisions that align with their unique needs and goals.

The following paragraphs provide more insights on the meaning of those three (3) different data architecture types, outlining their overall idea and concepts in the context of being applied into any kind of Data Platform, Data Space, etc. [6][7][8].

- **Centralized Data Approach:** A centralized data approach refers to the practice of *storing data under a common centralized governance*, typically within a data centre or a cloud environment. Thus, all the participating source systems copy their data to a single, centrally governed data repository where they are organized, integrated, and stored using a common data standard/formatting.

- Distributed Data Approach:** A distributed data approach involves *distributing data across multiple locations or systems*, where *each system performs its own actions* upon the data, finally contributing its results/data into a central place. However, since each system is not interacting with each other, it is not aware of the processes/standards/formatting applied upon the data. In essence, in a decentralized data approach every system makes its own decisions, and the result of the underlying Data Platform is the aggregate of the decisions of the individual systems.
- Federated Data Approach:** A federated data approach involves *multiple individual systems that maintain control over their own data* but agree to *share some or all information to other participating systems upon request*. Users of the system submit questions through a common intermediary interface, which searches the various source systems. In essence, following a federated data approach allows an organization to leave its data where it is, using a common (single source) platform to provide a unified view of the data. Through this way data consumers can retrieve information from multiple, disparate systems with a single query, in real time.

Choosing between a centralized, decentralized or federated data approach is not a one-size-fits-all decision. However, nowadays federated data management has emerged as an effective solution for managing raw data and empowering organizations and data consumers to put valuable data to use [9]. All of the involved parties are able to (i) maintain ownership of whatever data they produce, (ii) aggregate and standardize their data and then make it available on behalf of the data owners, and (iii) perform their data actions by themselves or allow data consumers to edit their data with their approval. Figure 3 depicts the different applied ideas among the abovementioned data approaches, highlighting the concept of the federated data approaches that are deeply investigated and adopted in the context of PHASE IV AI towards realizing the creation of a Federated Data Space.

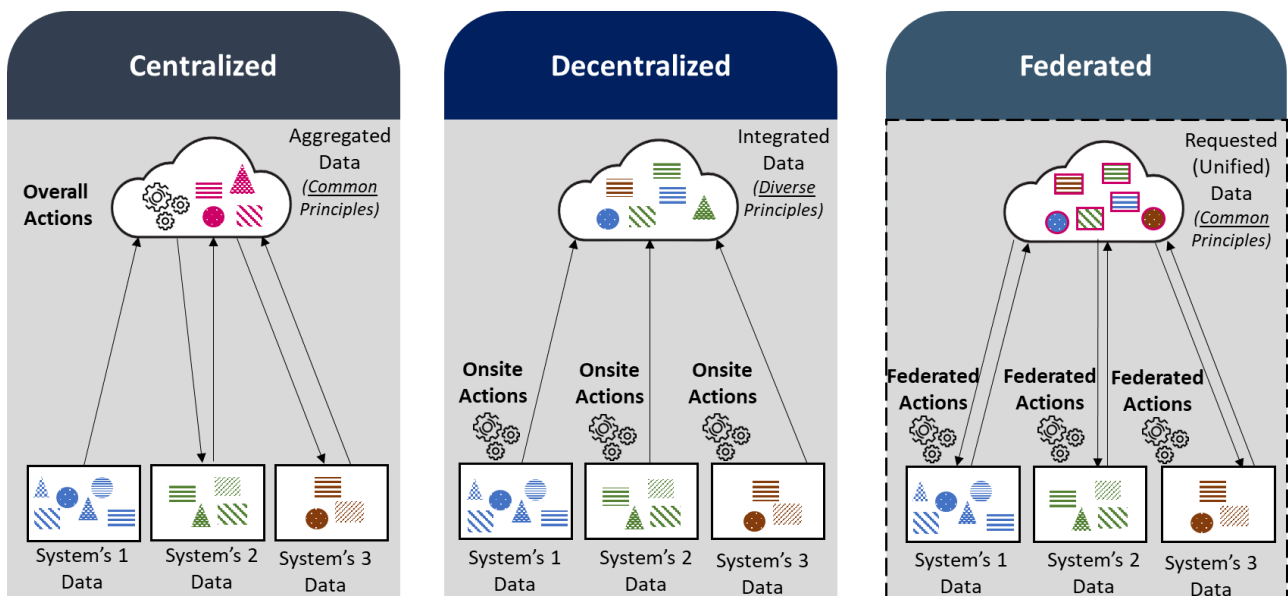


Figure 3 – Difference between centralized, decentralized, and federated data approaches

## 2.2 Challenges related to Health Data

The main challenge we are facing in the project trying to set up a framework that can train AI models on health data in a federated way is the practical **difficulty to collect sufficient data**.

The main reason for this lack of data is that health data is highly privacy sensitive data. Medical institutions are highly reluctant to share this data, even for research purposes, and even if anonymized. Medical organizations fear data leakages and data ending up in the wrong hands, and this fear makes them uncomfortable about the liability impact. In the current market conditions, there is no process available that can ensure that the data remains in good hands.

There is a common agreement that data should be anonymized and de-identified. However, even this process is very complicated and holds risks. As an example, the description of a rare disease in combination with the medical institute where the person having the disease is being treated could be sufficient to re-identify that person.

An extra risk factor is the fact that regulation on the processing on health data as well as on data for AI is still constantly evolving, and moreover vary with the jurisdiction that the data comes from.

Overall, organizations face a wide range of challenges when attempting to create and keep up a Data Space that satisfies the interests of all stakeholders. These challenges can be divided into two (2) primary categories: (i) intra-organizational (problems faced by data producers and consumers, as data sharing players), and (ii) inter-organizational (lack of adequate data sharing ecosystems).

The first major intra-organizational challenge is the difficulty to evaluate the value of data, due to the lack of data valuation and assessment tools. The arbitrary and party-dependent nature of data value as well as the general absence of producers' data sharing foresight make this issue even worse. The second issue is the challenge that data providers confront in balancing the perceived value of their data (after sharing) against the risks it exposes (upon sharing), even when they follow the rules.

In a business environment that is already fiercely competitive, there are multiple challenges that can be identified. One of them is the perceived loss of control over data, due to the fluid nature of data ownership (a concept that is still difficult or impossible to be defined legally). Other challenges are about the loss of trade secrets, due to accidental disclosure or malicious reverse engineering. The risk of evading legal constraints given potential data policy breaches (including General Data Protection Regulation (GDPR) and the disclosure of private identities but also policies related to new regulations such as the AI Act and the Data Governance Act are another category of challenges to be tackled.

Sharing data is merely to the benefit of the consumer of the data, while the risks lie with the data sharing organizations. As a result, from a data producer perspective, the legal and operational risks as well as the administrative burden to perform the process in a compliant way are so high that organizations are highly reluctant to share their data.

The top inter-organizational concern is the lack of reliable and meaningful data sharing ecosystems that compel quick widespread engagement. The main reasons are the absence of strong governance models, legal and ethical frameworks, and trustworthy intermediaries that ensure data quality, dependability, and fair usage. This is made worse by the fact that new best practices and standards (including interoperability, provenance, and quality assurance standards) are not being widely followed and whose maturity is likewise falling short of expectations. The rapid transition towards decentralized mixed-mode data sharing and processing architectures also creates considerable scaling issues. From a technical perspective, data sharing solutions need to satisfy European concerns like ethics-by-design for democratic AI.

All these challenges can be grouped into four (4) broad categories, referring to: (i) technical, (ii) business/organizational, (iii) legal compliance, and (iv) national/regional (as analyzed in the below sub-sections). It is possible to develop a viable Data Space that satisfies the demands of all the stakeholders and promotes the expansion of the European data ecosystem and economy by overcoming these obstacles and addressing each of these problems.

### 2.2.1 Technical Challenges

The need to create a cross-border, cross-sector sharing Data Space and give platforms the ability to handle “mixed” private, individual, and open public data creates new technical difficulties and exacerbates those that already exist. Following the emergence of opportunities for data sharing that extend beyond conventional raw data and its transformations along the processing chain to metadata, models, and processing algorithms, it is necessary to revisit the impact of known challenges (e.g., the Vs of Big Data: volume, velocity, variety, veracity) along the data lifecycle. The primary difficulties refer to:

- **Sharing-by-Design:** At the time of data generation, most data producers do not yet think that data sharing is a possibility. Existing data lifecycle management models need to accomplish a better job of incorporating all the pertinent processes, such as discovering the right data and preparing it for dissemination. Both the availability of the data itself and the maturity of the data services (such as cleaning and aggregation) in data sharing ecosystems are essential for the development of the data economy. Additionally, by separating the various types of data that can be shared into the categories mentioned above, the “variety” challenge becomes more complex, and interoperability solutions must take this change into account.
- **Digital Sovereignty:** A mixed data sharing space will only become a reality if data producers are assured to keep their ownership rights, allowing them to decide who can use their data, how it can be used, and under what conditions. To ensure digital sovereignty, additional research into appropriate data rights management frameworks or alternative ownership models is necessary.
- **Decentralization:** If data creators need to retain control over their data, decentralized data storage designs are favored over centralized data storage arrangements. If data storage is centralized, it also implies that control over the stored data comes under the control of the ‘data aggregator’. As a result, when discussing data volumes and data velocity (data streams), it is increasingly important to consider both the scalability of real-time operations over dispersed data at rest in arbitrary geographic distributions, and the distributed processing of data in motion, which does not require intermediate storage. There is also a rising need for standard data exchange protocols in decentralized architectures.
- **Veracity:** For data sharing ecosystems to continue to function, data integrity is still essential. Data at different processing stages will need to carry traceable information about their sources and processes (i.e., metadata about their initial state, algorithms, and processes they went through). To increase confidence, support for enhanced provenance is necessary. Solutions such as blockchain-based audit trails and SSI frameworks can enhance data veracity and trustworthiness.
- **Security:** Closed (proprietary, personal) data must be unlocked for interchange and sharing inside a trustworthy network, which necessitates a proper solution for problems like data confidentiality and digital rights management. Especially for health data, data confidentiality constraints are to be taken very seriously, even to the extent that the confidentiality requirements, at first sight, conflict with the need for cooperation. Furthermore, even in a decentralized peer-to-peer network, secure access control must be ensured. As a result, all the nodes and participants in the data sharing space must adopt standardized security solutions and exchange protocols. Meaning there is a need to provide in common infrastructure for both networking, computing and storage purposes.

- **Protection of Privacy:** Although there are technological solutions for secure and reliable data sharing (such as privacy-enhancing and privacy-preserving technologies, including digital identity management), these solutions must continue to mature to increase their acceptability and adoption.

### 2.2.2 Business & Organizational Challenge

The socio-economic viability of a pan-EU Industrial Data Platform (IDP) integrating various Data Spaces and providing Data Marketplaces is anticipated to provide the following business issues:

- **Values of EU:** IDPs created by the EU must uphold ideals like democracy, fair competition, and equality of treatment. These qualities can set businesses apart in the international market and get rid of dubious “shortcuts” that would benefit rivals from around the world. Additionally, new business models must show how they adhere to EU principles and how they are superior to current commercial solutions in this environment.
- **Overall Competition:** A major competitive advantage for the EU in the global market is the union of the digital and service industries. Therefore, it is necessary to find value-added data-driven services that could make “Made in EU” products competitive on a worldwide scale. Furthermore, Data Spaces are designed with a ‘co-opetition’ model, where companies collaborate in collecting data but then use the data in a competitive way. Co-opetition models require more research. Moreover, SMEs make up 99% of the EU industrial fabric, and the multiple contributing actors also complicate the definition of a model, as does the ‘Public-Private Partnership’ (PPP) structure like the Digital Innovation Hub (DIH).
- **Trust:** Understanding the commercial value of data produced by industry at all levels is essential for data markets. An issue is the lack of trust in the quality of data that is shared. Widespread, automatic data exchanges will not happen without quality requirements. Algorithms should also be subject to efforts to improve data accuracy. In this context, algorithm bias is a valid example. Additionally, costs associated with data preparation (such as cleaning and quality assurance) as well as risks (such as possible access to trade secrets and intellectual property sharing) must be considered. Additionally, careful adherence to GDPR requirements is required when sharing personal data in Business-to-Business (B2B) applications. Ad hoc and on-the-fly B2B data sharing mechanisms and contracts, given under clearly defined data sovereignty rules, must be taken into consideration to establish trusted data networks.
- **Standards for Valuation:** Data Marketplaces offer new possibilities and business models with the monetization or valuation of data assets at their core. The pricing of data poses new issues, such as determining whether this is done by the producer, by market demand, or by a broker or other third party. Another issue is determining whether the value of a particular data asset is fixed or dependent on the buyer-seller relationship. To help firms assess the value of involvement, guidelines and price models must be developed.

### 2.2.3 Legal Compliance Challenges

A complicated data policy environment has been created by all the various rules that have been implemented over the past ten years within the context of the digital single market. Therefore, a deeper knowledge of how data regulation interacts with and links to data platforms is required. Legal Compliance challenges are addressed in task T2.4 of this project and are described in the deliverables D2.3 (legal and ethical framework).

### 3. Business Process Workflow

To get a good understanding of what the Health Data Hub needs to offer, we will describe in this chapter different steps in a business workflow, starting from the onboarding of participants and ending with the usage of trained models in a federated setup.

#### 3.1 Onboarding process

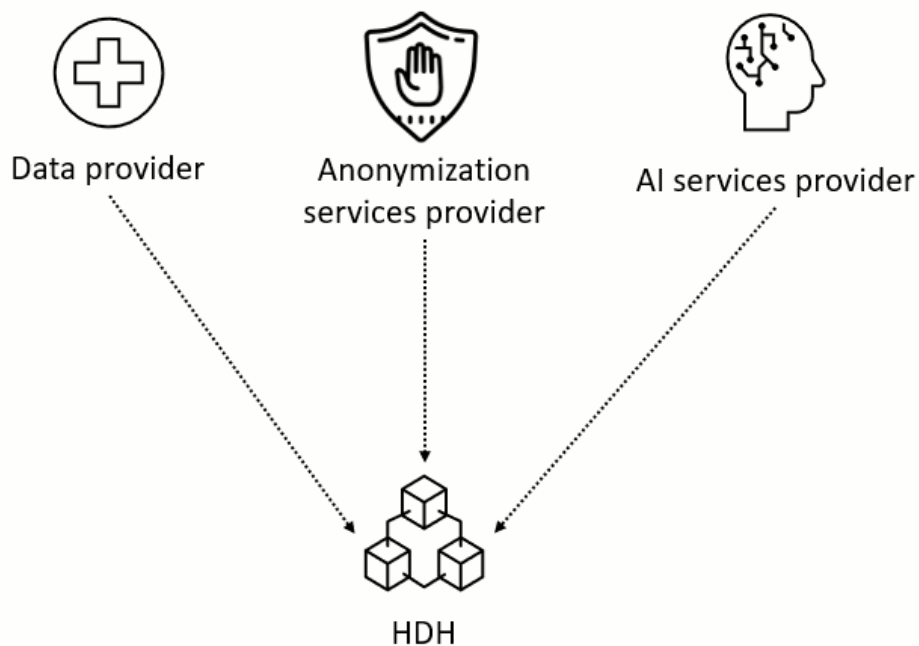


Figure 4 – HDH Onboarding Process

As a first step, we need to create our data space and define all the actors, including their role. It is of utmost importance to have this step done. Data may not fall in the wrong hands, and therefore we need a process that onboards the actors including the certification on the role they fulfil within the ecosystem.

Examples:

- Data providers need to be certified in their authority to give value to the data that is being published. If medical institutions are certified in their role, it gives a high degree of credibility to the published data as it shows that the offered data has been generated activity in the medical domain.
- Anonymization service providers are likely to be appointed as a ‘Neutral Data Intermediary’, as stipulated in the Data Governance Act, and for health data, it is likely that they will represent a national authority.
- AI Service Providers in the health domain, which is considered as a high-risk domain for AI, will be bound to regulatory constraints, and therefore will need to apply to be present in the space. Attaching a certificate proving what role to the credentials of this actor in the HDH system they have can highly facilitate in setting up the correct access management to data and services.

### 3.2 Register Offered Data and Services

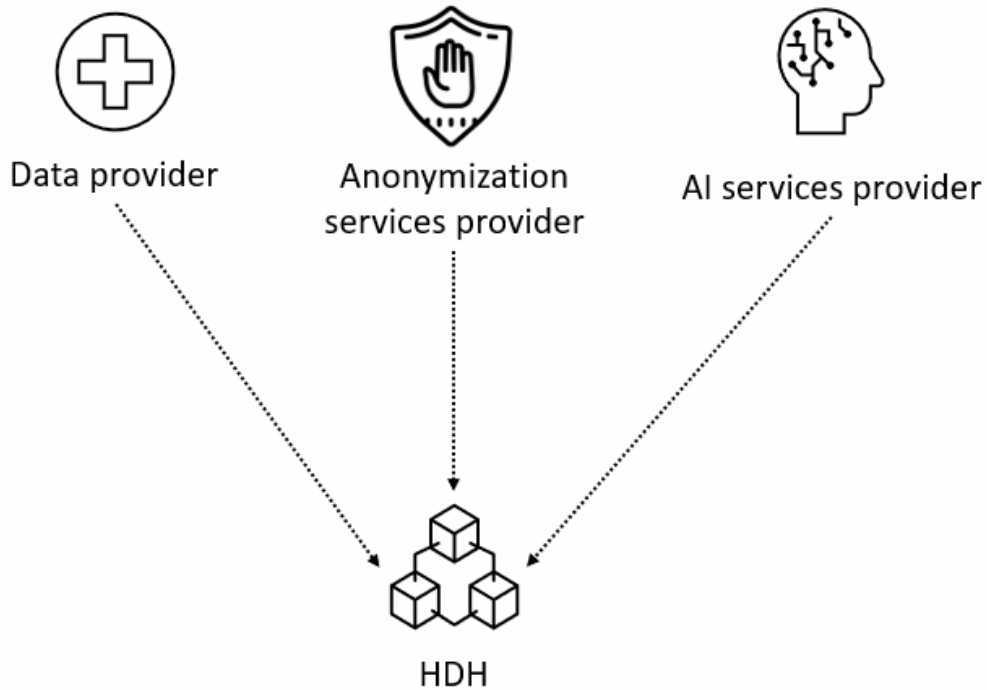


Figure 5 – HDH Registration of Data Sets and Services

Different types of assets can be registered in the Health Data Hub:

- Data sets around a certain topic, registered together with metadata in a catalogue by the data provider
- Services, like anonymization/de-identification services are published (ideally together with a certificate on the quality and content of the service, see chapter 4.6.5.4).
- AI models are published by AI service providers.

### 3.3 Deployment of Virtual Data Center

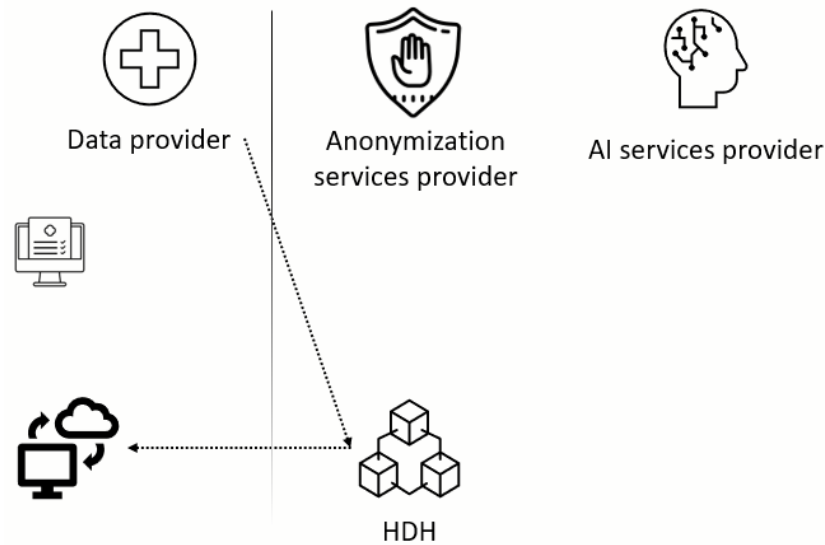


Figure 6 – HDH Deployment of Virtual Data Center

On request by a data consumer, a data provider needs to provide the infrastructure that provides in the preprocessing of the data for sharing, as well as providing the network connectivity to make the data available to the requested party.

We can translate this as the ‘deployment of a virtual data center’. With privacy constraints in mind, this infrastructure is ideally set up as self-managed, compliant computing infrastructure.

Through the HDH, the assigned resources are then enabled to provide in both secure computation, secure communication and/or data storage.

### 3.4 Deployment of Pre-processing Services handling sensitive data

Before sharing data, there might be a preprocessing service required, which anonymizes or de-identifies data or performs data harmonization on the data. These services are not necessarily provided by the data provider, but is likely a different actor. For the most performing result though, as well as to ensure privacy preserving, the service runs best in the virtual data center of the data provider.

The substeps for this preprocessing service could be as following:

- The data provider places a request for the using selected anonymization services from the HDH.
- Upon agreement, the anonymization services by certified provider get deployed on the Data provider’s own virtual data center.
- Privacy-sensitive data get processed while residing in realm and under control of the Data Provider.
- The anonymized data is made available to the Data Provider.
- Upon completion, the workload resources get released.

