

Layers, pipes and patterns: detailing the concept of data stations as a foundational building block for federated data systems in healthcare

Abstract We describe ...

1 A shift towards federated data systems as a design paradigm

The ambition for a seamlessly connected digital healthcare ecosystem, capable of leveraging vast quantities of patient data remains illusive. Designing and implementing health data platforms is notoriously difficult, given the heterogeneity and complexity of such systems. To address these issues, federated data systems have emerged as a design paradigm. This approach enables data to remain securely at its source, while allowing for distributed analysis and the generation of collective knowledge.

Recent technological inventions offer important new enablers to implement federated data systems, most notably:

- Capabilities of edge computing and single-node computing has increased significantly whereby it is now possible to process up to 1 TB of tabular data on a single node thereby enabling large volumes of data processing to be done efficiently in a decentralized fashion [1, 2];
- Federated analytics [3], and specifically federated learning, has matured as a means for training of predictive models, most notable through weights sharing of deep learning networks [4, 5];
- Privacy-enhancing technologies (PETs) such as secure multi-party computation (MPC) significantly improve secure processing across a network of participants and are now sufficiently mature to be used on an industrial scale [6, 7];
- The composable data stack provides a way to unbundle the venerable relational database into loosely components, thereby making it easier and more practical to implement FDS using cloud-based components with microservices, thereby opening up a transition path towards more modular and robust architectures for FDS [8, 9].

The architectural shift from centralized to federated data systems is not merely a technical evolution. Modern approaches to data governance are undergoing a similar paradigm shift towards federated solutions. As an example, the concept of a [data mesh](#) is increasingly being adopted at large corporations. From the perspective of sovereignty and solidarity, we believe that a commons-based, federated approach has distinct benefits in moving towards a more equitable, open digital infrastructure [10]. Federated data systems are inherently more aligned with contemporary data governance frameworks,

including the Data Governance Act (DGA), the European Health Data Space (EHDS) and the concept of data solidarity [11].

However, this ongoing paradigm shift towards is not without challenges. The notion of what constitutes a federated data system needs to be defined in more detail if we are to see the forest for the trees between different instantiations of the same concept. For example, ‘federation’ can mean any of the following solution patterns:

- Data federation addresses the problem of uniformly accessing multiple, possibly heterogeneous data sources, by mapping them into a unified schema, such as an RDF(S)/OWL ontology or a relational schema, and by supporting the execution of queries, like SPARQL or SQL queries, over that unified schema [12];
- Federation within the context of a Personal Health Train (PHT) refers to the concept by data processing is brought to the (personal health) data rather than the other way around, allowing (private) data accessed to be controlled, and to observe ethical and legal concerns [13–16], and is just one of many solutions designs that are collectively grouped as federated analytics [3];
- Federation as a mechanism for data sharing in a temporary staging environment within a network of research organizations in a Trusted Research Environment (TRE), with different types of federations services (localization, access);
- Federation services as defined in the DSSC Blueprint 2.0 pertain to the support the interplay of participants in a data space, operating in accordance to the policies and rules specified in the Rulebook by the data space authority.

What then, is a viable development path out of this creative chaos?

2 Data stations as a foundational building block

Inspired by previous calls to action to move towards open architectures for health data systems [17, 18] and the notion of the hourglass model [17, 19, 20], we hypothesize that the concept of a ‘data station’ can be used as a foundational building block for federated data systems. A data station should provide a set of minimal standards (at the waist of the hourglass), thereby maximizing the freedom to operate between data providers and data consumers within the context of a health data space. Note that this approach has many similarities with the FAIR Hourglass [20, 21]. Our approach of data stations presented here focuses on enabling on secondary data use of routine collected clinical data using the architecture of the Personal Health Train as a starting point [13–16]. The objective of this paper is to extend this architecture in order to address four design questions that are relevant in ongoing efforts to implement nation-wide federated systems.

2.1 Online transactional vs. analytical processing

First, it is well known data systems have different design and performance characteristics depending whether they are built for online transactional processing (OLTP) or online analytical processing (OLAP), as summarized in the Table 1 below (taken from Kleppmann M, Riccomini C [22]).