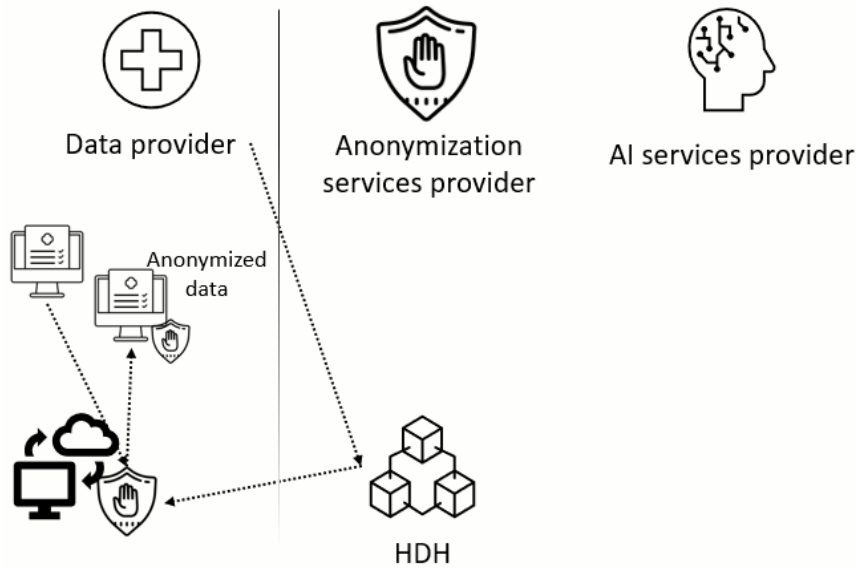


Figure 7 – HDH Deployment of Preprocessing Services

### 3.5 Anonymized Data Set Registration on HDH

The Data provider registers the anonymized data set to the HDH, meaning that it publishes it is available in its Virtual Data Center. The registration info includes but is not limited to info about:

- Metadata
- Data-set sample
- Anonymization guarantees (e.g. self-assessment, certificate provided by Anonymization provider, ...)

Once this registration step is done, the data set is discoverable via HDH marketplace

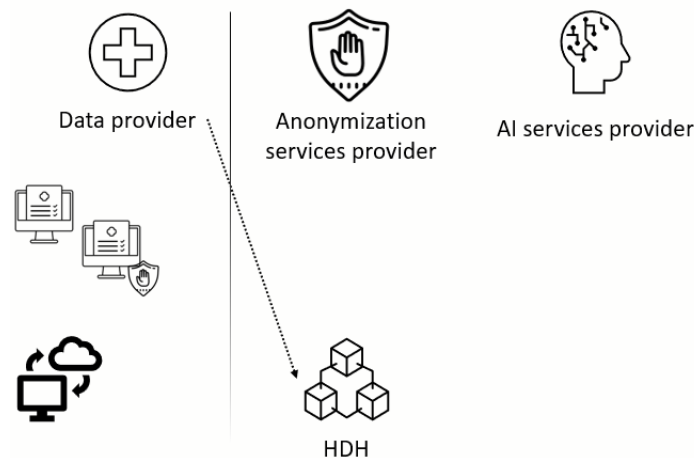


Figure 8 – HDH Anonymized Data Set Registration

### 3.6 Anonymized data set requested for modeling

In the next phase, a AI service provider wants to develop a model and searches for relevant data-sets in HDH marketplace.

The AI services provider can decide based on the available info and samples for matching data sets.

He requests to receive access to the de-identified data sets for ingestion into his model.

As a result, the anonymized dataset is accessible in a secure way for the scope of the AI training. The model training is executed on computing resources made available by the AI service provider.

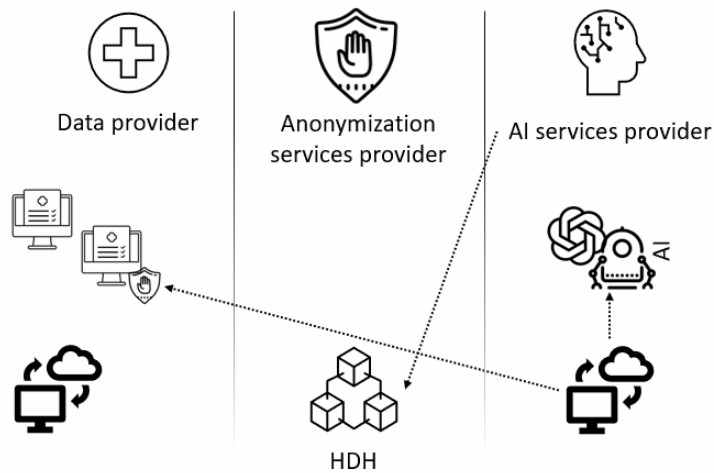


Figure 9 – HDH Data Set Request Process for modeling

### 3.7 AI model and services available for consumption

Once the model has been trained, the AI service provider publishes AI model and/or AI services in HDH marketplace.

Model and/or services are discoverable and consumption for interested parties can be orchestrated.

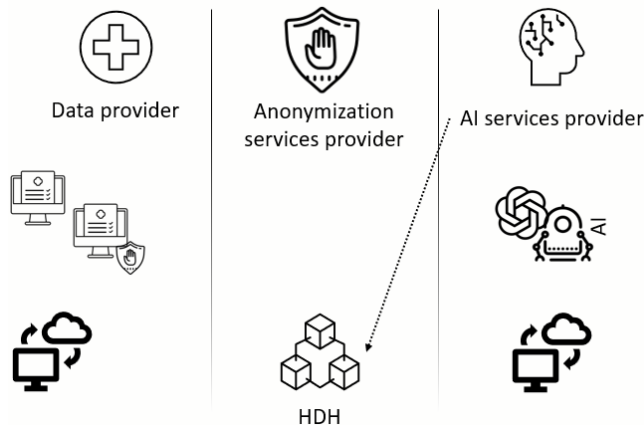


Figure 10 – HDH AI Model publication

## 4. PHASE IV AI Health Data Hub Design

This Section includes the design of the Phase IV Health Data Hub, aligned with the C4 Model of the Overall PHASE IV AI Architecture (D2.7 and D2.8).

The Phase IV Health Data Hub (HDH) Architecture follows the **C4 Model**, a widely adopted approach for structuring software architecture documentation. This model provides a hierarchical view of the system, allowing stakeholders to zoom in from a high-level system overview to detailed component interactions. The following sections break down the C4 Level 1 (System Context), Level 2 (Container), and Level 3 (Component) diagrams, offering increasing levels of detail while maintaining a structured and intuitive representation of the HDH ecosystem.

The C4 Model organizes architectural views into different levels of abstraction, helping software architects, developers, and business stakeholders understand how the system is structured and how its parts interact. The key concepts within this model are:

- **System:** A **high-level software platform** (such as the Health Data Hub) that provides services and interacts with users and external systems.
- **Container:** A **runnable unit of software**, such as a web application, a database, or an API service, which forms the building blocks of a system.
- **Component:** A **detailed internal module** within a container that carries out specific functions, such as handling authentication, executing data queries, or managing transactions.

This documentation follows the three main levels of the C4 Model, progressively zooming in to provide a deeper understanding of the system.

### 4.1 Context for Health Data Hub

On the image below you can find the representation of the system context diagram of PHASE IV AI. First, we will present all the actors involved. In C4, actors can be generic actors that interact with the system, roles or actual personas involved.

- **Data provider:** A data owner (e.g., hospitals, biobanks, national health authorities) who would like to share its data; he can also specify data anonymization requirements.
- **Data consumer:** A data consumer can be part of a research facility or from the industry sector. It is someone who, through a smart contract, wants to buy data.
- **Model producer:** As with data, also model can be published to the platform.
- **Model consumer:** As with the data consumer, the model consumer is part of a research facility or a company that wants to trade a model.
- **Data owner and infrastructure provider:** They are both data owners (e.g., hospitals, medical service providers) as well as facilities that can provide hardware for models' training when data location and privacy constraints need to be met (e.g., multi-party computation, federated machine learning).
- **Governance body members:** members who oversee managing the systems.

At the same time, we found three main systems involved in the system context diagram.

- The **Health Data Hub** system containing HDH subsystem and IAM subsystem
- The Data as a Service system
- The Model as a Service system

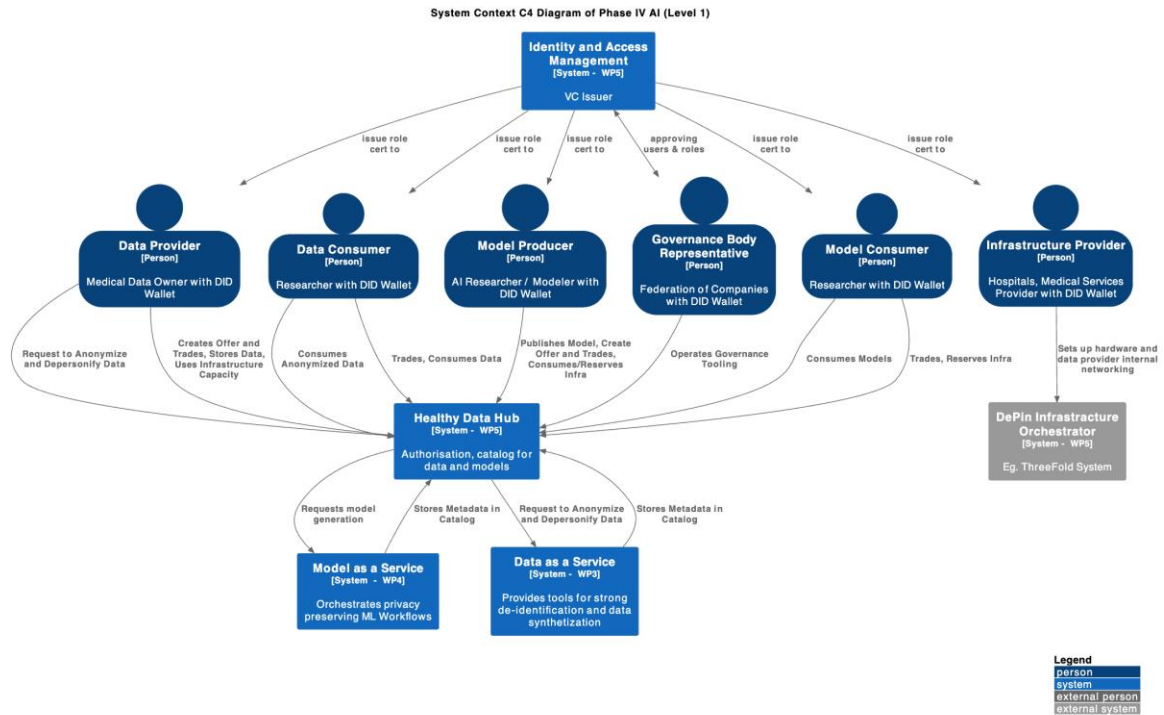


Figure 11 – System Context C4 Diagram of PHASE IV AI

The 5 system blocks can be described as follows:

1. **Health Data Hub (HDH):** A central platform that catalogs data and models, brokered by the Data Providers/Consumers and Model Producers/Consumers. It orchestrates key workflows—such as anonymizing data (via Data as a Service), training new models (via Model as a Service), and managing relevant trade or monetization actions.
2. **Data as a Service (DaaS):** Performs data anonymization and de-identification. It works closely with the HDH to ensure privacy requirements are met before making any dataset available for consumption or analysis.
3. **Model as a Service (Maas):** Handles the lifecycle of AI models—from training to publication. Model Producers register new models here, while Model Consumers discover and use them under agreed-upon terms, facilitated by the HDH.
4. **Identity and Access Management (IAM):** Issues verifiable credentials to actors, proving their roles (Data Provider, Consumer, etc.) within the ecosystem. Each system checks these credentials to enforce secure access.

Overall, this **context diagram** encapsulates the relationships and responsibilities among participants and systems, forming the robust foundation of data exchange, model sharing, and governance in PHASE IV AI.

An external system can be found also on the diagram, which is the DePIN Infrastructure Orchestrator subsystem. We consider it as external to our system as it is based on existing open-source technology, which will be used to deploy software on but won't be subject to modifications in this project.

## 4.2 Health Data Hub Container Diagram and Description

Below Container Diagram zooms into the Health Data Hub, breaking it down into major software components (containers) that work together to process and manage data. At this level, we:

- Define **core containers** within the HDH.
- Illustrate **how these containers communicate**, emphasizing APIs, database interactions, and integrations with external services.
- Explain **data flow** within the system, showing how requests are processed from users to backend services and external infrastructure.

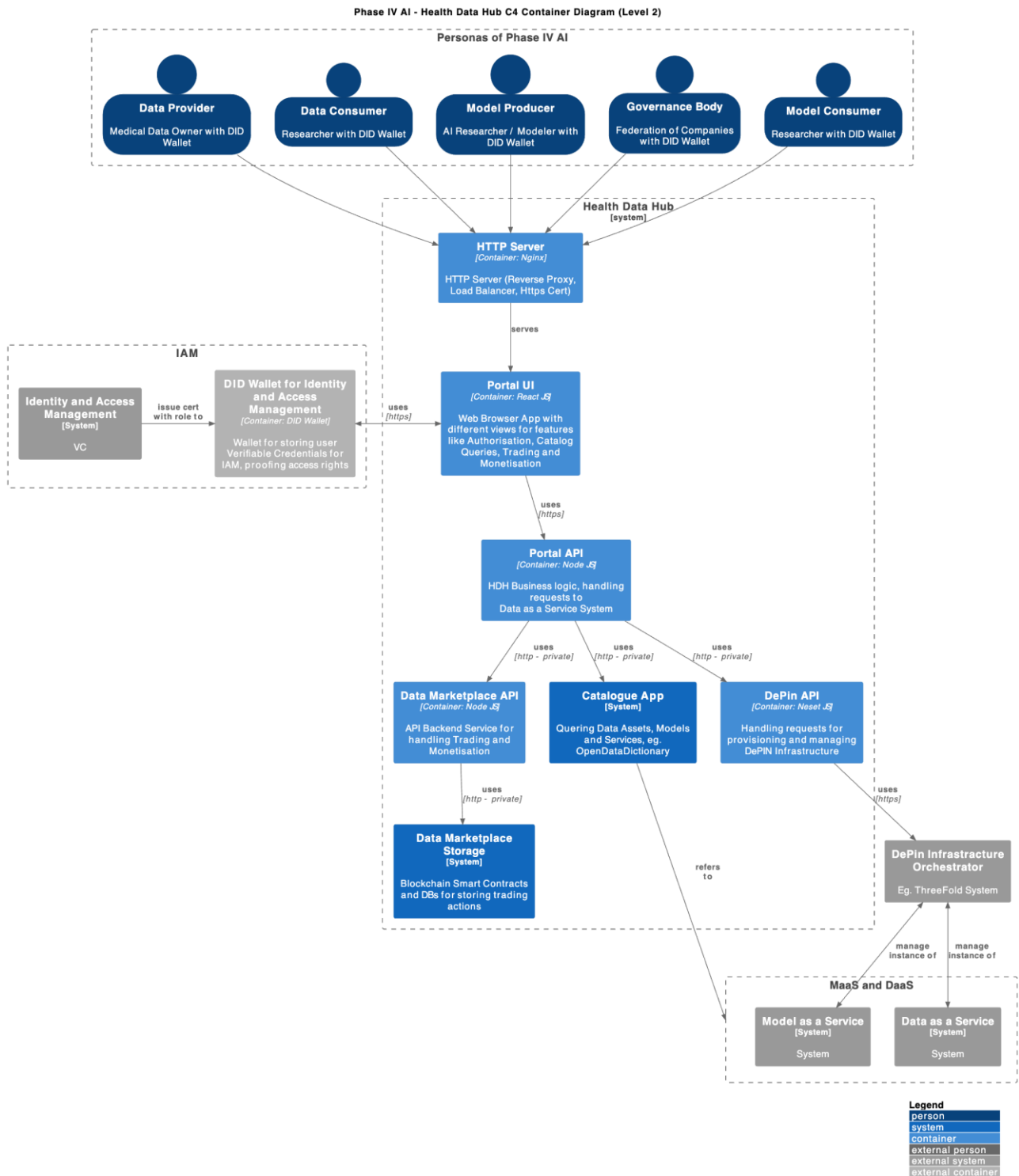


Figure 12 – Container C4 Diagram of PHASE IV AI – Part Health Data Hub

Above, the Level 2 Container Diagram provides an intermediate level of detail about how the Health Data Hub is structured internally. It identifies the major software components (containers), showing how they interact with each other as well as with external services, including Data as a Service (DaaS), Model as a Service (MaaS), and the DePIN Orchestrator.

The Health Data Hub remains the central aggregator coordinating requests between data or model providers, consumers, and various subsystems. Users continue to rely on the Hub to register, discover, and trade data assets or models, request anonymization services, or handle monetization. In addition, the Hub checks user credentials through an Identity and Access Management (IAM) service integrated with DID-based wallets.

Below is a concise overview of each **internal container** and its **responsibilities**, with mention of how the new elements fit in:

1. **HTTP Server (Nginx)**
  - Routes incoming traffic from all user roles—Data Provider, Data Consumer, etc.—to the appropriate back-end services.
  - Offers reverse proxy, load balancing, and TLS termination for secure communication.
2. **Portal UI (React JS)**
  - Provides a front-end where users can log in, browse catalogs, trade data/models, and perform monetization actions.
  - Makes secure HTTPS requests to back-end services like the Portal API and the Data Marketplace API.
3. **Portal API (Node.js)**
  - Manages core business logic related to data handling, anonymization requests (via DaaS), and interactions with the Catalog for data or model discovery.
  - Uses credentials from IAM to validate user requests.
  - Also calls the **DePin API** to manage infrastructure provisioning when a new environment for data or model processing is requested.
4. **Data Marketplace API (Node.js)**
  - Specializes in handling trading and monetization flows, including the creation of offers and transactions between data/model owners and consumers.
  - Persists transaction details in the **Data Marketplace Storage**.
5. **Data Marketplace Storage (Blockchain & DB)**
  - Uses a combination of on-chain smart contracts (for trust and immutability) and off-chain databases (for querying and history).
  - Stores trading records and references to data or model assets.
6. **Catalogue App**
  - Acts as the system-wide registry of discoverable data assets and AI models.
  - The Portal UI and Portal API communicate with the Catalogue to fetch or update information relevant to Health Data Hub users.
7. **DePin API (Nest JS)**
  - A new container responsible for provisioning and managing DePin (Decentralized Physical Infrastructure) resources on behalf of the Health Data Hub.
  - Receives requests from the Portal API to set up or modify infrastructure needed for data or model services.
  - Connects to the **DePin Orchestrator** to perform advanced infrastructure tasks, such as deploying new instances of DaaS or MaaS environments.

Beyond these internal containers, the Health Data Hub relies on external systems to ensure seamless operations:

- **IAM and DID Wallet:** The Hub checks user roles through verifiable credentials, using the Portal UI and Portal API to determine who can publish or consume datasets and models.
- **Data as a Service (DaaS) and Model as a Service (MaaS):** The Catalogue App references these systems for anonymization or model requests.
- **DePIN Orchestrator:** Invoked by the DePIN API to instantiate or update physical/virtual infrastructure used by DaaS or MaaS. This orchestration layer makes it easier for new facilities to spin up secure and compliant environments, integrating with the Health Data Hub’s trading and data-management ecosystem.

### 4.3 Identity Access Management System Container Diagram and Description

In addition to the Health Data Hub (HDH), we identified a need to design and develop Identity and Access Management (IAM) system, which ensures secure authentication and role-based access control within the PHASE IV AI ecosystem. The IAM Container Diagram (Level 2) provides a structured view of how verifiable credentials (VCs) are issued and used by participants.

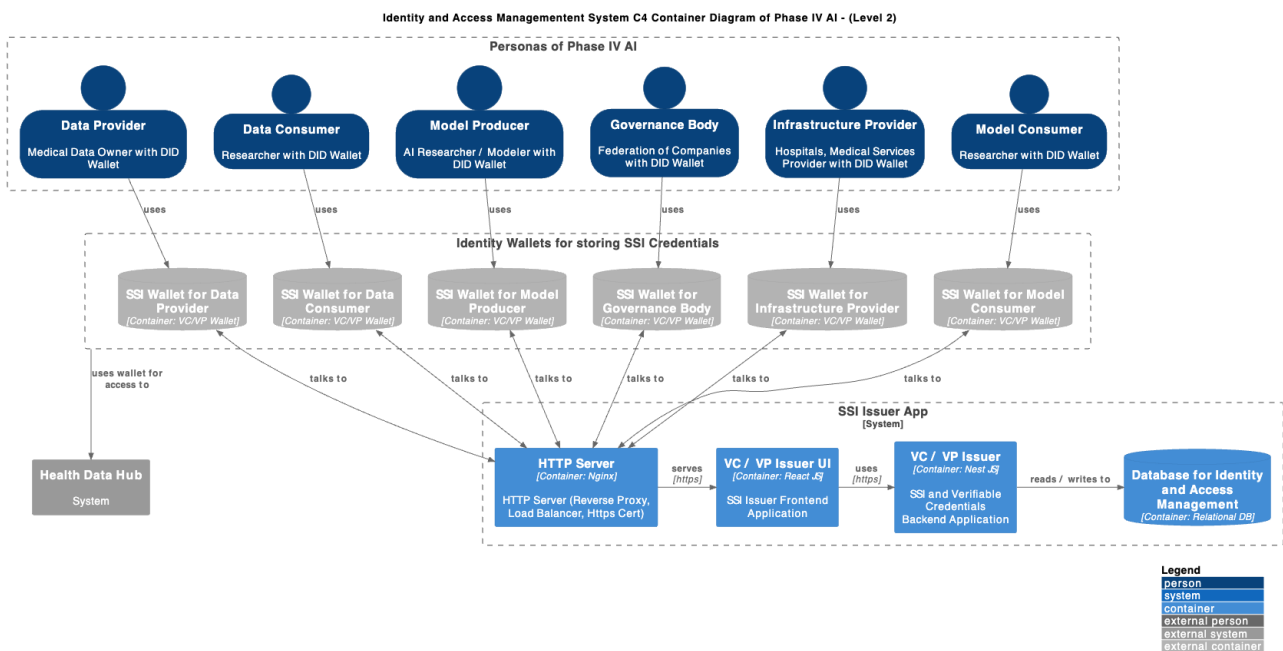


Figure 13 – Container C4 Diagram of Identity Access Management System of PHASE IV AI

#### Key Containers:

1. **HTTP Server (Nginx)**  
Provides a secure entry point for all wallet requests related to credential management. It routes traffic to the appropriate components, handles TLS/HTTPS, and balances incoming connections.

2. **VC / VP Issuer UI** (React JS)  
Offers a user-facing interface where participants can request or update their verifiable credentials. This interface also allows administrators or governance roles to review and approve credential issuance when necessary.
3. **VC / VP Issuer** (Nest JS)  
Implements the core logic for creating and validating self-sovereign identities. It receives requests from the Issuer UI, verifies user information, and generates or revokes verifiable credentials and presentations.
4. **Database** for Identity and Access Management (Relational DB)  
Stores credential metadata, user records, and any key information needed to maintain consistency and track issuance events. Although wallets hold the actual credentials, this internal database helps ensure system-wide coherence.

### **Interactions with External Systems**

User personas in PHASE IV AI—data providers, consumers, model producers, governance members, and infrastructure providers—each hold a dedicated SSI wallet outside the system boundary. These wallets communicate with the HTTP Server when owners need to request credentials or prove identities. Once valid credentials are issued, participants can present them to the Health Data Hub (HDH Subsystem) for role-based access and authentication. In this manner, the Identity and Access Management System anchors trust across the broader PHASE IV AI architecture, while leaving control of sensitive credentials in each user’s own wallet.

## **4.4 Data Consumer Workflow from Health Data Hub System Perspective**

This section details the step-by-step workflow for a Data Consumer interacting with the Health Data Hub (HDH) to search, request, and process datasets. The workflow is presented from the perspective of a researcher seeking specific medical data, utilizing the HDH platform and its integrated services.

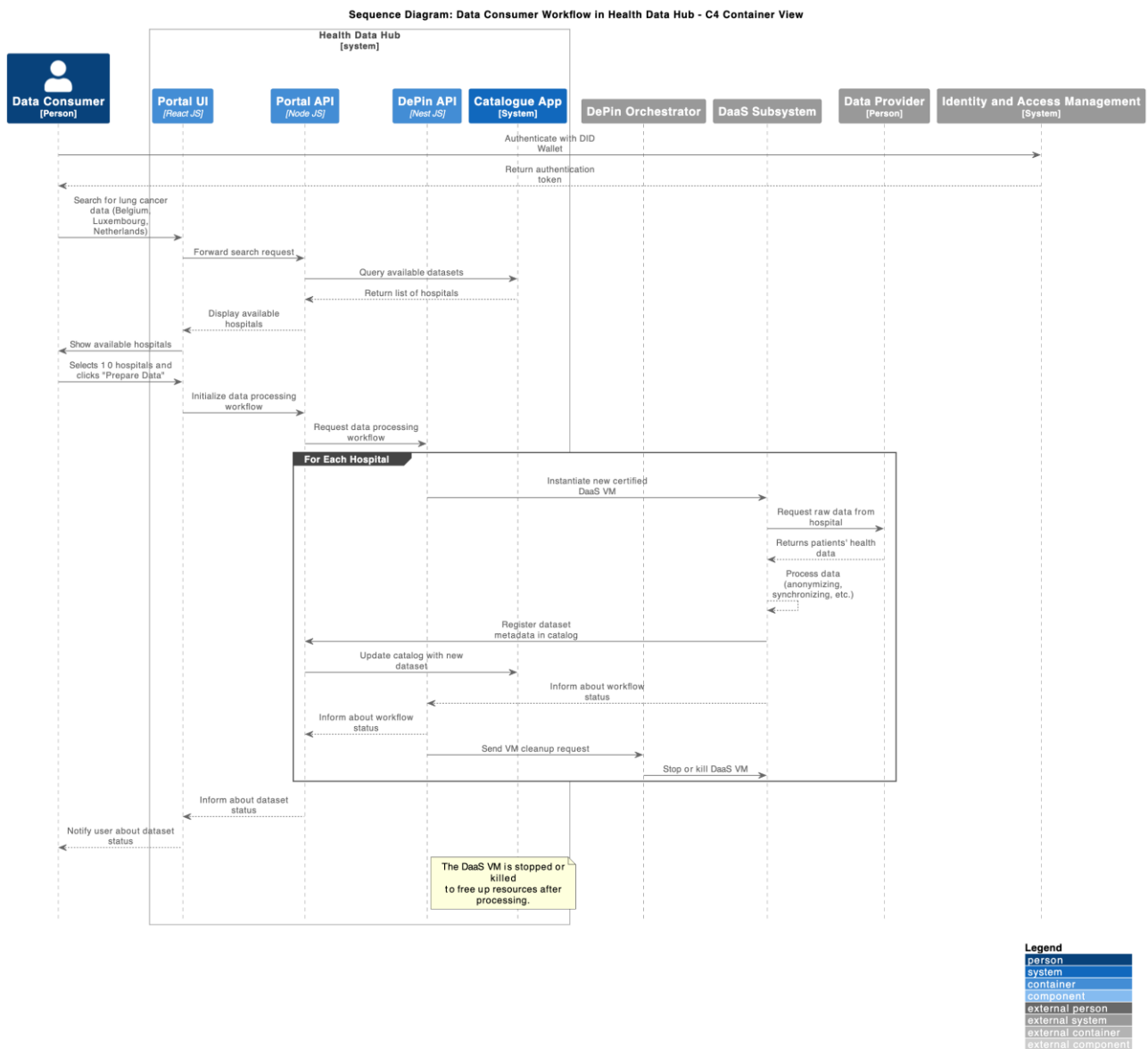


Figure 14 – Data Consumer Workflow in Health Data Hub (Container View Sequence Diagram)

A Data Consumer is typically a researcher from an academic institution or industry, seeking anonymized medical datasets for analytical purposes. The Health Data Hub provides a structured process through its interfaces (Portal UI, Portal API) and backend systems (Data as a Service, IAM, and DePin infrastructure) to facilitate secure and compliant data access.

The workflow is initiated when a Data Consumer searches for specific datasets through the HDH interface. Once a relevant dataset is found, the system ensures data privacy compliance by anonymizing and processing the raw data using virtualized Data as a Service (DaaS) instances. The processed dataset is then made available for the researcher in a structured format.

We can distinguish following key workflow steps:

### 1. Authentication and Authorization

- Before accessing the Health Data Hub (HDH), the IAM VC Issuer issues a Verifiable Credential (VC) to the Data Consumer (Researcher), proving their identity and role within the system.
- When interacting with the HDH Portal UI, the Data Consumer presents a Verifiable Presentation (VP) containing their role and key validation attributes.
- The HDH components validate the VP against the IAM system, ensuring the user has the required permissions to proceed with data search, selection, and processing.

### 2. Data Search and Discovery

- The authenticated user searches for specific datasets (e.g., "Lung cancer patient data from Belgium, Luxembourg, Netherlands").
- The search request is forwarded from the Portal UI to the Portal API.
- The Portal API queries the system's Catalogue App to fetch available datasets matching the request.
- The Catalogue App returns a list of hospitals that hold relevant datasets.
- The Portal API forwards this list to the Portal UI, displaying it to the Data Consumer.

### 3. Selection of Data Sources

- The Data Consumer selects multiple data providers (e.g., 10 hospitals) and requests data processing.
- The request is submitted via the Portal UI to the Portal API, which then initializes the data processing workflow.
- The Portal API interacts with the DePIN API to provision the required computational infrastructure for data processing.

### 4. Data Processing via DaaS

- For each selected hospital, the DePIN API instantiates a new virtualized Data as a Service (DaaS) instance.
- The DaaS instance requests raw health data from the selected hospital (Data Provider).
- The hospital provides patient health data, which is then processed within the DaaS instance:
  - Data is anonymized and depersonalized to meet compliance requirements.
  - Data is structured and synchronized according to predefined standards.
- The processed dataset metadata is registered in the Catalogue App.
- The DePIN API informs the Portal API about the status of the data processing workflow.

### 5. Cleanup and Completion

- Once processing is complete, the DePIN API sends a request to the DePIN Orchestrator to terminate the DaaS virtual machines, freeing up computational resources.
- The Portal API updates the Portal UI with the status of the dataset processing.
- The Portal UI notifies the Data Consumer that the requested dataset is now available.

The **Data Consumer Workflow** in the Health Data Hub ensures a secure and efficient process for researchers to access anonymized health data while maintaining strict compliance with privacy regulations. By leveraging **DID Wallet-based authentication**, the system guarantees that only authorized users can interact with sensitive datasets. The integration of decentralized infrastructure (DePIN) **and** privacy-preserving data

processing (DaaS) enables hospitals to share health data while ensuring anonymity and security. **The** automated provisioning and decommissioning of DaaS VMs optimizes resource usage, preventing unnecessary computational overhead. The structured workflow not only **enhances** data discoverability and traceability but also fosters a trust-based ecosystem, where research institutions and healthcare providers can confidently collaborate while safeguarding patient privacy.

## 4.5 Technology Choice and rationale

When choosing the right network infrastructure of this project, the following requirements and constraints need to be observed:

- Health Data has the highest requirements in terms of both data privacy and end-to-end security due to the high-risk categorisation of these data in GDPR. Moreover, data privacy for health data is, in general, a delicate matter, as sometimes personal information can be revealed unintentionally. This accounts especially for persons have rare diseases, for which a combination of the disease and the medical institution can be sufficient to point to and identify a named person. This has a substantial impact on the way that the platform we intend to build needs to be architected. Ideal would be to find a way that anonymizes the raw data as well as ingests it as training data for a model, without the possibility to extract and exploit the data during the ingestion process.
- We intend to build a data space, i.e. a federated data infrastructure allowing many participants to exchange their data in a secure and privacy-preserving way, with the participants retaining sovereignty on the data they contribute. Meaning that there is no place for a 'central' data aggregator.
- The data can only be made accessible to participants after a robust screening on the participant's right to view the information. This means that these participants need to be identified and certified in their role before grants can be made to access the data. This problem is common to many data spaces and has been addressed using new identity and access mechanisms, based on self-sovereign identity (SSI) principles, creating DIDs and certifying through Verifiable Credentials (VCs). This is also the approach that we propose in the design of the Health Data Hub.
- Other WPs of PHASE IV AI explore techniques and solutions to meet the security requirements which are advanced, such as multi-party computation (MPC) and full homomorphic encryption (FHE). These solutions rely however heavily on very specific foundations, which need to be rolled out in a common way to all participants, and puts some requirements at a very low level, close to the hardware. Multi-party computation has an impact on how CPU and GPU devices can/must be interconnected, FHE puts some requirements on a common encryption protocol when transferring data between locations and during the computation process. The way to control this is by interacting with a system which has control over these hardware primitives, both for compute, storage and network. The IT system that meets this goal is an Operating System (OS).
- PHASE IV AI has as an objective to also try to incorporate monetization into its model, so that owners of data assets can make a financial benefit from sharing the data with others. Again, in a data space context, we intend not to go through a financial aggregator (such as a bank), which is quite cumbersome to use in a federated setup. Moreover, blockchain technology is an alternative that has proven over the past few years as a way for exchanging value, and looks, from a technological perspective, promising as an enabler in the orchestration of data spaces.

We have screened the market and have come to a solution which has the promise to deliver on all of the above requirements, and which is known as “DePIN”. A more detailed view on the screening can be found in Annex A.

DePIN, or Decentralized Physical Infrastructure Network, is a concept that combines decentralized networks with physical infrastructure. It leverages blockchain technology and decentralized systems to manage, coordinate, and potentially own physical assets or infrastructure in a distributed manner.

The key-aspects of DePIN are listed in the following break-down:

1. **Decentralization:** Instead of relying on a central authority or organization to manage physical infrastructure, DePIN aims to distribute control and ownership among multiple stakeholders. This can increase transparency, reduce costs, and promote fairness.
2. **Physical Infrastructure:** This can include a variety of assets such as telecommunications equipment, energy grids, transportation networks, or even real estate. DePIN seeks to apply decentralized principles to the management and operation of these assets. In the case of this project, the main interest goes to the capabilities of physical or virtual ICT compute, storage and networking capabilities.
3. **Blockchain Technology:** Blockchain and related technologies provide the backbone for decentralized systems. They offer secure, transparent, and immutable records of transactions and ownership, which is crucial for managing and coordinating decentralized physical infrastructure.
4. **Tokenization:** In many DePIN models, physical assets are tokenized, meaning they are represented as digital tokens on a blockchain. These tokens can be traded, managed, and used to represent ownership stakes or access rights.
5. **Incentive Structures:** DePIN often incorporates incentive mechanisms to encourage participation and investment from a wide range of individuals or organizations. These can include rewards for contributing resources, maintaining infrastructure, or participating in governance.

Overall, DePIN represents an innovative approach to managing physical assets and infrastructure by combining the principles of decentralization with the tangible world. It aims to create more resilient, efficient, and equitable systems by leveraging the strengths of decentralized technologies.

One notable project in the DePIN space is Threefold. This is the technology that has the promise to satisfy all of the above requirements, through the combination of Zero-OS, a federated, secure-by-design version of Linux, and blockchain technology that is used for the orchestration of workloads among participants in an ecosystem, in a secure and privacy-enabling way.

## 4.6 Key Components Description

The Health Data Hub as a container holds multiple components to fulfil its activities.

### 4.6.1 DePIN Hardware Infrastructure

In this chapter we introduce main components of Threefold DePIN solution that are key enablers for the overall system.

#### 4.6.1.1 Decentralized Operating System

We need an approach to orchestrate in a structured way how hardware components work together, as an example between compute hardware (for multi-party computation), or in the networking (secure mesh network, allowing the secure interaction between multiple participants in a data space, without the need to go through a centralised 'hub').

These interactions are typically done by an Operating System. Traditional Operating systems, however, don't have the capability that allows federation of IT workload management over different participants, while in the setup we have chosen for PHASE IV AI, we explore federated learning to tackle the privacy concerns around a classic AI model training on health data. Federated learning is 'a' form of a federated workload.

An Operating System that does have these capabilities is Zero-OS. It is an important technology created by an open-source project called Threefold.

It has a Linux kernel inside, meaning that it can run any IT workload that runs under Linux. Around the kernel, the project has implemented features that enables federation of workloads in a secure way:

- Zero-OS is a stateless and lightweight operating system which only runs in memory, reducing the hacking surface.
- Zero-OS has no shell, and there is a complete abstraction made between the infrastructure providing side and the infra consuming side. Hosted workloads are protected from administrative exploits and human intervention.
- The different hardware nodes (called 3Nodes) contribute to compute, storage and network capabilities and are interconnected in a mesh network, and each have their own IPv6 address.
- Zero-OS runs autonomously on 3Nodes once booted, requiring no maintenance or administration.
- All nodes have their own network connector and their own IPv6 address, allowing them to be taken up in any private network when requested, in a fully secured way.
- Orchestration and registration of all the nodes and their hosting metadata (owner + public key, location, address information (IPv6), hardware capabilities) happens on TFChain blockchain infrastructure. This infrastructure also provides in smart contracts, which register and monitor the contracting on usage of the nodes.

The project must also provide the tooling to bootstrap and connect the hardware to the private network in an easy and scalable way.

#### 4.6.1.2 Mesh Network

As all the 3Nodes have the same stacks to interact without a problem, it is quite easy to connect them in a mesh network. Also, from a user/application perspective, workloads can be set up over different interconnected nodes, in a secure way.

Secure interconnection between nodes can be defined in different architectures. The classic setup is 'hub-and-spoke', with clients connected to servers with the intermediation of a 'VPN Hub', as shown in the below scheme.

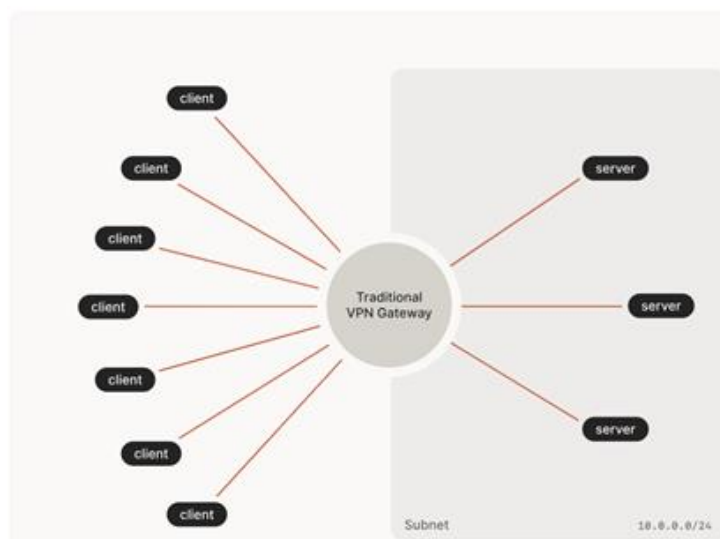
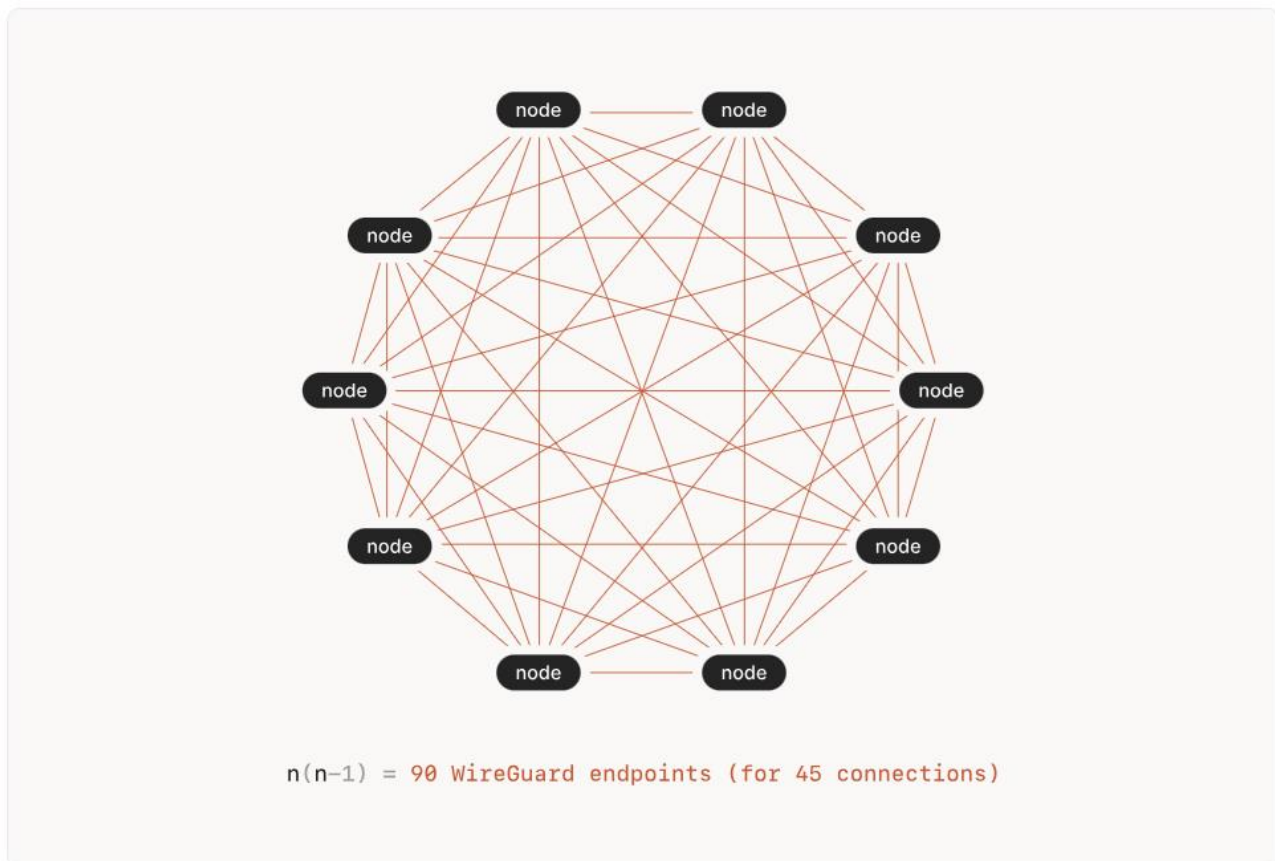


Figure 15 – Difference between centralized, decentralized, and federated data approaches

In a federated setup, however, there is not a single place that can be designated as a hub. There are multiple medical institutions and multiple datacentres in multiple regions and countries. In traditional VPN setups, companies configure a single VPN concentrator, and then set up secondary tunnels (often using IPsec) between locations. So remote users arrive at the VPN concentrator in one place, then have their traffic forwarded to its destination in another place.

This traditional setup can be hard to scale. First, remote users might or might not be close to the VPN concentrator; if they're far away, they then incur high latency connecting to it. Secondly, the datacenter they want to reach might not be close to the VPN concentrator either; if it's far away, they incur high latency again.

Recently, lightweight VPN tunnel technology has emerged, which is light enough to enable the creation of a multi-hub setup with good performance. Every node in the network can connect to any other node. This is what we call a 'mesh network', as illustrated by the scheme below.



*Figure 16 – Topology of a mesh network*

This allows to design elegant peer-to-peer apps.