Table 1: Distinguishing characteristics between online transactional and analytical systems

Property	Transaction processing systems (OLTP)	Analytic systems (OLAP)
Main read pattern	Small number of records per query, fetched by key	Aggregate over large number of records
Main write pattern	Random-access, low-latency writes from user input	Bulk import (ETL) or event stream
Data modeling	Predefined	Defined post-hoc, either schema-on-read or schema-on-write
Primarily used by	End user/customer, via web application	Internal analyst, for decision support
What data represents	Latest state of data (current point in time)	History of events that happened over time
Dataset size	Gigabytes to terabytes	Terabytes to petabytes

Current efforts to design and implement the EHDS in fact aims to support primary (i.e. OLTP) and secondary use (OLAP) in one go [23, 24]. This Herculean endeavour has spawned many initiatives to develop a coherent architecture and support implementation across Europe that ultimately should lead to interoperability in the broadest sense of the word, most notably:

- The Data Space Blueprint v2.0 (DSB2) by the Data Spaces Support Centre ([25]) that serves as a vital guide for organizations building and participating in data spaces.
- The Simpl Programme ([26]) that aims to develop an open source, secure middleware that supports data access and interoperability in European data initiatives. It provides multiple compatible components, free to use, that adhere to a common standard of data quality and data sharing.
- TEHDAS2 ([27]), a joint action that prepares the ground for the harmonised implementation of the secondary use of health data in the EHDS.

We believe, however, that in order to successfully design and implement health data spaces, more detailed analysis and solution patterns are required that distinguish between primary (OLTP) and secondary (OLAP) data use. Although functional components can be shared between these two, it is a matter of the devil being in the details. Hence one of the objectives of this paper is to detail an open, technology agnostic architecture for secondary use, to complement existing efforts and guide the development in the field.

2.2 Centralized vs decentralized processing

A second design question pertains to the choice of single-node (centralized) or distributed (decentralized) platforms, which are not only be driven by technical considerations

(scalability, elasticity, fault tolerance, latency) but are also strongly dependent on organizational, legal or regulatory requirements such as data residency. The general approach of EHDS and other data spaces is federative by nature, that is, decentralized. For example, DSB2 stresses the need for interoperability and federative protocols within and across data spaces.

Upon closer inspection, however, specific functional components that are foreseen within the EHDS are best characterized as centralized (sub-)systems. As an example, consider the secure processing environments (SPE) as defined in [recital 77 of the EHDS](#). Known examples of such SPEs include data platforms provided by national statistics offices (CBS Microdata environment), healthcare-specific national platforms (Finland's Kapseli platform) and Trusted Research Environments (TREs) within the domain of research (see [EOSC-ENTRUST](#) for examples across Europe). Given that healthcare data is often vertically partitioned (data elements of the same subject are scattered across various data holders), SPEs provide the most effective means to (temporarily) share, integrate and analyse such data. Hence many SPEs are best described as centralized systems, and thus we need to take into account that data spaces constitute a hybrid architecture that includes both centralized and decentralized components. Thus a more detailed analysis is required to arrive at scalable solution patterns that combine centralized vs. decentralized processing in a larger federated health data system.

2.3 Solution patterns to populate data stations

A third design issue pertains to the mechanisms through which the data stations are to be populated with data by the data holder. In essence the industry is converging towards solution patterns from data engineering and data warehousing. OpenHIE specification, for example, makes a distinction of a Shared Health Record as a OLTP system for primary health data use vis-a-vis FHIR data pipelines that populate data stations for analytical, secondary use. [TO DO: improve wording, add references]

2.4 Federation as system of systems

The fourth design challenge pertains to the need for having a 'system of systems' [28] in the federated health data system at large. In real-world setting, secondary health data sharing will need to take into account the limited resources and expertise of smaller health care providers. In fact, the EHDS explicitly addresses this issue in [article 50](#) that exempts micro enterprises from the obligation to directly participate in the EHDS, but that each member state may opt to form so-called data intermediaries to act as a go-between ([recital 59](#)). Along a different dimension, we foresee a system of systems of various domain-specific federated networks that are loosely coupled, which poses design challenges in terms of autonomy (to what extent can each sub-network make independent choices), connectivity (how to connect sub-systems) and diversity.