A few issues need to be addressed however:

- A 10-node network would require  $10 \times 9 = 90$  tunnel endpoint configurations; every node would need to know its own key plus 9 more, and each node would have to be updated every time you rotate a key or add/remove a user.
- The nodes would all have to find each other somehow and reconnect whenever one of them moves around.
- Opening a firewall port for every single node to allow new incoming connections is complicated.
- Compliance requirements might oblige you to block and audit traffic between all the nodes, even though it's no longer all going through a central location that you can clamp down.

This is where mesh networks come in, with all nodes interconnected, as each can obtain its own address. Best is to use private IPv6 addresses in a private network, which are abundantly available, over IPv4, and only use public IP addresses when connecting to the world outside of the interconnected nodes.

There are several private overlay network technologies that we can implement on the infrastructure consumption side to interconnect users in a secure way. We have explored Mycelium, Yggdrasil, Tailscale, Netmaker, Netbird, Twingate, Zrok and OpenZiti; Each of these has its own features, advantages and disadvantages, as explained in Annex B.

For choosing the right technology for the Mesh Network, we will limit ourselves to 2 technologies (for the sake of time saving) :

- Tailscale, as it is the most widespread mesh network technology
- Mycelium given the full integration with Zero-OS available, making it easier for non-technical end users to set up a network.

#### 4.6.1.3 Quantum-secure storage

Zero-OS allows storage to be set up using the 'quantum-secure storage (QSS)' mechanism [97]. This is a decentralized, globally distributed data storage system, presented as 'unbreakable, self-healing, append-only and immutable'.

The storage architecture follows a true peer-to-peer design. Any participating node only stores small incomplete parts of objects (files, photos, movies, databases...) by offering a slice of the present (local) storage devices. Managing the storage and retrieval of all of these distributed fragments is done by a software that creates development or end-user interfaces for this storage algorithm, called '**dispersed storage**'.

Peer-to-peer provides the unique proposition of selecting storage providers that match the criteria required on the application or service of business, e.g. for regulatory reasons. For example, you might be looking to store data for your application in a certain geographic area, for compliance reasons. Also, you might want to use different "storage policies" for different types of data. Examples are live versus archived data. All these use cases are possible with this storage architecture and could be built by using the same building blocks produced by infrastructure providers and consumed by developers or end-users.

Given that also Full Homomorphic Encryption capabilities will be extensively explored in WP4, quantum-secure storage will be explored only if dispersed storage is required.

#### 4.6.2 Data markets

The PHASE IV AI will explore the implementation of a market for 2 types of data assets:

- Market for Data sets - a collection of data and metadata in the medical domain such as Health Data Records, medical images, both personal, synthetic, anonymized, de-identified.
- Market for Data Services - programs or applications that process the data into a new form such as Data Anonymisation and De-Identification services (that pre-process raw data; Data Harmonisation services; Trained Model services)

The market for **data sets** will be explored in more detail for the next version of this document, D5.4.

The **Data Services** Market is a module which is set up in a decentralized way: the component consists of multiple smart contracts that each run on blockchain infrastructure.

Initially, ThreeFold's TFChain and the available smart contracts on the Parity Substrate will be used to fulfil the market feature.

We estimate that the best implementation based on this open-source component would be to fork this implementation and create a permissioned set-up based on it. This would require a substantial additional budget and effort and is therefore considered as out of scope for the PHASE IV AI research project. Moreover, there is a body required for governing the technology stack. We expect this to happen with the realisation of EHDS, but there is info missing at the time of writing on how EHDS will be shaped.

Based on our recent research we haven't encountered impediments in the DePIN to use it for a data services marketplace.

### 4.6.3 Catalog for Data Assets and Services

WP5 also intends to provide the tooling to provide a catalog with the available data assets and data services. For this purpose, a Web graphical user interface will be made available.

In the context of PHASE IV AI, selecting an optimal data catalog solution is critical for supporting privacy-compliant data sharing, effective data governance, and advanced AI-driven innovation. This document evaluates leading data catalog platforms through a detailed feature-by-feature analysis, highlighting their strengths, limitations, and alignment with PHASE IV AI's objectives. The evaluation emphasizes scalability, extensibility, privacy compliance, and AI/ML capabilities to ensure that the chosen solution meets the project's ambitious goals.

We decided to use and adapt Open-Source Catalog Software for PHASE IV AI and not build it by ourselves. There are following reasons for that choice:

1. **Leveraging Existing Mature Solutions:** Open-source data catalogs are well-recognized and mature systems, reducing the need to build complex systems from scratch. This saves significant time and resources, which is critical for a research-driven initiative like PHASE IV AI.
2. **Avoiding Reinventing the Wheel:** Developing a robust data catalog from scratch is resource intensive. By adopting an existing open-source platform, PHASE IV AI can focus on innovation rather than infrastructure development.
3. **Comprehensive Documentation:** Open-source platforms are often accompanied by detailed documentation, enabling contributors to integrate and customize the solution from the project's inception. This accelerates the onboarding of collaborators and the start of technical development.
4. **Transparency and Trust:** Open-source platforms provide full access to source code, ensuring stakeholders can verify functionality, security, and compliance with privacy regulations, which is crucial in handling sensitive healthcare data.
5. **Cost Efficiency and Customizability:** Without licensing fees, open-source solutions allow PHASE IV AI to allocate resources effectively while enabling platform customization to meet specific project requirements.
6. **Alignment with Academic and Research Principles:** Open-source's collaborative ethos mirrors the project's values, supporting partnerships among universities, research institutions, and private entities to drive innovation and maintain academic integrity.

A number of Data Catalog toolkits have been assessed and evaluated.

Two toolsets were selected for exploration during the project:

- **OpenDataDiscovery:**
  - Lightweight and community-driven with a strong focus on extensibility and open standards (ODD API Specification).

- Highly suited for ML-first use cases and metadata federation.
- Lower dependencies make it easier to deploy and manage.
- There is a good fit with the Use Case, as it appears to be excellent for projects requiring high customization and ML-centric workflows with minimal resource overhead.
- **OBiBa**: open-source software solution based on Opal and Mica projects for epidemiological data management, harmonization and dissemination, OBiBa software consists of a suite of stand-alone applications that support various study's data management activities. These modular applications can be integrated to create a comprehensive information management and analysis system for individual studies.

To date, OBiBa hasn't been analysed and evaluated in depth. However, we intend to deploy an instance of the tool given consortium partners have made first-hand contributions to the tool. This makes it easier to customize it to the needs of the project, more than other tools which require a learning phase. OBiBa will be further explored and analysed during the second phase of the WP5 design phase (M13-M31).

#### 4.6.3.1 Advantages of the OpenDataDiscovery Catalog Platform

After evaluating multiple open-source data catalog solutions, **OpenDataDiscovery** emerged as a strong candidate due to its **architectural simplicity, automation-first approach, deployment efficiency, cost-effectiveness, and extensibility**. Below, we outline the key benefits that make this platform particularly well-suited for PHASE IV AI's needs.

One of its standout advantages is its architecture, which allows for **flexible deployment and scalability**. By leveraging Elasticsearch, OpenDataDiscovery provides **powerful search capabilities**, enabling users to efficiently query large datasets with minimal latency. Unlike more complex platforms such as OpenMetadata or DataHub, OpenDataDiscovery is designed with **lightweight components**, making it **easier to maintain and operate** while reducing technical overhead.

A major strength of the platform lies in its **automation-driven metadata management**. It places a strong emphasis on **automated metadata collection**, which significantly reduces the burden of manual data curation. Built-in **query analysis capabilities** further enhance its usability by generating **automatic insights** from structured and unstructured metadata, ensuring data is always **up-to-date and easily discoverable**.

From a deployment perspective, OpenDataDiscovery is **simpler to set up** and manage compared to other open-source solutions. Its lower system requirements and streamlined deployment model result in a faster time-to-value, making it particularly appealing for projects that require **rapid implementation with minimal configuration overhead**. This ease of deployment makes it an ideal choice for research teams and organizations with limited DevOps resources.

In terms of **cost efficiency**, OpenDataDiscovery's lightweight architecture translates into lower infrastructure costs, as it does not require heavy computational resources to operate. The **open-source nature of the platform eliminates licensing fees**, and its reduced maintenance complexity allows teams to focus on data discovery rather than managing an intricate system. Compared to large-scale enterprise solutions, OpenDataDiscovery provides an **optimal balance between capability and affordability**.

Another crucial factor is its **extensibility**, which allows organizations to customize the platform to their specific needs. The modular design supports the development of **custom collectors**, ensuring seamless integration with **existing data pipelines and tools**. This flexibility makes OpenDataDiscovery adaptable to various organizational use cases, whether in academic research, healthcare, or AI-driven analytics.

Given these strengths, OpenDataDiscovery is particularly well-suited for **small to medium-sized organizations** that require an automated, lightweight, and low-maintenance metadata cataloging solution. It is an excellent choice for teams that prioritize simplicity over excessive complexity, especially when focusing

on data warehouses, databases, and AI-driven workflows. By reducing operational barriers and enhancing **data governance through automation**, OpenDataDiscovery aligns perfectly with PHASE IV AI's goal of building a scalable, privacy-compliant, and research-oriented data infrastructure.

#### **4.6.3.2 Main Trade-offs of OpenDataDiscovery Catalog Platform**

While OpenDataDiscovery offers significant advantages in simplicity, automation, and cost efficiency, it comes with certain trade-offs compared to more feature-rich enterprise solutions like OpenMetadata. The most notable limitation is the lack of extensive enterprise-grade functionalities, such as advanced role-based access control (RBAC), deep lineage tracking, or highly complex data governance workflows. However, this trade-off can actually be an advantage for organizations that prioritize ease of deployment, low maintenance overhead, and streamlined metadata management over heavyweight enterprise features.

For use cases where rapid deployment, automation-driven metadata collection, and lightweight architecture are preferred, OpenDataDiscovery stands out as an optimal choice. It eliminates the complexity that often comes with large-scale cataloguing platforms, making it ideal for teams that need an efficient, scalable, and low-cost solution without the burden of manual metadata curation. By focusing on core metadata collection and automation, OpenDataDiscovery enables organizations to maximize value with minimal resource investment, aligning well with the objectives of PHASE IV AI.

#### **4.6.4 Services deployment technique using flist**

We intend to make the deployment of IT services as lightweight and easy to use for non-technical people as possible. Deploy an installed workload is the easiest way to do this. This can be accomplished by the so-called flist.

An flist is a script to describe a software workload very effectively, so that it can be deployed fast and with high reliability.

In a flist, we separate the metadata from the data, where the metadata is a description of the files in that image, providing information about the app/software. Thanks to flist, the 3Node doesn't need to install a complete software program to run properly. Only the necessary files are installed. Zero-OS can read the metadata of a container and only download and execute the necessary binaries and applications to run the workload when it is necessary.

The Flist technology provided by ThreeFold can be built on the basis of existing standards: it is possible to convert any Docker image into an flist, using the easy-to-use [ThreeFold Docker Hub Converter tool](#).

Examples can be found on the Threefold flist hub (<https://hub.grid.tf>), see [tf-images](#).

The details of the flist creation steps can be found in Annex C.

#### **4.6.5 Identity and Access Management**

##### **4.6.5.1 Self-Sovereign Identity for Role-Based Access Control**

To meet non-functional requirements such as end-to-end security and privacy-by-design, the process needs a very strong gatekeeper.

Classic Role-Based Access Control, managed by one centralized governance body, isn't fit for purpose, however. In a federated ecosystem such as a data space, it is imperative to avoid as much as possible one centralized operator which is hierarchically situated above the other personas. A federated set-up has, by design, all personas interacting having the same level of influence. Data Spaces have investigated how to solve this issue and have identified Self-Sovereign Identity (SSI) as a means to develop a method for managing access control in a federated / decentralized way.

The goal of SSI is to have an identity framework that applies 10 principles:

- **Existence:** The user must have an independent existence. This principle states that a user **must be able to exist in the digital world**, without the need for a third party.
- **Control:** Users must stay in control over their own identities. Sovereignty allows the user to control how his/her identity is used without negatively disrupting the way society is organised.
- **Access:** Users must have access to their own data. Users must be able to access their data and any associated **claims without the interference of gatekeepers or intermediaries**.
- **Transparency:** Algorithms and infrastructures must be transparent. In tandem with the previous principle, transparency ensures that users can monitor any potential mismanagement of claims, credentials or associations related to their identity. In the broader context of identity, transparency also integrates fairness and support for a **balanced identity system**.
- **Persistence:** Identities should live in the long term. This principle argues that identities should be long-lived, at the discretion of the user. Amid constant changes in data storage and private key rotation (if a user has multiple wallets or identifiers within a blockchain), persistence allows users to maintain their identities, despite having multiple private keys. Persistence is **not only exclusive** to individuals; other institutions, organisations and collective entities should be subject to having their identities at the discretion of other entities. In the end, identifiers in an SSI system should be the exclusive property of the person(s) who created them.
- **Portability:** Identity-related information and services should be easily portable. Information and services must be easily portable and cannot be held exclusively by a centralised third-party entity. Even if a third-party entity works in the best interest of the user, **the problem of SPOF** (single point of failure) remains. Portability ensures that one's identity can be transferred and stored in multiple locations, at the user's discretion.
- **Interoperability:** Identities can be used as widely as possible. Interoperability overlaps with persistence and portability. The importance of portability in identity is that identity information and services must be portable. This ties in with interoperability, particularly as **portable identities** are more readily available to cross international borders.
- **Consent:** The user must agree to use their identity. SSI systems require that personal data are shared only with the **user's consent**. When building a decentralised self-sovereign identity, consent must be continuously kept in mind and incorporated into the system. This ensures whether the identity data will remain private (at the user's discretion).
- **Selective disclosure:** The disclosure of one's own data must be minimised. This principle emphasises the importance of protecting users' personal data when disclosing identity-related information. For example, if a user's minimum age is required (to access a page), a user should not be required to provide the exact day, month and year of their birth. Instead, user disclosure should be minimised by providing the minimum age requirement. By implementing **selective disclosure**, range testing and other zero-knowledge techniques, developers can facilitate minimisation to better support privacy. Basically, active minimisation allows for more privacy-preserving interactions between users and systems.
- **Protection:** Users' rights must be respected. To ensure user protection, there must be an independent  **censorship-resistant**  algorithm that can authenticate user identities.

W3C has developed a full framework, evolving into a standard on the implementation of SSI, with Decentralised Identifiers (DIDs), Verifiable Credentials (VCs) and Verifiable Presentations (VPs) as core concepts to realise this.

Health Data is categorized as high-risk data when using it for AI purposes. Also, in GDPR it is given a highly personal sensitivity, meaning that data itself should be protected. This also implies a check on who gets access to the data.

In a federated organisation, such as a data space, identity management is considered as a cornerstone to make sure that only eligible profiles get access to the data.

Groundbreaking work on Identity management for data spaces has been made in the Gaia-X framework. We can say that a standard is being set there, and we will explore ways to set up an Identity and Access Management System that is founded on the same principles.

A well-developed framework founded on the W3C standards and applicable to Data Spaces is the Gaia-X Trust Framework. See <https://docs.gaia-x.eu/policy-rules-committee/trust-framework/22.10/>

The Gaia-X Trust Framework is the set of rules that define the minimum baseline to be part of the **Gaia-X Ecosystem**, which forms the solid basics for data spaces across Europe. Those rules provide a common governance and the basic level of interoperability across individual ecosystems while letting the users in full control of their choices.

The Gaia-X Trust Framework operationalizes the requirements as defined by Gaia-X, e.g. in the Policy Rules and Labelling Document, or the Architecture Document - where the latter especially allows for federated and interoperable Gaia-X ecosystems.

The Trust Framework foresees verifiable credentials and linked data representations as cornerstone of its future operations. Trusted information shall be retrieved in machine readable manners, and where such manners are missing, Gaia-X will define processes to translate trusted information in a machine-readable format. This is a prerequisite of federating trusted statements within the Gaia-X Ecosystem and developing mechanisms to re-assess validity of claims within the Gaia-X Trust Framework.

In addition to Gaia-X, the European Blockchain Services Infrastructure (EBSI) Verifiable Credentials (VC) Framework is key to enabling secure and trusted digital interactions in data spaces. EBSI leverages blockchain technology to issue, verify, and present tamper-proof credentials, ensuring privacy and trust. By adhering to W3C standards, it supports decentralized identity management, making it ideal for handling sensitive data while ensuring compliance with regulations like GDPR. For more details, see <https://ec.europa.eu/digital-building-blocks/sites/display/EBSI/EBSI+Verifiable+Credentials>.

Finally, also the Simpl project (<https://simpl-programme.ec.europa.eu/>) is building a framework that helps to easily set up data spaces, and this includes a framework, derived from the Gaia-X Trust Framework, to create certificates based on Verifiable Credentials. The outcome looks very promising, though the presentation of this Simpl framework was only done at the moment that this delivery (D5.3) is submitted. We intend to closely follow up on its evolution and readiness for usage by other projects.

#### 4.6.5.2 Certification of IoT devices

Next to identity verification of people and organisations, we will touch upon the potential need for PHASE IV AI to manage access control for systems and devices, e.g. to fetch data and train an AI model with it. It can be beneficial for data to be directly generated from scanners and sensors, but in order to do this, the authenticity of these devices need to be explored. Also there, we need to make sure that the accessing system can deliver the right credentials before getting access to the data. We will investigate a way to allow access control to happen to (IoT) devices, at the edge.

DID methods have been developed and there is a start of an attempt to define the best DID method for the best use case. The [Identity Foundation](#) [99] is trying to match the best DID method to each

use case, however for the case of PHASE IV AI there is no clear answer yet, and current DID methods might fall short into this.

We feel that existing DID methods lack the capability to make a clear link between the service requested or given and the certificate that is used as an entry point to get or give that service. The main issue is that the service provider or user is not addressable in a digital way.

The exception to this is `did:web` [100], which contains a url in the DID description, and, through this, allows to easily address the DID holder. The method has its limitations, however: all needs to be translated through a DNS service; individual users need to be registered with the company linked to this domain to get a personalized DID; this method is best fit for an organisation and its users, but there is no easy way to clearly identify (IoT) devices using this method.

In this project we want to explore the possibility for using a DID method which can make the unique mapping between persons and devices on the one hand, and their identity on the other hand.

This high-level investigation will be further elaborated in the upcoming period (M13-M24) and the results will be included in deliverable D5.4.

#### 4.6.5.3 Usage of SSI in the data asset reservation process

A process that will then use these Verifiable Credentials in the overall data and services reservation will be as follows.

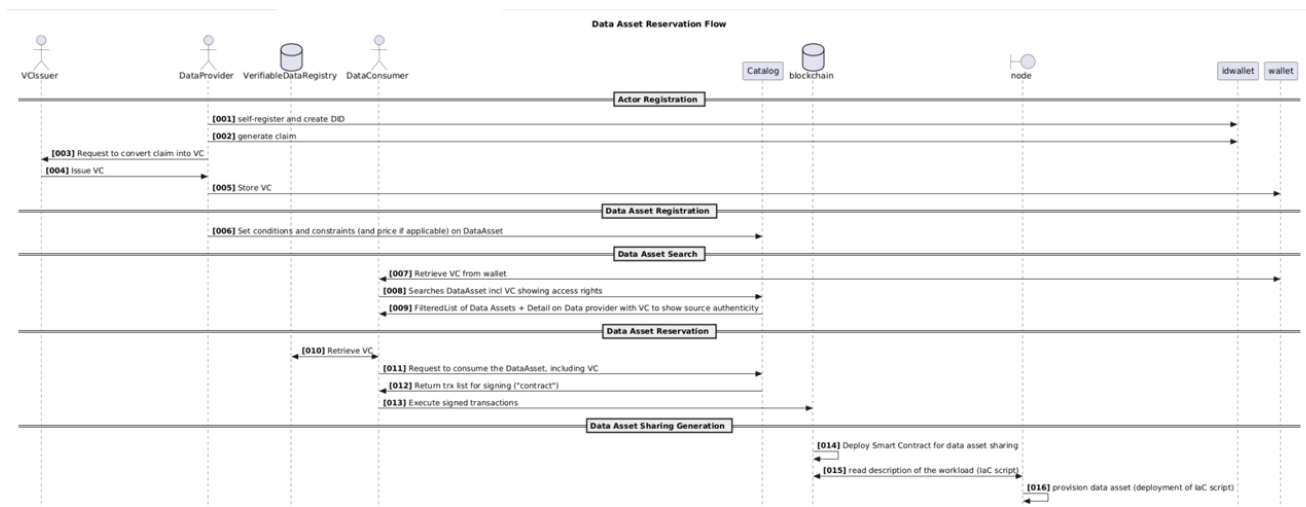


Figure 17 – Data and Services Reservation Process Flow Diagram

The basic principle is that all actors as well as devices can present themselves with the identity certification proving that they are who they claim they are (important foundation to achieve end-to-end security), and that they have the authority to get access to data and information (important privacy constraints in the health space). Moreover, we don't want to have one central authority issuing these certificates, nor a central authority that monitors all accesses.

The self-sovereign identity concept brings a solution to tackle these constraints:

- Each participant is uniquely identified through a public/private key pair. The public key is known, the private key is only known to its holder, as a secret, and can be used to sign any digital document, creating through this hashes that can only be made using the private key.
- Certificates can be issued by a certified party (VCIssuer in the diagram), which has the authority to certify some claim of an actor (ex. on being a medical institution, having the consent from a patient to use his data, etc.). These authorities issue credentials but are not active participants in the overall process.
- The certificates can come along with the access request, making it a token that can be exploited for access control.

#### 4.6.5.4 Usage of SSI in software certification

The AI Model training process on health data is bound to very stringent regulatory constraints:

- The ‘Data Governance Act’ stipulates the data requires an intermediation process by neutral organisations, in the case of health data appointed by national authorities, which process the data to perform de-identification and anonymization.
- The ‘AI Act’ stipulates that modelling on high-risk data such as health data requires a human intervention to avoid trained models that have a bias and generate critical risks. In the case of health data, an abusive interpretation of data could lead to serious health issues and even death, and a substantial liability issue arises through this.

The constraints expressed in both regulations imply measures that, if not implemented using a thoughtfully designed architecture, lead to scaling and even practical feasibility issues. In the case of the Data Governance Act, the first premise is that data sets are sent to trusted data intermediaries, are de-identified or anonymized by these organisations. The idea of imposing the intermediation of each individual data set by a neutral authority creates a heavy bottleneck on the preparation of data sets to be used for AI training, which is a very heavy data intensive ingestion process. High data consumption volumes are essential for the generation of highly qualitative AI training. The installation of a process that acts on each individual data set leads to substantial scaling issues.

The constraints put by the AI Act are about feasibility and auditability of the training process. The existing IT artifacts that describe a workload, such as code, Docker containers or Kubernetes setups cannot be complemented with a certification process to indicate that the training has indeed happened with a human validator. Without this certificate, the enforceability of the AI Act is questionable.

We will explore the possibility to use DePIN infrastructure and tooling such as blockchain and Flists, combined with certification through verifiable credentials to address the issues raised by the constraints in both Acts.

In this context, the following process is proposed:

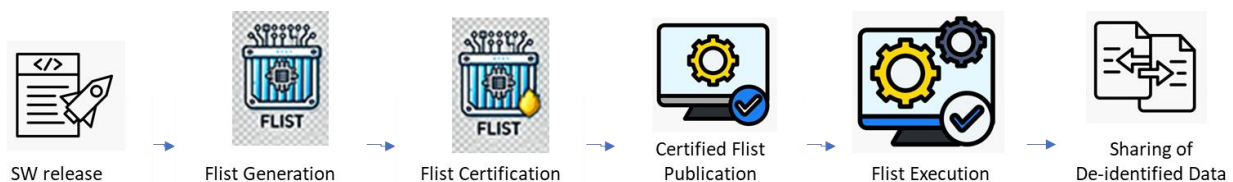


Figure 18 – Sequence diagram of Flist usage

**Step 1a:** Software is published as an installation binary, such as a Docker Image or an executable.

**Step 1b:** The goal of the software is described as a ‘claim’, which can be of various nature, such as for example

- To de-identify or anonymise health data
- To generate an AI model on health data

**Step 2:** The software is prepared for deployment, including parameters which are specific to a specific workload execution such as context-specific parameters and rule sets, access keys into a ‘Flist’ (deployment binary).

**Step 3:** The Flist has a unique identifier (e.g. by creating a Flist content hash) and is certified by a ‘trusted data intermediary’.

This is an essential step in the process, as it satisfies the compliance goal with regulations (ex. EU Data Governance Act, AI Act).

A neutral data intermediary certifies the software, which is described as a claim holding an flist, and uses its personal private key to convert the claim as an ‘issuer’ into a Verifiable Credential.

**Step 4:** The Flist is published in a service catalog, along with the certificate. The Flist is deployable on premise / under full control of the one executing it.

**Step 5:** The software runs on data provider or model provider premise and a trace of it is stored on blockchain infrastructure. This registration of the Flist execution creates an auditable record of the process. The auditable record is important, both for de-identification software - as it proves that the resulting data set has gone through a certified software – as for model training, as the certifying authority can do in-depth controls before certifying, and the registration on blockchain infrastructure creates an auditable proof that a certified AI model has been used.

**Step 6:** The resulting de-identified data is good for sharing as it has gone through a compliant/certified de-identification process.

#### 4.6.6 Interconnected Hardware Infrastructure reserved for PHASE IV AI

The best way to create a process that is end-to end secure and privacy-enabling by design for all the activities, is by having an orchestrator that operates at the lowest level, very close to the hardware, able to orchestrate the hardware to be used and securing all contributing hardware to run the process. Orchestration of hardware is done by an Operating System. In this section we will describe high-level how we see this federated hardware orchestrated. The available primitives in the DePIN technology chosen are an operating system, a low-level data storage protocol, a mesh network protocol, a blockchain component with smart contract capabilities and wallets (payment and identity) bound to and identifying the participants in the ecosystem through their public/private key pair.

##### 4.6.6.1 Primitives

###### 4.6.6.1.1 Zero-OS operating system

Zero-OS is an operating system that serves as the foundational layer for a **federated data infrastructure**, eliminating interoperability issues between different independent actors.

The Operating system has a Linux kernel, making it fit for any Linux workload. Around the Linux kernel, all primitives have been built from scratch to make it a fully federated grid of interoperating nodes, with all features available to enable end-to-end security and privacy by design.

Zero-OS is the foundational OS on the capacity layer. It has been designed bottom up, starting from a Linux kernel and secure boot BIOS. It combines 3 primitive functions: **storage capacity**, **compute capacity**, and

**intelligent network functions for running the network services.** The **distributed storage layer** allows storage of exabytes of information at a low cost while maintaining a high privacy and reliability level. Hardware capacity in any size and of multiple nature (CPU, GPU, memory, HDD, SSD, IPv4 and IPv6 addresses) hosted on secure bootable devices (any Intel/AMD HW) can be added to the grid of interconnected hardware nodes.

A very important consequence of this is that Zero-OS can run at the edge, on any location that has an internet connection, so it is the perfect installation for a medical infrastructure, remaining in full control of the hospital without imposing the burden to these institutions to have extensive in-depth technical knowledge. All measures around networking, security and privacy have been taken up into the architecture of the solution, by design (the audit and certification on this claim is ongoing).

Some more detailed specifications of Zero\_OS can be found in annex E.

#### 4.6.6.1.2 Quantum-Secure File Storage

The data storage can be dispersed over different Zero-OS nodes in a way that stored data on an individual node is meaningless. The same mechanisms are used as in Data Centre RAID systems, with the difference that storage is dispersed here over different capacity owners, which results in the truth being no longer held in one location.

In annex A, paragraph 6.2.2.1.1, dispersed storage is explained in more detail, with an example.

#### 4.6.6.1.3 Secure Overlay Network

Networking is set up through a secure meshed overlay IPv6-network (using the new internet standards), encrypting all traffic between workloads (containers, VMs, etc.) running on this network. Also there, capacity owners have no view on the content of the traffic generated on their infrastructure, herewith making a complete abstraction between the infra-providing and infra-consuming side.

The network has a high security protection against malicious external intruders as both internal and external connections can only be established from the inside. Inside the private network, nodes talk to each other using sockets through Remote Direct Memory Access (RDMA), and the link with the outside has very high protection against intrusion using a web gateway, which has been implemented at a level which is very low, close to the hardware.

**The network component is an Overlay network** which lives on top of the existing internet or other peer2peer networks created.

Inside the private network, **everyone is connected** to everyone. Also, there is **End-to-end encryption** between users of an app and the app running behind the network wall.

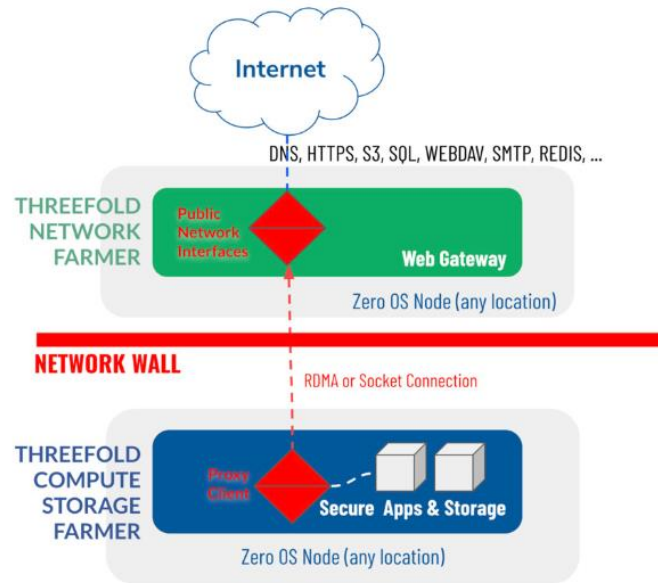


Figure 19 – Network Wall in a Secure Overlay Network

Each user end network point is **strongly authenticated** and uniquely identified, independent of network carrier used.

Benefits are the following:

- Communication between peers on a network happens over the shortest possible path.
- The end-to-end encryption of all exchanged data over the established network guarantees all data to be fully secured.
- The network topology allows for peer2peer links like meshed wireless.
- The system is resilient: it can survive broken internet links and re-route when needed.
- Given all peers have IPv6 addresses and exchange data using these addresses, there is no more need for IPV4 addresses, hence the issues and cost related to the IPv4 address shortage are herewith tackled.

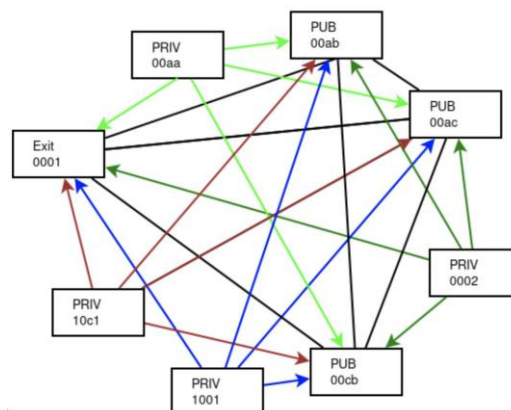


Figure 20 – Mesh network topology

The network topology allows redundancy as:

- Any app can get (securely) connected to the internet by any chosen IP address made available by ThreeFold network farmers through Web Gateway (WebGW).
- An app can be connected to multiple Web Gateways at once, the DNS round-robin principle will provide load balancing and redundancy.

- An easy clustering mechanism exists where Web Gateways and nodes can be lost, and the public service will still be up and running.
- Network is self-healing. When containers are moved or re-created the same end user connection can be reused as that connection is terminated on the Web Gateway. The moved or newly created Web Gateway will recreate the socket to the Web Gateway and receive inbound traffic.

Zero-OS directly interacts with the Network Interface Controller (ZNIC), which can be implemented as an interface to both the mesh network (private network set up among consortium partners) as to a public IP address on Zero-OS.

#### 4.6.6.1.4 Blockchain Component and Smart Contract for IT

For the procurement of infrastructure, reservation of it for workloads and the financial handling, a blockchain component and smart contract are in place.

Deployment of IT workload using a so-called “**Smart Contract for IT**” makes the deployment process of IT workload resilient to human error and hacking. The system is **self-driving** and **self-healing**, therefore removing the human requirement for deploying and operating IT infrastructure and services.

All transactions are recorded on blockchain infrastructure, ensuring a personalized immutable record of any workload reservation, infrastructure registration or update to it.

This mechanism allows for self-healing, i.e. the restoration of a workload if needed, due to failing hardware, dropout in electricity etc.

There are no people involved to run and keep the IT workloads operational, as IT architectures are configured and installed by bots. By default, there is even no human intervention and access possible.

This is a very different way of thinking – it leads to much more security, efficiency, and higher performance, and will lead to big network savings.

The 5-step process is explained as a key component in annex A, paragraph 6.2.2.1.2.

#### 4.6.6.1.5 Wallet

Zero-OS can only be accessed by a PKI enabled Virtual System Administrator (VSA), making the owner of the keys in full control of the workload he is running. Also, this is an important privacy feature: a VSA basically is an enabler for a user to be in control of his workload, and the starting point for any decentralised application (like a decentralized social media app, decentralized video, chat, ...) where info is only shared with the people you know, and no data centralisation happening. Linked to the PKI infrastructure can be activated a ssh key, proper to the user who set up the workload.

The wallet connected to the Threefold infra is Threefold Connect. For the time being, we propose to also use this wallet, as it contains TFT, the payment token used for reserving infrastructure and workloads.

Threefold Connect has some more key features and benefits available:

- 2FA Authenticator:
  - Funds are protected by the unique seed phrase on the phone, combined with a password.
- Decentralized:
  - The app is self-managed and decentralized, allowing users to access the ThreeFold Network platforms and their fully private digital wallet.
- Private Digital Wallet:
  - Users can manage their ThreeFold Tokens (TFT) and view their transaction history on the TF chain.

#### 4.6.6.2 Hardware Infrastructure Provisioning

The hardware infrastructure provisioning is a pretty straightforward, plug-and-play process, which is described in annex D of this document.

#### 4.6.7 User Wallet

A user wallet enables a user to securely manage blockchain-based assets.

In this project, 2 types of wallets are of importance:

- Payment wallets hold the digital assets that can be used for payment and provide services to launch transactions that are registered on blockchain infrastructure. For this project, we limit ourselves to use TFT (= Threefold Tokens) as the payment instrument to be used. Replacing TFT by other payment instruments would lead us too far in this project, even though we consider it as a recommended step to adopt stablecoins, digital EUR, ... It would lead us too far though to proceed with this kind of instrument and this will be a topic to be taken up by a governance body operating the Health Data Hub.
- Identity wallets hold the identity and verifiable credentials about a certain claim, expressed digitally. It will describe the identity of the entities that are participating in the data space, with a cryptographic certification of their role, as a 'Verifiable Credential'. Identity wallets are accepted as a good way to build trust into a data space, and valuable work has been performed in the establishment of the Gaia-X Trust Framework.

#### 4.6.8 UI Portals

All portals are unique pieces of technology, set up in a way that they can be controlled by their holder through their public/private key pair.

This key pair resides in the user's wallet for physical persons, for edge devices, we can opt to have the key pair, the DID as well as the VCs certifying the claims about this identity, in the device's storage infrastructure. They need to be available whenever the holder of the identity needs to authenticate himself.

Each actor will have a certificate/VC that is specific to the role that he is taking up in the ecosystem, giving them access to a portal that is specific to a role taken up by the actor in the data space: Infrastructure Provider Portal; Data Provider Portal; Data Consumer Portal; Model Producer Portal; Model Consumer Portal.

#### 4.6.9 Infrastructure as Code (IaC) Providing Deployment Services

WP5 will provide in ways to deploy in a dynamic way any infrastructure that is needed to store data, consume data, provide compute resources to train models, and destroy the infrastructure again once training of the models has been achieved (for economic reasons).

The most flexible way to address all these requirements is 'Infrastructure as Code'. We will explore several mechanisms that allow to deploy and destroy infrastructure easily, such as Terraform and Pulumi.

**Infrastructure as code (IaC)** is the process of managing and provisioning computer data centre resources through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. The IT infrastructure managed by this process comprises both physical equipment, such as bare-metal servers, as well as virtual machines, and associated configuration resources. The definitions may be in a version control system, rather than maintaining the code through manual processes. The code in the

definition files may use either scripts or declarative definitions, but IaC more often employs declarative approaches.

During M13-M24 we will define recommendations about the best IaC tooling for PHASE IV AI. A shortlist of candidate IaC tooling can be found below.

**Terraform** is an open-source tool that enables you to describe and deploy infrastructure using a declarative configuration language. With Terraform, you can define your infrastructure components, such as virtual machines, networks, and storage, in a human-readable configuration file. This file, often referred to as the Terraform script, becomes a blueprint for your entire infrastructure.

The Terraform tool can automate the provisioning and management of infrastructure across various infrastructure providers, ensuring that your deployments are reproducible and scalable. It promotes collaboration, version control, and the ability to treat your infrastructure as code, providing a unified and seamless approach to managing complex environments.

With **Pulumi**, you can express your infrastructure requirements using many languages, creating a seamless bridge between development and operations.

Another way to deploy infrastructure is using a **Typescript tool**, implemented on the Threefold dashboard. A web interface has been created to use this kind of IaC tool.

## 4.7 Suggested Approach during the Project

The ideal solution for the project would be to establish a proprietary blockchain infrastructure, as this would allow us to operate in a fully private environment tailored to the unique needs of PHASE IV AI. However, after careful consideration, we propose leveraging the Threefold stack as the foundational blockchain technology for this project.

The Threefold stack offers several key advantages that align with the goals and constraints of PHASE IV AI. Firstly, it is a comprehensive and coherent technological solution that is continuously evolving. Developing and maintaining a dedicated blockchain infrastructure exclusively for PHASE IV AI would impose an overwhelming burden on the project team, particularly given the absence of allocated budget for such an undertaking. By contrast, adopting the Threefold stack allows us to focus on the project's core objectives while benefiting from an established and reliable technology base.

Secondly, the Threefold stack is open-source and highly versatile, offering the flexibility for future adaptation or even duplication. This ensures that utilizing the Threefold stack now does not limit our options for pivoting to alternative solutions or custom implementations in the future.

Finally, it is worth noting that Threefold Tech is a Belgian company operating within the European Union. This geographical and regulatory alignment could facilitate collaboration, providing the opportunity to negotiate a mutually beneficial way of working with them as part of our approach.

By leveraging the Threefold stack, the project can benefit from an advanced, scalable, and flexible blockchain solution while avoiding unnecessary complexity or resource constraints, enabling us to achieve our objectives efficiently and effectively.

## 5. Conclusions

With this deliverable D5.3, we have initiated the description of a solution that allows to interconnect multiple actors of a Health Data Space in an end-to-end secure and privacy preserving way. It will allow to generate AI models on health data in a way that is compliant with all regulations, such as GDPR, AI Act, Data Governance Act, Data Act and the upcoming EHDS (based on what is known about it now).

Globally, for building the health data hub, we intend to combine **DePIN** with an **identity layer** based on **self-sovereign identity** principles. We believe that combining both techniques will allow us to achieve the outcomes as defined at the start of this project and this WP, i.e. to create “A user-friendly Health Data Hub facilitating easy to use privacy preserving exchange of high-quality health data enabling the innovation of data driven healthcare services”. We added physical hardware infrastructure to the architectural design of the HDH, because we see it as the only full-proof way to design a data hub that also tackles the key non-functional requirements of federation in a secure and privacy-preserving way.

As the technology we aim to implement is cutting-edge though and still subject to heavy research, this document is no more than a reflection of ongoing work. More extensive research and development will be done during M17-M31 of this project and concluded before Milestone 5 on Month 31.

## 6. Annex A: Reference Architectures and DePIN

### 6.1 Significance of Reference Architectures

The architectural design of many modern software systems, such as Data Applications, Data Platforms, and Data Spaces, has grown increasingly difficult due to the size and the complexity of these systems. In this situation, RAs have been demonstrated to be highly pertinent to support the architectural design of systems in numerous essential application domains, including but not limited to, health, avionics, transportation, agriculture, and finance.

Towards this direction, the primary goal of a RA is to provide a common vocabulary, reusable designs, and best practices that are used as a guidance for more concrete software (e.g., systems, platforms) architectures in a specific domain (e.g., healthcare, smart cities, agriculture, finance). Typically, a RA includes *common architecture principles, patterns, building blocks, and standards*, outlining the *components needed to compose a system, the externally visible properties of those components, and the relationships among them*. In essence a RA is not a solution architecture (i.e., they are not implemented directly), but primarily intended to provide a methodology and/or set of practices and templates that are based on the generalization of a set of successful solutions for a particular category of solutions. Hence, a RA provides guidance on how to apply specific patterns and/or practices to solve problems. It acts as a “reference” for the architectures that businesses will use to address their own problems in this way. A RA is never meant to be implemented as-is; rather, it should be utilized as a benchmark or a place to start for the architectural efforts of different organizations. Thus, a RA should be technology and domain-independent, abstract, and flexible, being defined at various levels of detail, from high level principles to detailed implementation guides [10][11].

Summarizing, a RA can serve as a *template that can be used to create a specific architecture (i.e., solution architecture) for a software*. What is worth mentioning is that it can be used to **develop either standard or custom applications or even scalable system solutions** (i.e., solutions scaling up or down as needed the RA’s provided concepts).

By adopting a RA within an organization, the latter is provided with a structured approach to design and deploy software systems, helping to ensure that the final product meets the desired requirements. By leveraging RAs, organizations accelerate delivery through the re-use of an effective solution, thus reducing time to market, improving quality, and increasing efficiency. On top, organizations can be benefitted since a RA provides them with: (i) provision of a *frame of reference* that helps them to get an overview of a particular domain, being provided by a starting point, (ii) systematic *reuse of common functionalities and configurations* throughout the development of their systems, (iii) *risk reduction* through the use of proven and partly qualified architectural elements included in the RA, (iv) enhanced quality by facilitating the achievement of software quality aspects already addressed by the RA, (v) *interoperability* among different systems and their software components establishing common means for information exchange, and (vi) *regulatory compliance* accounting principles, practices, and processes that are already in place.

However, to obtain such benefits, these architectures should be suitably described (i.e., represented/modelled) aiming at reliably communicating the knowledge that they contained [12][13].

Nowadays, there exists a variety of different types of RAs, deriving from diverse application domains [14], all of them however sharing a common goal: to provide a starting point for organizations that need to solve a particular problem. Such architectures have been described in many different approaches, such as using textual description, informal models, modelling languages (e.g., Unified Modelling Language (UML)), and in widely adopted system architecture models (e.g., 4+1 model, C4 model).

The following sub-Sections provide a list of state-of-the-art RAs that have been considered into the realization of the SA, taking into consideration the technical experience and knowhow of the consortium partners in different domains.

## **6.2 Overview of Reference Architectures for Data Marketplaces and Data Spaces**

### **6.2.1 Simpl**

Simpl EC is an **open-source, secure middleware platform designed to support data access and interoperability in European data initiatives**, and more especially a framework to create and orchestrate data spaces. It aims to create a trustworthy platform for data sharing and cloud-to-edge federations, enabling Common European Data Spaces. These data ecosystems will allow users in similar sectors to access data efficiently and safely.