2.5 Outline

In this paper we loosely follow an Action Design Research approach [29, 30] to address these design questions. Our main contributions are:

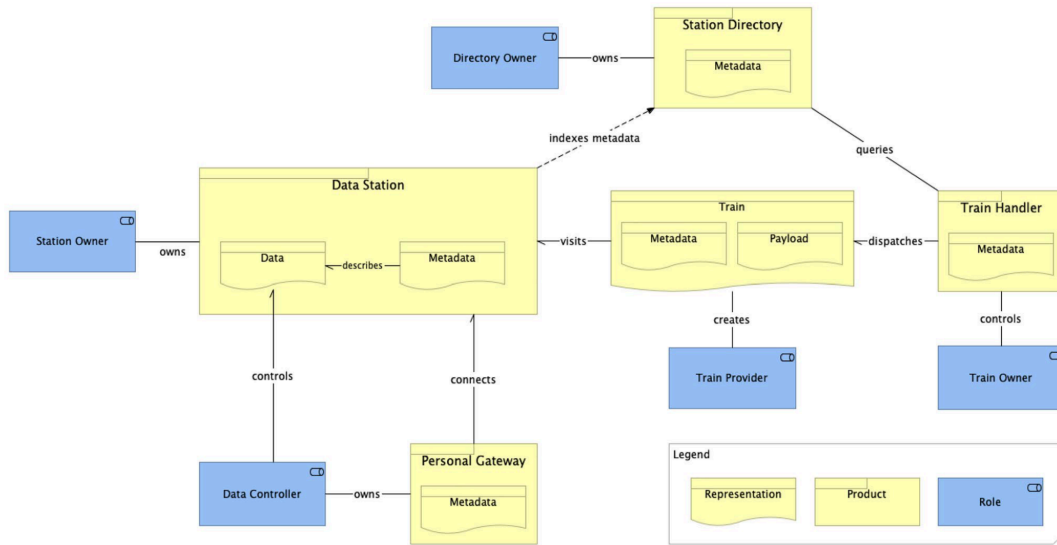
- A harmonized ontology of a data station and data hub, that integrates the PHT architecture [15] and the DSSC Blueprint 2.0
- Comparative analysis of existing implementations
- Synthesis of the above into functional and technical description of a data station in Archimate, thereby focusing on two primary patterns [31]:
 - the layers pattern for addressing various aspects of interoperability across the stack, extending earlier work by Welten S, Arruda Botelho Herr M de, Hempel L, et al [32] who have introduced five layers of interoperability;
 - the pipes and filters pattern for addressing various solution designs for the extract-transform-load (ETL) mechanisms through which data stations are populated, following the current common practice of datalakes and lakehouse solution designs [33–37];
- A reference implementation of data stations using the trains based on containers as a generic infrastructure for federated learning and federated analysis.