#### **6.2.1.1 Key Features**

- **Secure and interoperable:** Simpl EC ensures trust, trustworthiness, and compliance with regulations, enabling seamless data exchange between participants, regardless of their data processing environments.
- **Open-source:** The platform provides multiple compatible components, free to use, adhering to a common standard of data quality and data sharing.
- **Cloud-to-edge federations:** Simpl EC plays a crucial role in establishing Common European Data Spaces, facilitating data sharing and interoperability among European data initiatives.
- **Data sovereignty, privacy, and fair markets:** The platform supports European values by providing middleware for building data ecosystems and cloud infrastructure services.

The idea behind Simpl is very similar to what we have discovered with DePIN and is what can be highly beneficial to ensure end-to-end security and respect for privacy when exchanging data among actors in the medical domain, which very often host their data at the edge i.e. on their premise.

#### **6.2.1.2 Relevance to PHASE IV AI & Added-Value**

Simpl is, at this stage, **not yet in a state that it can be reused for other projects**, and we expect the Simpl deliverables won't be available in time for PHASE IV AI. At the time of writing, the planning of the Simpl project foresees the integration of AI to be explored not before mid-2025. We will monitor the initiative however and try to evaluate on potential synergies in the final version of this Data Hub Design document, planned for M31 of this project. First open-source deliveries have been presented in January 2025, right before submission of this deliverable.

## 6.2.2 Threefold Tech / DePIN

Threefold Tech has created a, to our knowledge first and only so far, peer-to-peer cloud infrastructure technology. It is a technology that focuses on offering a decentralized cloud set-up, with a focus on security and privacy in the implemented product, as well as enhancing interoperability between hardware infrastructures.

### 6.2.2.1 Architecture Overview

ThreeFold technology aims to meet the demand for decentralisation within the data economy in terms of *decentralised computing & decentralised data*.

- **DECENTRALISED COMPUTING:** the decentralisation of the digital infrastructure is a must to meet the demands of the data economy. Decentralisation in the ownership and physical distribution is a must.
- **DECENTRALISED DATA:** the value of our data in the data economy is priceless but the respect for privacy and our lack of ownership of our data is hugely overlooked. If data is the »oil« of this economy and the collateral for our economic participation, then the sovereignty, ownership and privacy of data are non-negotiable.

The **game-changing elements** in the technology are threefold:



Figure 21 – Threefold key components

#### 6.2.2.1.1 Zero-OS Linux-based, lightweight, decentralized operating system

Zero-OS is an operating system that overcomes all **interoperability issues** and enables a truly **federated data infrastructure**, required a full redesign. Our operating system (Zero-OS), eliminates layers of complexity, ensures maximum efficiency in terms of the use of the infrastructure and ensuring absolute security by removing all human interface and hacking surface, making it the world’s first decentralized OS.

**Zero-OS** is the foundational OS on the capacity layer. It delivers the required hardware capacity for any Linux IT solution. Any application which can run on Linux can run on our Zero-OS, and can exploit this hardware, bringing more decentralization, security, and efficiency. The Zero-OS Operating System has been designed from the bottom up: we started from a Linux kernel and secure boot BIOS – everything else in the operating system has been created from scratch. It combines 3 primitive functions: **storage capacity**, **compute capacity**, and **intelligent network functions for running the network services**. Our **distributed storage layer is**

**unique** and allows storage of exabytes of information at a lower cost compared to any other technology while maintaining a much higher privacy and reliability level.

Capacity in any size hosted on secure bootable devices (any Intel/AMD HW) can be added to the TF Grid.

**Privacy** is built into this layer of the technology through different measures:

The data storage is dispersed over different Zero-OS nodes in a way that stored data on an individual node is meaningless. The same mechanisms are used as in Data Centre RAID systems, with the difference that storage is dispersed here over different capacity owners, making the truth is no longer held in one location.

The following, simplified example shows how that works:

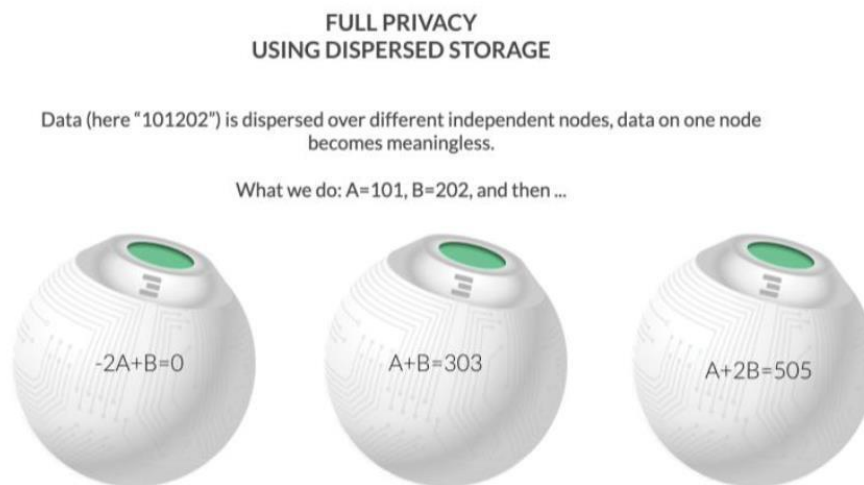


Figure 22 – Threefold Dispersed Storage

The drawing shows the principle in a simplified example (we evidently have binary numbers and more than two dimensions in the equations), but it explains the basic principles:

Low-level storage is split in an intelligent way into different shards, in a way that on one location, there is only part of the information stored. Moreover, the data is described in a descriptive way (through equations) so that a person aiming to hack into the low-level data (which is almost impossible in itself), will only find non-relevant information on this storage infrastructure. The fact that no data shard can be created only accessing one location, makes it hackerproof when attacks happen on one location, as no compute power can ‘imagine’ what these missing data are.

The dispersed storage mechanism is the base that we will use to make fully quantum proof storage infrastructure. In order to be quantum proof, we need to go one step further. By only dispersing the data/storage (a.o. public and private key data), there is still one weak point, which is the metadata describing the location where the different data shards are located.

**Hardware capacity owners**, called **farmers**, rent out their storage but have no access to the data or apps that are being hosted. The dispersed storage system ensures data is dispersed over multiple nodes (*dispersed, not replicated*) and can only be viewed/assembled by the data owner using own private key.

The redundancy process automatically regenerates the workload after detection that some infrastructure is unavailable (due to power outage or hardware that fails to operate correctly).

Networking is set up through a secure meshed overlay IPv6-networking platform allowing any compute and storage workload to be connected and exposed to the existing internet network. This peer-to-peer network platform (based on Wireguard open-source technology) allows any workloads to be connected over secure encrypted networks which will look for the shortest path between the nodes., encrypting all traffic between workloads (*containers*) on the network. Also there, capacity owners have no view on the content of the traffic generated on their infrastructure. The network can even be secured more as connections can only be established from the inside. We call it a “network wall”. Networks are created 100% peer 2 peer. No connection is made with the public internet. Everything stays 100% private.

Each container has a secure connection to the web gateway through a socket.

The container calls out to one or more web gateways and sets up a secure & private socket connection to the web gateway. The connection required is defined on the smart contract for IT layer and as such is very securely defined. There is no IP (TCP/UDP) coming from the internet towards the containers providing more security.

On the web gateways DNS interface is provided and HTTP(s) or TCP gets answered on the web gateway layer. The payload (socket level or http level) gets forwarded over the secure socket connection to the container.

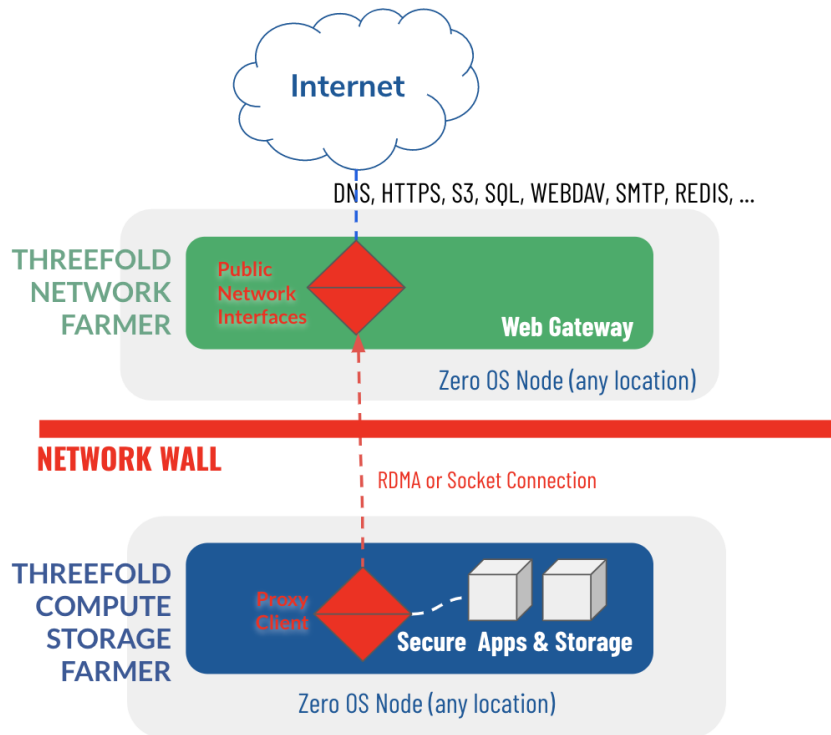


Figure 23 – Threefold Network Wall

Full redundancy can be achieved: any app can get connected to any chosen IP address of ThreeFold network farmers. An app can be connected to multiple gateways at once which provides good redundancy.

This allows easy clustering mechanism, where web gateways and nodes can be lost, and the service will still be up and running.

When containers are moved or re-created the same end user connection can be reused.

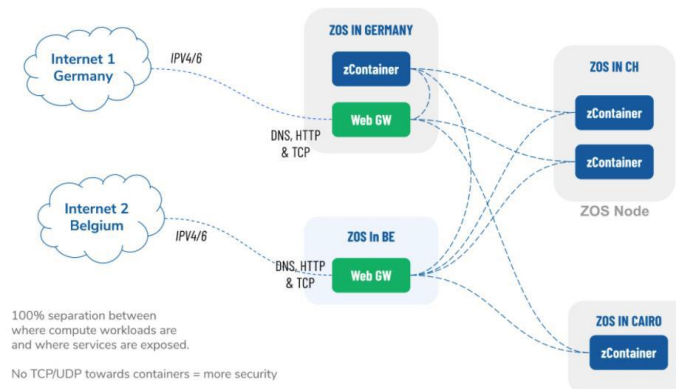


Figure 24 – Threefold Web GW

In conclusion, the content of the nodes is invisible to capacity owners (farmers)

- Zero-OS comes without a shell.
- Networking is encrypted between nodes through the secure overlay network technology.
- The network connections can only be established from inside out.

The private environment for a user's workload in Zero-OS can only be accessed by a PKI enabled virtual system administrator (VSA, see below), making the owner of the keys in full control of anything he runs on the infrastructure. Also, this is an important privacy feature. It disables the freeway for abuse currently existing in business models like Facebook or Google.

The **Virtual System Administrator (VSA)**, acting on your behalf to deploy workloads on the DePIN grid / on the internet. This VSA, under full control and only listening to the one owning the user's individual private keys, ensures an immutable record of any workload, which also enables the self-healing functionality as any workload can easily be restored if/when needed. With the VSA peer-to-peer architecture, everyone is equal, and no centralized institution nor big tech corporation can acquire control over one's digital information or applications. Secure, decentralized two-factor authentication login, a decentralized ledger and lean coding ensure highest security, based on PKI technology. The VSA is identified by a key pair, and the key for a user launching IT workloads to remain in full control over his IT workloads.

#### 6.2.2.1.2 Zero-Chain and the 'Smart Contract for IT'

Threefold has already integrated blockchain technology into its model and has implemented its first version of a 'Smart Contract for IT'. Deployment of IT workload using a so-called "**Smart Contract for IT**" makes the deployment process of IT workload resilient to human error and hacking. The system is **self-driving** and **self-healing**, therefore removing the human requirement for deploying and operating IT infrastructure and services. **This represents a breakthrough in IT.** The VSA records all transactions on TFChain blockchain infrastructure, ensuring a personalized immutable record of any workload. This enables self-healing, i.e. the restoration of a workload if needed, due to failing hardware, dropout in electricity etc.

There are no people involved (**IT architectures are configured and installed by bots**) to run and keep it operational (**no human intervention and access possible**). This is a very different way of thinking – it leads to much more security, efficiency, and higher performance, and will lead to huge network savings.

The 5-step process is as follows (all exists, apart from step 1b, see below):

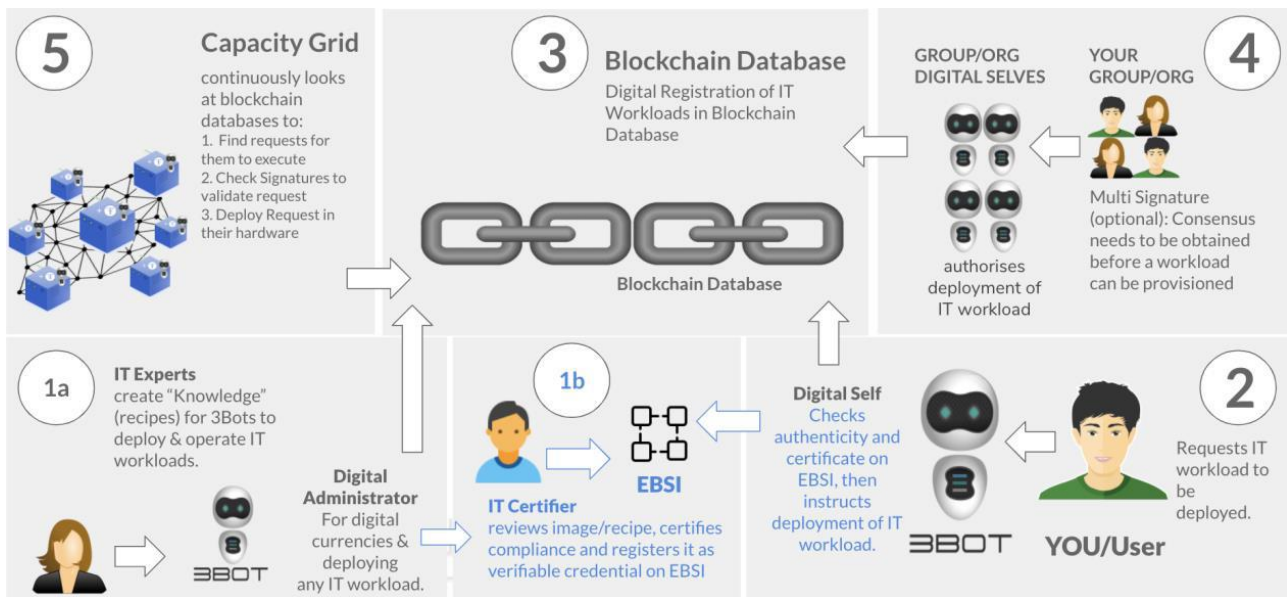


Figure 25 – Threefold Smart Contract for IT

**STEP 1: IT Experts create smart contracts:**

IT experts create smart contracts describing what needs to be done to deploy this architecture. The smart contract has to be specific and describe each little detail of the IT architecture. The experts create deployable workloads/scripts.

**STEP 2: Business and or End-user customers consume smart contracts:**

Users have digital needs and to procure services for their digital needs they will find smart contracts describing applications (application setups) meeting their needs. Consumers will instruct their VSA to deploy an IT workload following their requirements by using a smart contract created e.g. give me an archive of 1 PB in CH, e.g. deploy a CRM for 100 users, ...

- e.g. deploy a certified de-identification software on hardware on my premise
- e.g. deploy an AI model training software

This gets translated to a very low-level description of the IT workload as required (networking topology, files needed, processes to start, ...)

**STEP 3: Registration of the smart contract into a blockchain registry (used as a digital notary system):**

Creates & Registers the “IT” smart contract on TFChain, a permissioned blockchain registry. The Zero-OS nodes in cooperation with the smart contract execution code, will provision all the compute and storage capacity needed to meet the IT architecture’s requirement and do all the commercial trades required to get this. It will then leave instructions for the nodes in the digital notary system for nodes to be able to grab instructions on what they have to do in order to meet smart contract completion.

**STEP 4: Business IT Workload Stakeholders approve the workload execution:**

is optional but when required stakeholder can be defined to give consensus and sign off on the successful execution of the “IT smart contract” delivering the appropriate digital service. Stakeholders can be defined in a “multi signature” blockchain to provide sign off on regulatory, commercial, and other business requirements.

**STEP 5: Execution on the capacity layer (3Nodes)**

- Thousands of 3Nodes can work together to execute and deliver the “IT Smart Contract” (if required)
- Verify if consensus was reached between the business stakeholders
- Verify the validity of the smart contract and download the “IT workload definition”.

- Download the right files to execute the smart contract and each file gets verified (signature)
- Run the required processes and again signatures are checked to make sure the workload is pure.
- Ensures that no person (hacker or IT person) can ever gain access or influence on the execution process.

### 6.2.2.2 Relevance to PHASE IV AI & Added-Value

Threefold technology is, as far as we have identified, the only technology that is readily available to address the need of PHASE IV AI to interconnect multiple independent organizations in a secure and privacy-preserving way down to the hardware infrastructure layer. We also believe that taking up this hardware infrastructure layer of compute, storage and network, orchestrated by a federated yet IT industry compliant operating system, as a core layer for our architecture is essential to guarantee end-to-end security and privacy-by-design.

Building an architecture with security, privacy and federation principles addressed in its foundational design gives the best guarantee that these very hard-to-tackle non-functional requirements can be dealt with without flaws.

Threefold has been identified as the only DePIN technology that has implementation specifications for everything that software requires (storage, compute, networking, memory and an operating system) in a combined way.

Other DePIN technology only focus on specific hardware capabilities:

- Storage: Filecoin, Storj, Bluzelle
- Compute: Golem, Mawari
- Network: Helium

ICP (Internet Computer Protocol) focuses on all but derives from the established IT standards and is limited in applicability, making it currently unfit for our project.

### 6.2.3 IDS

The International Data Spaces Association (IDSA) [32] functions as a virtual data realm that utilizes established standards, technologies, and widely embraced governance models within the data economy. Its primary goal is to enable secure and standardized data exchange and data linkage in a trusted business ecosystem. This establishes a foundation for developing smart-service scenarios and promoting innovative cross-company business processes, all while ensuring data sovereignty for data owners.

Data sovereignty is a central aspect of the International Data Spaces (IDS). This entails the inherent ability of individuals or organizations to have complete self-governance over their data. The initiative of IDS introduces a Reference Architecture Model (RAM) that encompasses this very ability, along with associated factors such as the necessity for secure and trustworthy data interchange within business ecosystems.

#### 6.2.3.1 Architecture Overview

IDS-RAM was originally defined as part of the research activities conducted in the Industrial Data Space project by Fraunhofer [33] and continues to evolve through the work of many research and industrial projects under the steering of the IDSA Architecture working group. Furthermore, IDSA promotes the IDS-RAM, IDS implementations and use cases, to establish an international standard for secure data exchange and data sharing facilitated by the IDS Connector, the central technical component of the International Data Spaces.

Focusing on the broad conceptualization of functionalities, capabilities, and the overarching processes engaged in establishing a *secure network of trusted data*, the IDS-RAM exists at a more elevated level of abstraction compared to typical architectural models of specific software implementations. Figure 26 illustrates the general structure of the IDS-RAM that uses five (5) layers and three (3) perspectives.

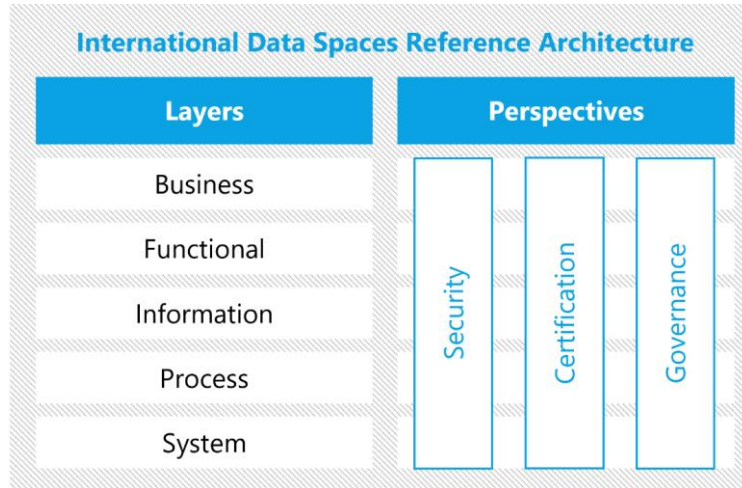


Figure 26 – General structure of IDS Reference Architecture Model

- *Business Layer* [34]: Specifies and categorizes the different roles which the participants of the International Data Spaces can assume, and it specifies the main activities and interactions connected with each of these roles. The Business Layer specifies the requirements such as establishing trust and technical frameworks for technically enforced agreements to be addressed in the Functional Layer and provides an abstract description that can be considered as a blueprint for the other, more technical layers.
- *Functional Layer* [35]: Defines the functional requirements of the IDS and the concrete features to be derived from the: (i) Trust that represents the fundamental features of data spaces, the roles, the identity management, and the user certification, (ii) Security and Data Sovereignty for performing authentication/authorization, usage policies usage enforcement, trustworthy communication security by design, and technical certification, (iii) Data Ecosystem that consists of the data source’s description, metadata brokering, and vocabularies, (iv) Standardized Interoperability, (v) Value Adding Applications, and (vi) Data Markets that consider the monetary value concepts of data like clearing and billing, and governance.
- *Information Layer* [36]: Defines a conceptual model that makes use of linked-data principles for describing both the static and the dynamic aspects of the IDS constituents.
- *Process Layer* [37]: Specifies the interactions taking place between the different components of IDS; using the BPMN notation, it provides a dynamic view of the RAM. The following processes and their sub-processes are included: (i) Onboarding, (ii) Data Offering, (iii) Contract Negotiation, (iv) Exchanging Data, and (v) Publishing and using Data Apps.
- *System Layer* [38]: Is concerned with the decomposition of the logical software components, considering aspects such as integration, configuration, deployment, and extensibility of these components.

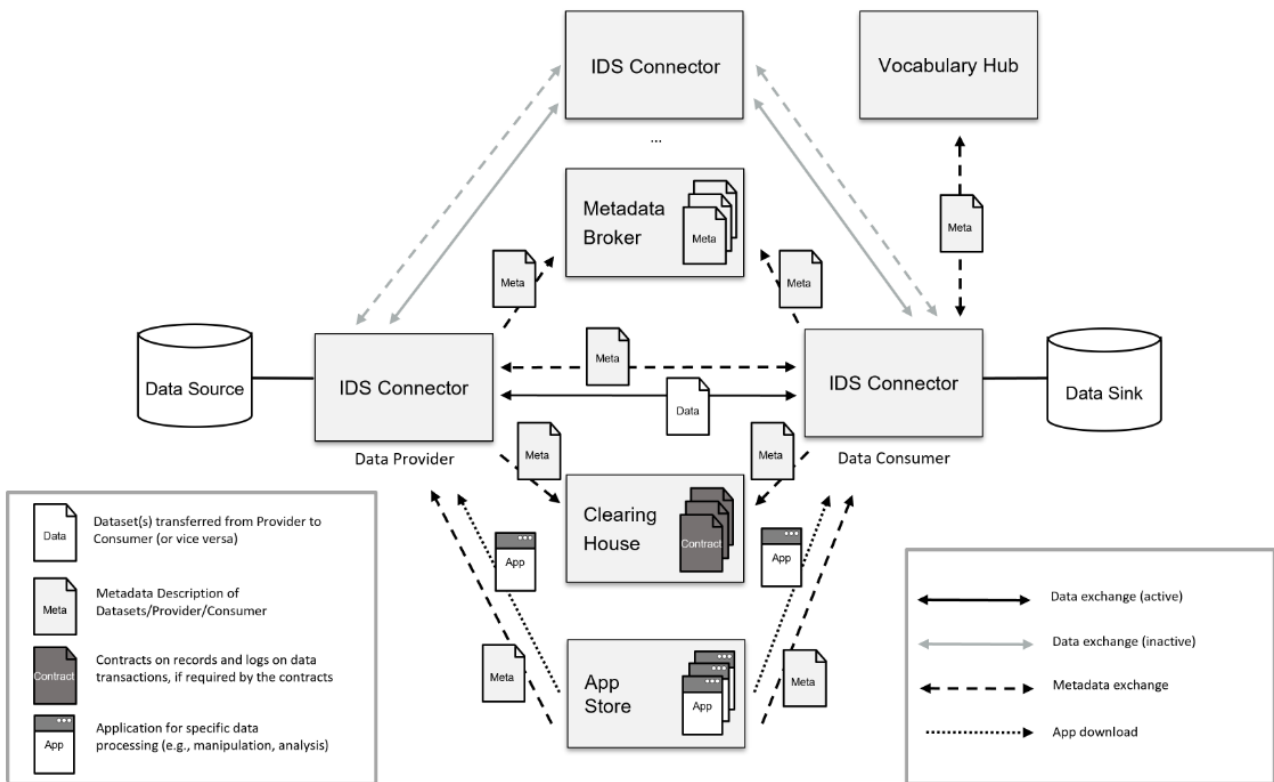


Figure 27 – Interaction of technical components

Figure 27 illustrates the interaction among the various existing technical components. It is crucial to note that the actual data is solely transmitted from the Data Provider to the Data Consumer. The remaining interactions with the other components rely on Metadata. Metadata delineates the appearance of the data, its format, user policies, data ownership, and more. The Metadata is submitted to a broker due to the absence of a centralized data lake. Connectors need to locate each other; hence they furnish the broker with their connection details and Metadata. A connector has the potential for augmentation with Data Apps. These Apps can be installed from an App Store within the connector to alter data before transmission or to perform analytical functions on the data within the connector.

In addition, as stated above, the IDS-RAM comprises three (3) perspectives that need to be implemented across all the five (5) layers:

- **Security** [39]: The IDS Security Architecture provides means to identify devices in the IDS, protect communication and data exchange transactions, and control the use of data after it has been exchanged.
- **Certification** [40]: Any organization or individual seeking permission to operate components in the International Data Spaces needs to pass the Operational Environment Certification that ensures secure processes and management of components.
- **Governance** [41]: The Governance Perspective defines the roles, functions, and processes of the International Data Spaces from a governance and compliance point of view.

### 6.2.3.2 Relevance to PHASE IV AI & Added-Value

IDS is mainly active in doing foundational work that is relevant for data spaces, to be taken up by projects applying these foundations. The work of IDS, such as the Data Space Protocol, is very relevant for further exploration, both directly and through the frameworks being developed, integrating the concepts that have been initiated within IDS.

## 6.2.4 GAIA-X

Gaia-X [49] is a framework being created for a Federated and Secure Data Infrastructure, having as a primary goal the innovation through digital sovereignty. This is achieved by establishing a decentralized ecosystem in which data is made available, collated, and shared in a trustworthy environment, where users always retain sovereignty over their data. Towards this direction, the Gaia-X community consists of multiple stakeholders who are specifying and developing a set of functional and interoperable components consisting of: (i) Federation Services and other technical components, (ii) a Governance Framework, and (iii) a Trust Framework.

To make the Gaia-X concept operational, the Gaia-X Federation Services (GXFS) toolbox [50] has been developed, aiming to provide the minimum technical requirements/set of services needed to build and operate this cloud-based, self-managed data infrastructure ecosystem.

### 6.2.4.1 GXFS Elements

To be more specific, GXFS consists of several components (Figure 28) enabling federations in data ecosystems and providing interoperability across federations, being the OSS Toolbox for Building Federated Data Ecosystems. These components are categorized into the five (5) core groups of (i) Identity and Trust, (ii) Sovereign Data Exchange, (iii) Sovereign Federated Catalogue, (iv) Compliance, and (v) Portal, as further described below.

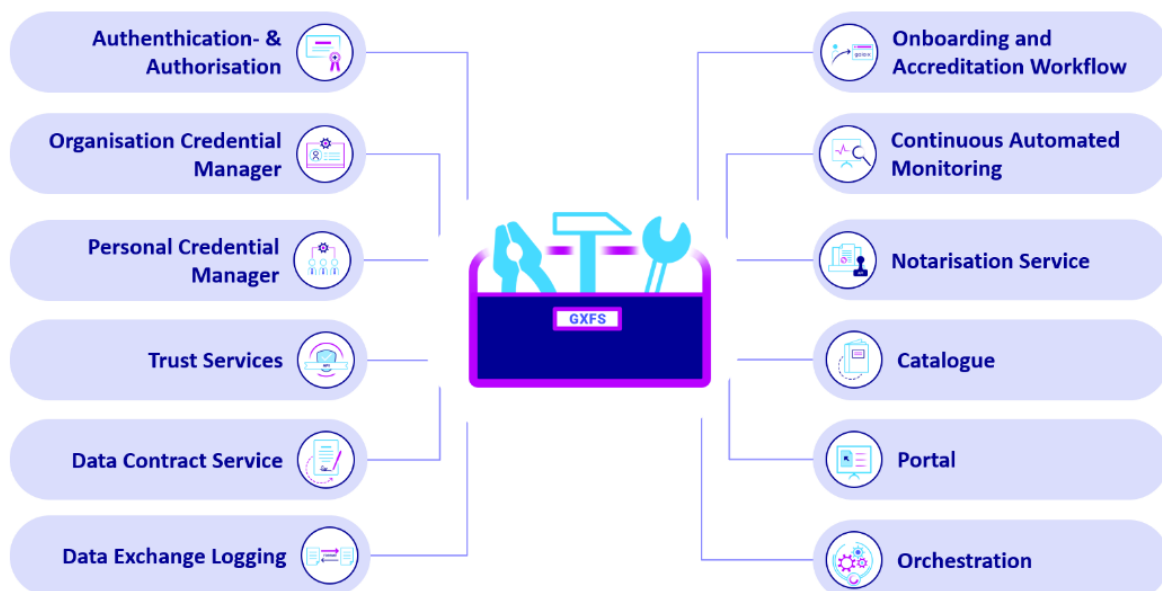


Figure 28 – GXFS supported elements

- Identity and Trust Elements

- *Authentication & Authorization* [51]: This service enables Gaia-X participants to authenticate other users and systems in a trusted, decentralized, and self-sovereign manner without the need for a central source of authority.
  - Request verifiable, decentralized, and cryptographic credentials and identity attributes from other participants in a Federation.
  - Maintain control over what information is shared with others.
- *Organization Credential Manager* [52]: This service establishes trust between the different participants within the decentralized Gaia-X ecosystem. It includes all trust-related functions required to manage and offer Gaia-X self-descriptions in the W3C Verifiable Credential Format, providing the following core functionalities:
  - Configure a self-determined and easy entry into a Federation for companies by, e.g., independently issuing digital participation credentials to employees.
  - Provide software services and data assets with a digitally verifiable seal.
  - Create cryptographic verifiable Self-Descriptions.
  - Manage credentials and certificates of employees, data and services.
- *Personal Credential Manager* [53]: This service enables Gaia-X users to manage their credentials themselves. To do this, the user needs secure storage (user wallet) and presentation capabilities in the authentication and authorization processes. Hence, the supported functionalities refer to:
  - Manage self-sovereign of one's own credentials, e.g., identity documents, certificates, or authorizations of individual participants (self-employed or employed).
  - Maintain control over which credentials are used for authentication and authorization purposes.
  - Authenticate using a mobile app or browser application.
  - Authenticate and authorize natural persons as well as machines and digital twins to enable trust-based machine-to-machine communication.
- *Trust Service's API* [54]: This service ensures that a consistent level of trust can be established between all the components and the participants in a Gaia-X ecosystem. They are the central, technical implementation of cryptographic functions for enforcing policies in the SSI context for the use of the capabilities provided in a decentralized and self-governing manner. Consequently, the supported include:
  - Enforce usage policies.
  - Ensure chains of trust among multiple participants, organizations, and authorities.
  - Establish trust anchors with verification standards such as W3C Verifiable Credentials/Presentations.
  - Establish rule-based trust on an attribute basis.
- *Sovereign Data Exchange Elements*
  - *Data Contract Service* [55]: This service enables data exchange in a secure, trustworthy, and auditable way in a Gaia-X ecosystem. It provides interfaces for negotiating data contracts that define the agreed terms (Data Asset Usage Policy) for the planned data exchange. Thus, the supported functionalities include to:
    - Obtain legally binding consent between data provider and data user for data access, exchange, and use.
    - Sign cryptographically a data contract.
    - Subsequent provision of the signed contract.
  - *Data Exchange Logging Service* [56]: This service is used to run evidence whether data has been transmitted, received and rules and terms of use (data usage policies) have been respected or not within the Gaia-X ecosystem, offering functionalities to:
    - Track whether data has been transmitted and received or not.

- Track whether data usage policies have been respected or violated, e.g., to clarify operational issues or detect fraudulent transactions.
- Create an auditable transaction log that is only accessible to the contracting parties.
- *Sovereign Federated Catalogue Elements*
  - *Federated Catalogue* [57]: This service includes a catalog where Gaia-X resources, asset items, and participants can be found by potential consumers and end-users. Resources, asset items and participants are provided at Gaia-X using self-descriptions. Hence, the offered functionalities refer to:
    - Search and select providers and their service offerings in a Federation based on self-descriptions.
    - Monitor relevant changes in service provision.
    - Support of Self-Description Tools, including a Creation Wizard (for creating valid Self-Descriptions (claims) using interactive web forms), a Visualization Tool (for visualizing created Self-Descriptions, and a Validation Wizard (for validating the created Self-Descriptions (claims), e.g., check whether data types are correct and all mandatory information is present).
- *Compliance Elements*
  - *Authentication & Authorization*
    - Support the implementation of a validation process for participants, resources, and service provision prior to inclusion in a Federation's catalogue.
    - Document the validation process and create an audit trail to ensure compliance with generally accepted conformity assessment practices.
  - *Continuous Automated Monitoring* [58]: This service provides Gaia-X users with transparency about whether individual service offerings in a Gaia-X Federated Catalog are compliant with the rules or not. This compliance is based on certain requirements and rules that Gaia-X itself has set for its system. Thus, its core functionality is to:
    - Continuously automate rule compliance monitoring based on Self-Descriptions in a Federation's catalogue.
  - *Onboarding & Accreditation Workflows* [59]: This service ensures that all participants and offerings within the Gaia-X ecosystem undergo a validation process before being added to the Federated Catalog.
  - *Notarization API* [60]: This service authenticates given master data and transforms it into a W3C-compliant, digitally verifiable representation. These tamper-proof digital assertions about specific attributes are central to gaining the desired trust in provided self-descriptions of assets and participants. As for the service's supported functionalities these refer to:
    - Issue a verifiable credential following successful validation of a participant to confirm status as a registered participant in a Federation.
    - Process notarization requests and issue digital, legally binding, and trustworthy credentials.
- *Portal Elements*
  - *Portal* [61]: This service serves as a RA for interacting with core service functions via an intuitive user interface and corresponding backend implementation functions. The user interface provides mechanisms for interacting with core functions via API calls. As for the service's supported functionalities these refer to:
    - Operate a business web client for each Federation.
    - Integrate the individual Federation Services such as: (i) querying Federation databases and displaying search results for services and data within a Federation, (ii) profile management of member's account to create and edit self-descriptions, organizational data, login history, etc., (iii) credentials management, (iv) addition and authorization

of new members of a Federation, and (v) dashboard with overview of all the active and inactive services, the status of the booked services, as well as the history of the used services.

- *Orchestration* [62]: This service allows Gaia-X consumers to instantiate and manage infrastructure services, such as virtual machines, from the Federated Catalog search results via the Gaia-X portal.
- *IDM & Trust Architecture*:
  - Decentralized identity management.
  - Trust Layer with signature and validation mechanisms.
  - Service components/features supporting on-/offboarding processes.
  - Access management.

#### **6.2.4.2 Relevance to PHASE IV AI & Added-Value**

Gaia-X has merely focused on the Trust Framework and the use of Verifiable Credentials to be used for organisations and services. It is one of the more mature frameworks we have identified around SSI and VCs, and especially the possibility to not only certify persons and organisations but also services is very relevant. There is however some more in-depth analysis required to check on what point the Gaia-X delivery will be stable and ready for consumption. We hope that by Milestone 5 (M31), the framework will have come to a sufficient maturity to integrate the services into our overall implementation.

## 7. Annex B: Private Overlay Network Technologies

### 7.1 Private Overlay Network technologies

This annex gives an overview of the different private overlay network technologies that we can implement on the infrastructure consumption side to interconnect users in a secure way.

#### 7.1.1 Mycelium

- [https://manual.grid.tf/documentation/system\\_administrators/mycelium/overview.html](https://manual.grid.tf/documentation/system_administrators/mycelium/overview.html)
- <https://github.com/threefoldtech/mycelium>
- Less battle tested, experimental
- Based on Babel routing protocol
- Implementation on Threefold grid off the shelf
- Availability of a shared key that controls network membership

##### Features

- Mycelium is locality aware, it will look for the shortest path between nodes.
- All traffic between the nodes is end-2-end encrypted.
- Traffic can be routed over nodes of friends, location aware.
- If a physical link goes down Mycelium will automatically reroute your traffic
- The IP address is IPV6 and linked to private key.
- A simple reliable messagebus is implemented on top of Mycelium.
- Mycelium has multiple ways how to communicate quic, tcp, ... and we are working on holepunching for Quick which means P2P traffic without middlemen for NATted networks e.g. most homes.
- Scalability is very important for us, we tried many overlay networks before and got stuck on all of them, we are trying to design a network which scales to a planetary level.
- You can run mycelium without TUN and only use it as reliable message bus.
- Integration ready with Zero-OS operating system.

#### 7.1.2 Yggdrasil

<https://yggdrasil-network.github.io/>

- End-to-end encryption over IPv6 or IPv4
- Experimental, alpha stage
- Built on Yggdrasil routing protocol
- Node's identity = its public key

##### What are the benefits?

There are several benefits to a routing scheme such as this:

- Devices need to only maintain a comparatively small amount of state in order to function and to be able to forward packets — there is no need for any Yggdrasil node to maintain “full routing tables” like in BGP, and most nodes only have a handful of routing table entries in total.
- Paths are discovered and built through the network automatically, so manual configuration of routing entries is not required — the only configuration needed is the peering connections between nodes themselves.

- The network can setup and tear down paths quickly without needing to discard all routing state, which helps significantly in handling node mobility events without dropping many packets if at all.
- We can bridge reliable/static networks very easily with dynamic/non-static networks without needing to flood large amounts of state.
- Networks automatically form when any two or more Yggdrasil nodes are connected to each other, even if those connections are entirely ad-hoc in nature, which allows building true wireless mesh networks.
- Sparse routing knowledge and only small amounts of protocol traffic should mean that Yggdrasil is able to efficiently scale to very large networks.

### 7.1.3 Tailscale

<https://tailscale.com/blog/how-tailscale-works>

[What is Tailscale? · Tailscale Docs](#)

- Very simple to set up.
- Built on WireGuard.
- The Tailscale approach avoids centralization where possible, resulting in both higher throughput and lower latency as network traffic can flow directly between machines. Additionally, decentralization improves stability and reliability by reducing single points of failure.
- Tailscale is simple and effortless. The service handles complex network configuration on your behalf so that you don't have to. Network connections between devices pierce through firewalls and routers as if they weren't there, allowing for direct connections without the need to manually configure port forwarding. It allows for connection migration so that existing connections stay alive even when switching between different networks (for example, wired, cellular, and Wi-Fi). With MagicDNS, you don't have to deal with IP addresses – you can SSH or FTP into your device, transfer files between devices, or access a web server or database by just using a memorable hostname.

### 7.1.4 Netmaker

<https://www.netmaker.io/resources/overlay-networks#toc-overlay-network-overview>

- Netmaker can only be selfhosted.
- Set-up is pretty complicated.
- Built on WireGuard.

### 7.1.5 Netbird

<https://docs.netbird.io/>

- Very simple to set up
- Built on WireGuard
- 

### 7.1.6 Twingate

- Built on WireGuard
- Focus on Zero Trust
- Focused on connecting services rather than devices.

- Least privilege.
- Micro-segmentation
- Attribute-based access.
- Ability to route by DNS.
- No self-hosted option
- Considered as Not fit for PHASE IV AI

### 7.1.7 Zrok.io

- Zero-trust overlay network

SaaS or self-hosted

### 7.1.8 OpenZiti

<https://github.com/openziti>

[What is OpenZiti? | OpenZiti](#)

- Zero Trust Overlay network
- Open source
- Can be self-hosted.
- Many networking security solutions act like a wall around an internal network. Once you are through the wall, you have access to everything inside. Zero trust solutions enforce not just access to a network, but access to individual applications within that network.
- Every client in a Ziti system must have an identity with provisioned certificates. The certificates are used to establish secure communications channels as well as for authentication and authorization of the associated identity. Whenever the client attempts to access a network application, Ziti will first ensure that the identity has access to the application. If access is revoked, open network connections will be closed.
- This model enables Ziti systems to provide access to multiple applications while ensuring that clients only get access to those applications to which they have been granted access.
- In addition to requiring cert-based authentication for clients, Ziti uses certificates to authorize communication between Ziti components.

## 8. Annex C : Flist creation steps

The creation of a flist is a process, which consists of the following high-level main steps:

- Create a workload in a binary format, ex. a Docker image.
- Push the Docker image to the Docker Hub.
- Convert the Docker image to a Zero-OS flist.
- Deploy a micro VM with the flist in a portal (ex. PHASE IV AI model generation portal)

We present an example based on Docker.

### 8.1.1.1.1 Docker Image Creation

A detailed description of how to create an flist from scratch will be elaborated in the future version of the guide. For now, we want to take existing codes and further elaborate on them. This is not only quicker, but it is also a good way to get to know the ThreeFold's ecosystem and repositories.

We will be using the code available on the [ThreeFold Tech's Github page](#). In our case, we want to explore the repository [tf-images](#).

For this purpose, we take an example deploying the Debian Linux distribution.

For this case study, we built inspiration from the [Ubuntu 22.04](#) directory.

The Ubuntu 22.04 directory tree shows the following:

```
.
├── Dockerfile
├── README.md
├── start.sh
├── zinit
│   ├── ssh-init.yaml
│   └── sshd.yaml
```

In the following sections, we explore each of those files to get a better view at the whole repository to understand how all elements work together.

#### 8.1.1.1.1.1 [Dockerfile](#)

To make a Docker image, it is necessary to create a Dockerfile. As per [Docker's documentation](#), this is "a text document that contains all the commands a user could call on the command line to assemble an image".

The Ubuntu 22.04 Dockerfile is as follows:

File: Dockerfile

```
FROM ubuntu:22.04

RUN apt update && \
  apt -y install wget openssh-server

RUN wget -O /sbin/zinit
https://github.com/threefoldtech/zinit/releases/download/v0.2.5/zinit && \
  chmod +x /sbin/zinit

COPY zinit /etc/zinit
COPY start.sh /start.sh

RUN chmod +x /sbin/zinit && chmod +x /start.sh
ENTRYPOINT ["zinit", "init"]
```

We can see from the first line that the Docker file will look for the docker image `ubuntu:22.04`. In our case, we want to get the Debian 12 docker image. This information is available on the Docker Hub (see [Debian Docker Hub](#)).