3 Towards a unified ontology of data stations and data hubs

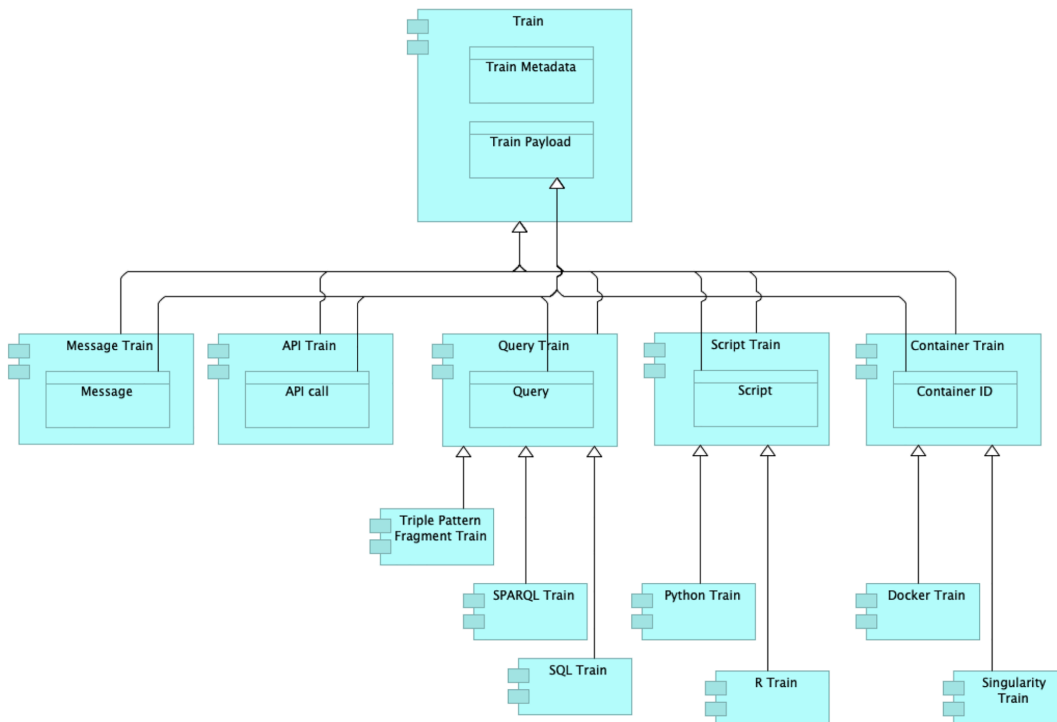
3.1 The Personal Health Train architecture

We take the Personal Health Train (PHT) architecture as our starting point [15], wherein a Data Station is defined as a software application primarily responsible for making data and its associated metadata available to users under conditions determined by applicable regulations and Data Controllers. The main concepts are shown in Figure 1a, while the various types of trains are shown in Figure 1b. A glossary of the concept is provided in Section 8.

Figure 1: High-level overview of the Personal Health Train (PHT) architecture
(a) Main roles and components



(b) Train types



In their paper Silva Santos LO Bonino da, Ferreira Pires L, Martinez V, Moreira J, Guizzardi R [15] continue to describe more details of the PHT architecture including i) the various functions, services, interface and internal components of the data station; ii) the data visiting process; and iii) the data staging concept in the case data access has been authorized, but the station is not capable of executing the train and needs to stage a

capable station with enough resources to run the train. We will consider these details later.

As an aside, it is good to mention that the authors of the PHT architecture have initiated the development of two specifications after publication of this paper, namely:

- [FAIR Data Point](#) specification, which covers only the metadata and catalog part of the PHT architecture;
- the [FAIR Data Train](#) specification, which covers the full scope of the original paper but at the time of writing is still incomplete.

3.2 Mapping PHT to the DSSC Blueprint 2.0

To arrive at consistent conceptualization of data stations and trains, Table 2 maps the PHT architecture to the DSSC Blueprint 2.0 (DSB2). Some mappings are relatively evident. For example, the concept of Data and Metadata as defined in PHT is subsumed in the concept of a Data Product in DSB2. Less evident, is the mapping of the notion of a Train that ‘... represents the way data consumers interact with the data available in the Data Stations. Trains represent a particular data access request and, therefore, each train carries information about who is responsible for the request, the required data, what will be done with the data, what it expects from the station, etc.’ to Value Creation Services in DSB2 that includes data fusion and enrichment, collaborative data analytics and federated learning. We tentatively conclude that it is possible to have a consistent conceptual mapping between, at least at the high level, of the PHT architecture into DSB2. We will return to this matter, when more detailed functions and technical standards are considered for the Archimate specification.

Table 2: Mapping the key concepts from the PHT architecture [15] to the [concepts of the DSSC Blueprint 2.0](#).

Component PHT	mapping to DSSC Blueprint 2.0 concepts
<ul style="list-style-type: none"> • Data Station 	<ul style="list-style-type: none"> • Data Space Building Block
<ul style="list-style-type: none"> • Data • Metadata 	<ul style="list-style-type: none"> • Data Product
<ul style="list-style-type: none"> • Data Controller 	<ul style="list-style-type: none"> • Data Rights Holder
<ul style="list-style-type: none"> • Station Controller 	<ul style="list-style-type: none"> • Data Product Provider
<ul style="list-style-type: none"> • Personal Gateway 	<ul style="list-style-type: none"> • included in Participant Agent Services
<ul style="list-style-type: none"> • Station Directory 	<ul style="list-style-type: none"> • included in Federation Services • Catalogue provisions and discovers offerings of data and services in a data space
<ul style="list-style-type: none"> • Directory Owner 	<ul style="list-style-type: none"> • Common Intermediary provides federation services that are common to all participants of the data space
<ul style="list-style-type: none"> • Train 	<ul style="list-style-type: none"> • Value Creation Services
<ul style="list-style-type: none"> • Train Provider 	<ul style="list-style-type: none"> • Service Provider
<ul style="list-style-type: none"> • Train Handler 	<ul style="list-style-type: none"> • specialization of Data Space Component that realizes the Train Value Creation Service
<ul style="list-style-type: none"> • Train Owner 	<ul style="list-style-type: none"> • included in Service Provider as most generic role • concept of Intermediary (specialization of SP) is closer to definition of Train Owner

3.3 The lakehouse architecture as the *de facto* standard for populating data stations

The PHT architecture does not specify *how* the data stations should be populated with data. Also the DSB2 only describes how the ‘Data, Services and Offerings descriptions’ building block should provide data providers the tools to describe a data product appropriately and completely, that is, tools for metadata creation and management.

One of the key questions of this paper is to detail [the ‘data conformity zone’ as defined in the Cumuluz canvas](#) as the functionality through which the data station is populated

4 Comparative analyses of existing data stations

5 Synthesis of solution patterns patterns for data stations

We take Klepmann (2017) as our starting point, who states that “Many applications today are *data-intensive*, as opposed to *compute-intensive*. Raw CPU power is rarely a limiting factor for these applications—bigger problems are usually the amount of data, the complexity of data, and the speed at which it is changing.”

Generically, we want:

Reliability	Scalability	Maintainability
tolerating hardware & software vaults	Measuring load & performance	Operability, simplicity & evolvability
human error	Latency percentiles, throughput	

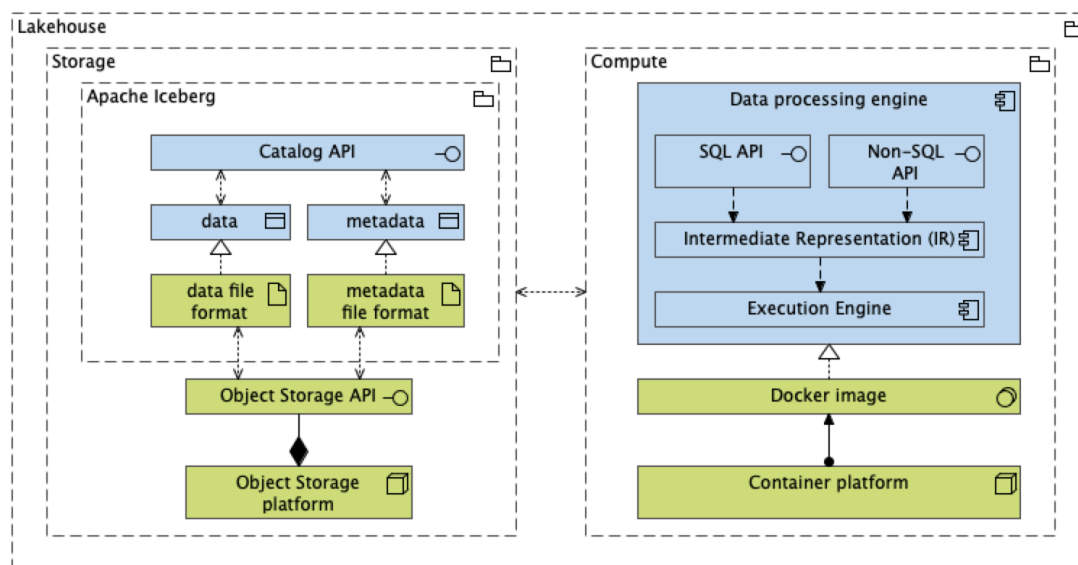
We focus on analytical data systems, with different patterns from transactional data systems.