We will thus need to change the line `FROM ubuntu:22.04` to the line `FROM debian:12`. It isn't more complicated than that!

We now have the following Docker file for the Debian docker image:

File: Dockerfile

```
FROM debian:12

RUN apt update && \
  apt -y install wget openssh-server

RUN wget -O /sbin/zinit
https://github.com/threefoldtech/zinit/releases/download/v0.2.5/zinit && \
  chmod +x /sbin/zinit

COPY zinit /etc/zinit
COPY start.sh /start.sh

RUN chmod +x /sbin/zinit && chmod +x /start.sh
ENTRYPOINT ["zinit", "init"]
```

#### 8.1.1.1.2 [Docker Image Script](#)

The other important file we will be looking at is the `start.sh` file. This is the basic script that will be used to properly set the docker image. Since there is nothing more to change in this file, we can leave it as is. As we will see later, this file will be executed by zinit when the container starts.

File: `start.sh`

```
#!/bin/bash
```

```
mkdir -p /var/run/sshd
mkdir -p /root/.ssh
touch /root/.ssh/authorized_keys

chmod 700 /root/.ssh
chmod 600 /root/.ssh/authorized_keys

echo "$SSH_KEY" >> /root/.ssh/authorized_keys
```

#### 8.1.1.1.1.3 [zinit Folder](#)

Next, we want to take a look at the zinit folder.

Zinit is a process manager (pid 1) that knows how to launch, monitor and sort dependencies. It thus executes targets in the proper order. For more information on zinit, check the [zinit repository](#).

When we start the Docker container, the files in the folder zinit will be executed.

If we take a look at the file `ssh-init.yaml`, we find the following:

```
exec: bash /start.sh
log: stdout
oneshot: true
```

We can see that the first line calls the [bash](#) Unix shell and that it will run the file `start.sh` as seen earlier.

In this zinit service file, we define a service named `ssh-init.yaml`, where we tell zinit which commands to execute (here `bash /start.sh`), where to log (here in `stdout`) and where `oneshot` is set to `true` (meaning that it should only be executed once).

If we look at the file `sshd.yaml`, we find the following:

```
exec: bash -c "/usr/sbin/sshd -D"
after:
  - ssh-init
```

Here another service `sshd.yaml` runs after the `ssh-init.yaml` process.

#### 8.1.1.1.1.4 [README.md File](#)

In the `README.md` file, we can explain what our code is doing and offer steps to properly configure the whole deployment. For the users that will want to deploy the flist on the ThreeFold Playground, they would need the flist URL and the basic steps to deploy a Micro VM on the TFGrid. We will thus add this information in the `README.md` file. This information can be seen in the [section below](#). To read the complete `README.md` file, go to [this link](#).

#### 8.1.1.1.1.5 [Putting it All Together](#)

We've now gone through all the files available in the Ubuntu 22.04 directory on the tf-images repository. To build your own image, you simply need to put all those files in a local folder on your computer and follow the steps presented at the next section, [Docker Publishing Steps](#).

To have a look at the final result of the changes we bring to the Ubuntu 22.04 version, have a look at this [Debian directory](#) on the ThreeFold's tf-images repository.

#### 8.1.1.1.2 [Docker Publishing Steps](#)

##### 8.1.1.1.2.1 [Create Account and Access Token](#)

To be able to push Docker images to the Docker Hub, it is necessary to create a Docker Hub account.

The steps to create an account and an access token are the following.

- Go to the [Docker Hub](#)
- Click `Register` and follow the steps given by Docker
- On the top right corner, click on your account name and select `Account Settings`
- On the left menu, click on `Security`
- Click on `New Access Token`
- Choose an Access Token description that you will easily identify then click `Generate`
  - Make sure to set the permissions `Read, Write, Delete`
- Follow the steps given to properly connect your local computer to the Docker Hub
  - Run `docker login -u <account_name>`
  - Set the password

You now have access to the Docker Hub from your local computer. We will then proceed to push the Docker image we've created.

##### 8.1.1.1.2.2 [Build and Push the Docker Image](#)

- Make sure the Docker Daemon is running
- Build the docker container
  - Template:

```
docker build -t <docker_username>/<docker_repo_name>
```

- Example:

```
docker build -t username/debian12
```

- Push the docker container to the [Docker Hub](#)

- Template:

```
docker push <your_username>/<docker_repo_name>
```

- Example:

```
docker push username/debian12
```

- You should now see your docker image on the [Docker Hub](#) when you go into the menu option My Profile.

- Note that you can access this link quickly with the following template:

```
https://hub.docker.com/u/<account_name>
```

#### 8.1.1.1.3 [Convert the Docker Image to an Flist](#)

We will now convert the Docker image into a Zero-OS flist. This can be done by means of the following easy procedure:

- Go to the [ThreeFold Hub](#).
- Sign in with the ThreeFold Connect app.
- Go to the [Docker Hub Converter](#) section.
- Next to `Docker Image Name`, add the docker image repository and name, see the example below:

- Template:

```
<docker_username>/docker_image_name:tagname
```

- Example:

```
username/debian12:latest
```

- Click `Convert the docker image`
- Once the conversion is done, the flist is available as a public link on the ThreeFold Hub.
- To get the flist URL, go to the [TF Hub main page](#), scroll down to your VSA ID and click on it.
- Under `Name`, you will see all your available flists.

- Right-click on the flist you want and select `Copy Clean Link`. This URL will be used when deploying on the ThreeFold Playground. We show below the template and an example of what the flist URL looks like.

- Template:

```
https://hub.grid.tf/<VSA_name.3bot>/<docker_username>-  
<docker_image_name>-<tagname>.flist
```

- Example:

```
https://hub.grid.tf/idrnd.3bot/username-debian12-latest.flist
```

#### 8.1.1.1.4 [Deploy the Flist using IaC](#)

A flist can then be deployed on any infrastructure using Infrastructure-as-Code tooling, such as Terraform or Pulumi. More info on this is available 4.6.9.

## 9. Annex D: Bootstrapping a Threefold node

### 9.1 An easy way to make hardware available to a secure network

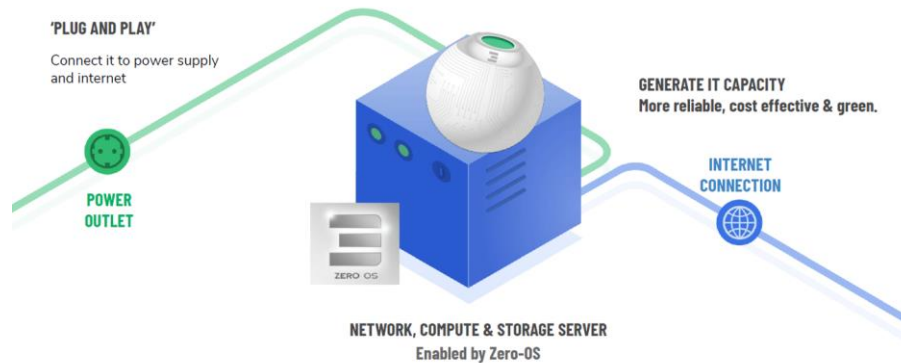


Figure 29 – Plug-and-Play Hardware Infrastructure Provisioning

Installing and bootstrapping an 3Node (= hardware infrastructure running Zero-OS federated operating system) is plug-and-play. For physical hardware, only things to do are plugging the hardware into electricity and connect it to the internet using a wired network. Creation of the personalized bootstrap node only requires a generated identity of the hardware owner, making that person/organisation a Cloud Infrastructure provider, also called farmer.

### 9.2 Eligible Hardware

Below, eligible hardware is listed, as it can also be found in the Threefold manual on the following link: [https://manual.threefold.io/documentation/farmers/3node\\_building/3\\_set\\_hardware.html?highlight=eligible%20hardware#3node-requirements-summary](https://manual.threefold.io/documentation/farmers/3node_building/3_set_hardware.html?highlight=eligible%20hardware#3node-requirements-summary).

You need a theoretical minimum of 500 GB of SSD and 2 GB of RAM on a mini pc, desktop or server. In short, for peak optimization, aim for 100 GB of SSD and 8GB of RAM per thread (thread is equivalent to virtual core or logical core).

3Node **optimal** farming hardware ratio is 100 GB of SSD + 8 GB of RAM per Virtual Core.

Note that you can run Zero-OS on a Virtual Machine (VM), but you won't farm any TFT from this process. To farm TFT, Zero-OS needs to be on bare metal.

Also, note that ThreeFold runs its own OS, which is Zero-OS. You thus need to start with completely **wiped disks**. It is impossible to farm TFT with Windows, Linux or MAC OS installed on your disks. If you need to use such OS temporarily, boot it in Try mode with a removable media (USB key).

Note: Once you have the necessary hardware, you need to [create a farm](#), [create a Zero-OS bootstrap image](#), [wipe your disks](#) and [set the BIOS/UEFI](#). Then you can [boot your 3Node](#). If you are planning in building a farm in data center, [read this section](#).

Any computer with the following specifications can be used as a DIY 3Node.

- Any 64-bit hardware with an Intel or AMD processor chip.
- Servers, desktops and minicomputers type hardware are compatible.
- A minimum of 500 GB of SSD and a bare minimum of 2 GB of RAM is required.
- A ratio of 100GB of SSD and 8GB of RAM per thread is recommended.

- A wired ethernet connection is highly recommended to maximize reliability and the ability to farm TFT.
- A [passmark](#) of 1000 per core is recommended.

### **Bandwidth requirements**

A 3Node connects to the ThreeFold Grid and transfers information, whether it is in the form of compute, storage or network units (CU, SU, NU respectively). The more resources your 3Nodes offer to the Grid, the more bandwidth will be needed to transfer the additional information. In this section, we cover general guidelines to make sure you have enough bandwidth on the ThreeFold Grid when utilization will be happening.

**The strict minimum for one node is 1 mbps of bandwidth.**

If you want to expand your farm, you should check the following to make sure your bandwidth will be sufficient in case of there will be Grid utilization.

### **Bandwidth per Node Equation**

min Bandwidth per 3Node (mbps) =  $10 * \max((\text{Total SSD TB} / 1 \text{ TB}), (\text{Total Threads} / 8 \text{ Threads}), (\text{Total GB} / 64 \text{ GB})) + 10 * (\text{Total HDD TB} / 2)$

This equation means that for each TB of HDD you need 5 mbps of bandwidth, and for each TB of SSD, 8 Threads and 64GB of RAM (whichever is higher), you need 10 mbps of bandwidth.

This means a proper bandwidth for a node would be 10 mbps. As stated, 1 mbps is the strict minimum for one node.

### **Using Onboard Storage (3Node Servers)**

If your 3Node is based on a server, you can either use PCIe slots and PCIe-NVME adapter to install SSD NVME disk, or you can use the onboard storage.

Usually, servers use RAID technology for onboard storage. RAID is a technology that has brought resilience and security to the IT industry. But it has some limitations that ThreeFold did not want to get stuck with. ThreeFold developed a different and more efficient way to [store data reliably](#). This Quantum Safe Storage overcomes some of the shortfalls of RAID and can work over multiple nodes geographically spread on the TF Grid. This means that there is no RAID controller in between data storage and the TF Grid.

For your 3Nodes, you want to bypass RAID for Zero-OS to have bare metals on the system.

To use onboard storage on a server without RAID, you can

1. [Re-flash](#) the RAID card
2. Turn on HBA/non-RAID mode
3. Install a different card.

For HP servers, you simply turn on the HBA mode (Host Bus Adapter).

For Dell servers, you can either cross, or [re-flash](#), the RAID controller with an “IT-mode-Firmware” (see this [video](#)) or get a DELL H310-controller (which has the non-RAID option). Otherwise, you can install a NVME SSD with a PCIe adaptor, and turn off the RAID controller.

Once the disks are wiped, you can shut down your Node and remove the Linux Bootstrap Image (USB key). Usually, there will be a message telling you when to do so.

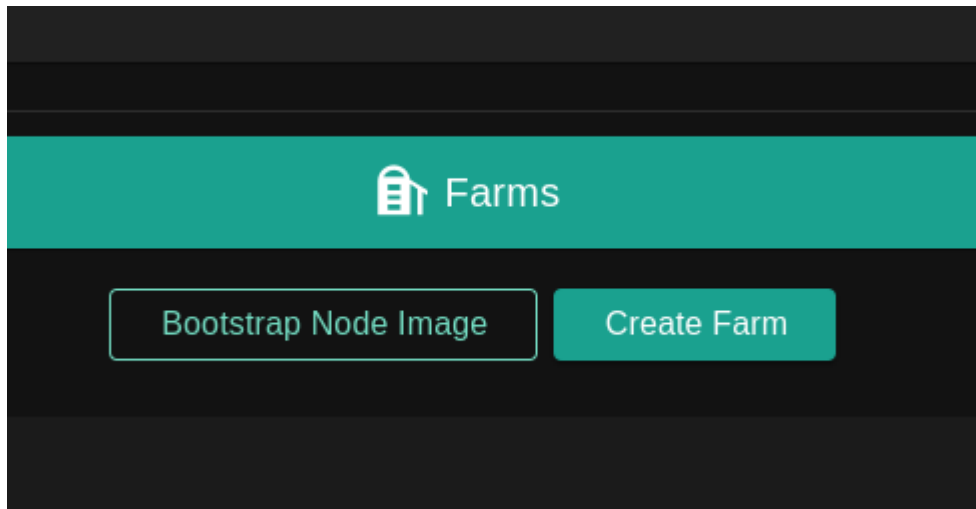
## **9.3 Bootstrap image**

We will now learn how to create a Zero-OS bootstrap image to boot a Node.

### 9.3.1 [Download the Zero-OS Bootstrap Image](#)

Let's download the Zero-OS bootstrap image.

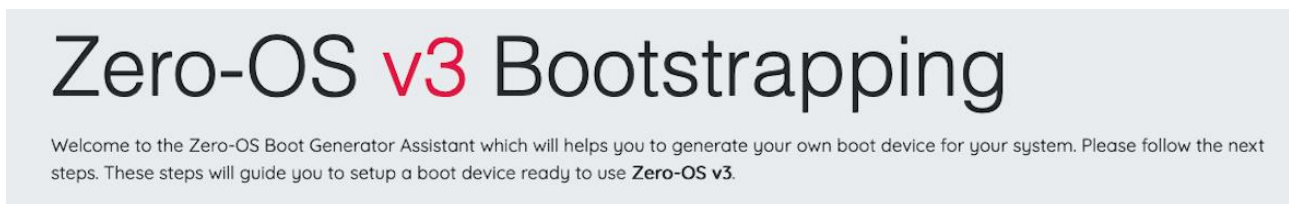
In the Farms section of the Dashboard, click on **Bootstrap Node Image**



*Figure 30 – Setting up a farm and a bootstrap node*

or use the direct link <https://v3.bootstrap.grid.tf>:

`https://v3.bootstrap.grid.tf`



*Figure 31 – Zero-OS Bootstrap*

This is the Zero-OS v3 Bootstrapping page.

## Who are you ?

In order to identify and link the operating system to your farm, you need to set your Farmer ID on the system.  
[You can find more information here.](#)

Farmer ID  Invalid

## Which release do you want ?

Zero-OS comes by default with 3 different **running modes**. Each mode has its own settings, own network and own purpose.  
Please choose the release you want.

Production	Testing	Development
<p><b>The real network</b></p> <p>Production releases are used for server ready to serve the real grid.</p> <p>Thank you to join this next release but keep in mind this is not fully stable yet.</p>	<p><b>Make it bulletproof</b></p> <p>Yaaay, you were waiting for this, it's now available for our grid version 3!</p> <p>The test network is the way to go if you want to enjoy features of our version 3 release without going too deep into</p>	<p><b>Crashtest</b></p> <p>Our current version 3 is the next awesome release of the grid. This version includes real decentralization and lots of new features, but it's still in an early stage.</p>

Figure 32 – Zero-OS Bootstrap set-up interface

Write your farm ID and choose production mode.

## Choose your image format

**EFI IMG** Generate an USB image with EFI (UEFI) bootable kernel. USB Image »

**This is the easiest way to go if your machine is EFI enabled. Any recent computer should support this out-of-box.**

To use this image, you'll need to erase and overwrite the complete USB Flash Drive.

This image is a 'dd' able image, it's made to be copy directly on your flash drive and not be used as a file.

.....

<https://v3.bootstrap.grid.tf/uefimg/prod/1>

Figure 33 – Zero-OS Bootstrap Image setup – Image format EFI

If your system is new, you might be able to run the bootstrap in UEFI mode.

---

<b>ISO</b>	Generate an ISO file you can burn into a CD-ROM. ..... <a href="https://v3.bootstrap.grid.tf/iso/prod/1">https://v3.bootstrap.grid.tf/iso/prod/1</a>	<a href="#">ISO Image »</a>
<b>USB</b>	Generate an USB image which can be copied into a USB Flash Drive directly.  This image won't work on recent computer, please use EFI USB instead.  Warning: to use this image, you'll need to erase and overwrite the complete USB Flash Drive.  This image is a 'dd' able image, it's made to be copy directly on your flash drive and not be used as a file. ..... <a href="https://v3.bootstrap.grid.tf/usb/prod/1">https://v3.bootstrap.grid.tf/usb/prod/1</a>	<a href="#">USB Image »</a>

Figure 34 – Zero-OS Bootstrap Image setup – Image formats ISO and USB

For older systems, run the bootstrap in BIOS mode. For BIOS CD/DVD, choose **ISO**. For BIOS USB, choose **USB**.

Download the bootstrap image. Next, we will burn the bootstrap image.

### 9.3.2 [Burn the Zero-OS Bootstrap Image](#)

The easiest way to burn a Zero-OS bootstrap image is on a USB key and boot the hardware from that key, but also other ways of creating and running a bootstrap image are accepted. More info can be found in the Threefold manual on

[https://manual.threefold.io/documentation/farmers/3node\\_building/2\\_bootstrap\\_image.html?highlight=USB#usb-key-biosuefi](https://manual.threefold.io/documentation/farmers/3node_building/2_bootstrap_image.html?highlight=USB#usb-key-biosuefi)

## 10. Annex E: Zero-OS federated operating system

Zero-OS is an operating system that serves as the foundational layer for a **federated data infrastructure**, eliminating interoperability issues between different independent actors.

The Operating system has a Linux kernel, making it fit for any Linux workload. Around the Linux kernel, all primitives have been built from scratch to make it a fully federated grid of interoperating nodes, with all features available to enable end-to-end security and privacy by design.

Zero-OS is the foundational OS on the capacity layer. It has been designed bottom up, starting from a Linux kernel and secure boot BIOS. It combines 3 primitive functions: **storage capacity**, **compute capacity**, and **intelligent network functions for running the network services**. The **distributed storage layer** allows storage of exabytes of information at a low cost while maintaining a high privacy and reliability level. Hardware capacity in any size and of multiple nature (CPU, GPU, memory, HDD, SSD, IPv4 and IPv6 addresses) hosted on secure bootable devices (any Intel/AMD HW) can be added to the grid of interconnected hardware nodes.

A very important consequence of this is that Zero-OS can run at the edge, on any location that has an internet connection, so it is the perfect installation for a medical infrastructure, remaining in full control of the hospital without imposing the burden to these institutions to have extensive in-depth technical knowledge. All measures around networking, security and privacy have been taken up into the architecture of the solution, by design.

Zero-OS has multiple security protection mechanisms built-in ([Zero-OS Protect](#)).

The operating system of the 3node (which is hardware with Zero-OS running on it) is made to exist in environments without the presence of technical knowhow. 3nodes are made to exist everywhere where network meet a power socket. The OS does not have a login shell and does not allow people to log in with physical access to a keyboard and screen nor does it allow logins over the network. There is no way the 3node accepts user-initiated login attempts.

3Nodes boot from a network facility. This means that they do not have local installed operating system files. Also, they do not have a local username / password file or database. Viruses and hackers have very little work with if there are no local files to plant viruses or trojan horses in. Another measure is that the boot facility provides hashes for the files sent to the booting 3node, so that the 3node can check whether it receives the intended file, no more man in the middle attacks.

The Zero-OS file system provides the same hash and file check mechanism. Every application file presented to a booting container has a hash describing it and the 3node on which the container is booting can verify if the received file matches the previously received hash.

Every deployment of one or more applications starts with the creation of a (private) [znet -](#) private overlay network. It is single tenant network, and it is not connected to the public internet. Every application or service that is started in a container in this overlay network is connection to all other containers via a point to point, encrypted network connection.

Detailed information about the current Threefold setup can be found in following resources:

- TFChain info on Polkadot portal :
  - Mainnet : <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Ftfchain.grid.tf#/extrinsics>
  - Testnet : <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Ftfchain.test.grid.tf#/extrinsics>
- GraphQL querying :
  - Mainnet : <https://graphql.grid.tf/graphql/>
  - Testnet : <https://graphql.test.grid.tf/graphql/>
- Portal to deploy infrastructure and applications :

- Mainnet : <https://dashboard.grid.tf/#/>
  - Testnet : <https://dashboard.test.grid.tf/#/>
- Open-source repository containing the Threefold open-source code :  
<https://github.com/threefoldtech>
- Interesting repo's :
  - Zero-OS, Linux based OS : <https://github.com/threefoldtech/zos>
  - TFChain, the blockchain component : <https://github.com/threefoldtech/tfchain>
  - Terraform Deployment scripts :  
<https://github.com/threefoldtech/terraform-provider-grid/tree/development/examples/resources>
  - Pulumi (= other IaC tool, better supporting API interface) deployment scripts :  
<https://github.com/threefoldtech/pulumi-threefold/tree/development/examples/go>
- Wiki :  
<https://manual.grid.tf>