5.1 Detailing the layers of a data station

TO DO: provide detailed layers, and explain how interoperability works across the layers:

- Storage layer (technology): all reference architectures stipulate use of S3-compliant blob storage
- Data and metadata (application): resides in the data station
- We propose to move towards open table formats, that is, Apache Iceberg, whereby storage and compute can be separated

Figure 2: Solution of a minimal lakehouse that sits at the core of a data station



5.2 Detailing the data conformity zone

TO DO: explain that

- data conformity zone is essentially a lakehouse pattern
- the architecture of a lakehouse has stabilized and converged towards:
 - **Colum-oriented storage and memory layout:** Apache Arrow ecosystem, including Apache Flight
 - **Late-binding with logical data models most suited for analytics:** ELT pattern with zonal architecture
 - *staging zone*: hard business rules (does incoming data comply to syntactic standard), change data capture
 - *linkage & conformity zone*: concept-oriented tables, typically following a data vault modeling principle, ascertain referential integrity across resources, with tables per concept and linking tables. Mapping to coding systems. Entity resolution for record linkage at the subject level
 - *consumption zone*: convenient standardized views like an event table (patient journey, layout for process mining) with uniformity of dimensions using a star schema

5.3 Detailing the trains

TO DO: explain

- difference between centralized and distributed federated learning (causes lots of confusion)
- basically Train is a generalization of all types of computes
- difference between
 - Train for secondary use, which usually with batch-wise, less strict latency requirements
 - Train for primary use, like API call and messaging, with stricter latency requirements. This also includes deployment of AI for inference

6 Reference implementation with current open source software (OSS) components

6.1 Data stations from the composable data stack

- Python and SQL(-like) languages as the *de facto* standard for analytical processing i.e. the most commonly used analytical scripting languages. Where necessary, using Intermediate Representations (IR), any analytical query can be transpiled to the target engine of choice
- Single-node compute capable of efficiently processing up to 1 TB of data within tens of seconds (polars, DuckDB), so we do away with distributed processing
- Open table formats (Iceberg, Hudi, Delta) and open file formats (parquet, AVRO)

6.2 Container-based trains as the most versatile solution

6.3 Bringing it all together for federated analytics & machine learning (FL)

- Local data stations are conceptualized as serverless lakehouses
 - Local ELT pipelines
 - Decentralized (pre-)processing, including quality control upon ingest
 - ...
- For horizontally partitioned data, we can apply FL techniques where only aggregated results are combined centrally
- For vertically partitioned data, we need an intermediate/temporary zone for linking the data
- For both horizontally and vertically partitioned data, we can choose to add PETs, most specifically MPC, as an extra security measure
 - Horizontally partitioned data: [one-shot FL](#)
 - Vertically partitioned data:
 - linkage in the blind
 - reversible pseudonimization
- standardized approach to mappings [38]

7 Parking lot

- Difference with data mesh: mesh of domains, federation is in the same domain. Underlying technology of a data station, however, is functionally identical
- UMCU: CQRS pattern for separately optimizing read/write patterns
- DSSC Blueprint: FL subsumed in value adding services

Table 3 lists known examples of existing health data platform architectures along these two trade-offs.

Table 3: Broad categorization of health data platforms

	primary		secondary
centralized	openHIE	[39], Digizorg, Nordics	kapseli, Mayo, ...
decentralized	RSO Zuid Limburg, Twiin portal, ...	many federated analytics research networks such as x-omics programme and EUCAIM	

8 Glossary

- *Data Controller* (Business Role) is the role of controlling rights over data.
- *Data Station* (Business Product) is a software application responsible for making data and their related metadata available to users under the accessibility conditions determined by applicable regulations and the related Data Controllers.

- *Directory Owner* (Business Role) is the role of being responsible for the operation of a Station Directory.
- *Personal Gateway* (Business Product) is a software application responsible for mediating the communication between Data Stations and Data Controllers. The Data Controllers are able to exercise their control over the data available in different Data Stations through the Personal Gateway.
- *Station Directory* (Business Product) is a software application responsible for indexing metadata from the reachable Data Stations, allowing users to search for data available in those stations.
- *Train* (Business Representation) represents the way data consumers interact with the data available in the Data Stations. Trains represent a particular data access request and, therefore, each train carries information about who is responsible for the request, the required data, what will be done with the data, what it expects from the station, etc.
- *Train Handler* (Business Representation) is a software application that interacts with the Stations Directory on behalf of a client to discover the availability and location of data and sends Trains to Data Stations.
- *Station Owner* (Business Role) is the role of being responsible for the operation of a Data Station.
- *Train Owner* (Business Role) is the role of using a Train Handler to send Trains to Data Stations.
- *Train Provider* (Business Role) is the role of being responsible for the creation of a specific Train, e.g. the developer of a specific analysis algorithm.

Bibliography

1. Raasveldt M, Mühleisen H (2019) DuckDB: An Embeddable Analytical Database. In: Proceedings of the 2019 International Conference on Management of Data. ACM, Amsterdam Netherlands, pp 1981–1984
2. Nahrstedt F, Karmouche M, Bargiel K, Banijamali P, Nalini Pradeep Kumar A, Malavolta I (2024) An Empirical Study on the Energy Usage and Performance of Pandas and Polars Data Analysis Python Libraries. In: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering. ACM, Salerno Italy, pp 58–68
3. Wang Z, Ji H, Zhu Y, Wang D, Han Z (2025) A Survey on Federated Analytics: Taxonomy, Enabling Techniques, Applications and Open Issues. IEEE Communications Surveys & Tutorials 1. <https://doi.org/10.1109/COMST.2025.3558755>
4. Rieke N, Hancox J, Li W, et al (2020) The Future of Digital Health with Federated Learning. npj Digital Medicine 3(1):1–7. <https://doi.org/10.1038/s41746-020-00323-1>
5. Teo ZL, Jin L, Liu N, et al (2024) Federated Machine Learning in Healthcare: A Systematic Review on Clinical Applications and Technical Architecture. Cell Reports Medicine 5(2):101419. <https://doi.org/10.1016/j.xcrm.2024.101419>
6. (2023) The PET Guide

7. (2023) From Privacy to Partnership
8. Pedreira P, Erling O, Karanasos K, et al (2023) The Composable Data Management System Manifesto. *Proceedings of the VLDB Endowment* 16(10):2679–2685. <https://doi.org/10.14778/3603581.3603604>
9. The Composable Codex. <https://voltrondata.com/codex.html>. Accessed 16 Oct 2024
10. Krewer J, Warso Z (2024) Digital Commons as Providers of Public Digital Infrastructures
11. Prainsack B, El-Sayed S (2023) Beyond Individual Rights: How Data Solidarity Gives People Meaningful Control over Data. *The American Journal of Bioethics* 23(11):36–39. <https://doi.org/10.1080/15265161.2023.2256267>
12. Gu Z, Corcoglioniti F, Lanti D, et al (2022) A Systematic Overview of Data Federation Systems. *Semantic Web* 15(1):107–165. <https://doi.org/10.3233/SW-223201>
13. Beyan O, Choudhury A, Soest J van, et al (2020) Distributed Analytics on Sensitive Medical Data: The Personal Health Train. *Data Intelligence* 2(1–2):96–107. https://doi.org/10.1162/dint_a_00032
14. Choudhury A, Soest J van, Nayak S, Dekker A (2020) Personal Health Train on FHIR: A Privacy Preserving Federated Approach for Analyzing FAIR Data in Healthcare. In: Bhattacharjee A, Borgohain SK, Soni B, Verma G, Gao X-Z (eds) *Machine Learning, Image Processing, Network Security and Data Sciences*. Springer, Singapore, pp 85–95
15. Silva Santos LO Bonino da, Ferreira Pires L, Martinez V, Moreira J, Guizzardi R (2022) Personal Health Train Architecture with Dynamic Cloud Staging. *SN Computer Science* 4. <https://doi.org/10.1007/s42979-022-01422-4>
16. Zhang C, Choudhury A, Volmer L, et al (2023) Secure and Private Healthcare Analytics: A Feasibility Study of Federated Deep Learning with Personal Health Train. <https://www.researchsquare.com/article/rs-3158418/v1>. Accessed 24 Oct 2024
17. Estrin D, Sim I (2010) Health Care Delivery. *Open mHealth Architecture: An Engine for Health Care Innovation*. *Science* (New York, NY) 330(6005):759–760. <https://doi.org/10.1126/science.1196187>
18. Mehl GL, Seneviratne MG, Berg ML, et al (2023) A Full-STAC Remedy for Global Digital Health Transformation: Open Standards, Technologies, Architectures and Content. *Oxford Open Digital Health* 1:oqad18. <https://doi.org/10.1093/oodh/oqad018>
19. Beck M (2019) On the Hourglass Model. *Communications of the ACM* 62(7):48–57. <https://doi.org/10.1145/3274770>
20. Schultes E (2023) The FAIR Hourglass: A Framework for FAIR Implementation. *FAIR Connect* 1(1):13–17. <https://doi.org/10.3233/FC-221514>

21. Silva Santos LO Bonino da, Guizzardi G, Prince Sales T (2022) FAIR Digital Object Framework. <https://fairdigitalobjectframework.org/>. Accessed 19 Jun 2025
22. Kleppmann M, Riccomini C (2026) Designing Data-Intensive Applications, 2nd Edition (Early Release). O'Reilly
23. (2025) European Health Data Space Regulation (EHDS). https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space-regulation-ehds_en. Accessed 9 Jun 2025
24. Cascini F, Pantovic A, Al-Ajlouni YA, Puleo V, De Maio L, Ricciardi W (2024) Health Data Sharing Attitudes towards Primary and Secondary Use of Data: A Systematic Review. *eClinicalMedicine* 71:102551. <https://doi.org/10.1016/j.eclinm.2024.102551>
25. Data Spaces Blueprint v2.0. <https://dssc.eu/space/BVE2/1071251457/Data+Spaces+Blueprint+v2.0+-+Home>. Accessed 9 Jun 2025
26. Simpl Programme. <https://simpl-programme.ec.europa.eu/>. Accessed 9 Jun 2025
27. TEHDAS2. <https://tehdas.eu/>. Accessed 9 Jun 2025
28. Gorod A, Sauser B, Boardman J (2008) Paradox: Holarchical View of System of Systems Engineering Management. In: 2008 IEEE International Conference on System of Systems Engineering. pp 1–6
29. Sein M, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action Design Research. *MIS Quarterly* 35:37–56. <https://doi.org/10.2307/23043488>
30. Venable J, Pries-Heje J, Baskerville R (2016) FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems* 25(1):77–89. <https://doi.org/10.1057/ejis.2014.36>
31. Buschmann F, Meunier R, Rohnert H, Sommerlad P, Stal M (1996) A System of Patterns. Wiley
32. Welten S, Arruda Botelho Herr M de, Hempel L, et al (2024) A Study on Interoperability between Two Personal Health Train Infrastructures in Leukodystrophy Data Analysis. *Scientific Data* 11(1):663. <https://doi.org/10.1038/s41597-024-03450-6>
33. Harby AA, Zulkernine F (2025) Data Lakehouse: A Survey and Experimental Study. *Information Systems* 127:102460. <https://doi.org/10.1016/j.is.2024.102460>
34. Schneider J, Gröger C, Lutsch A, Schwarz H, Mitschang B (2024) The Lakehouse: State of the Art on Concepts and Technologies. *SN Computer Science* 5(5):449. <https://doi.org/10.1007/s42979-024-02737-0>
35. Hai R, Koutras C, Quix C, Jarke M (2023) Data Lakes: A Survey of Functions and Systems. *IEEE Transactions on Knowledge and Data Engineering* 35(12):12571–12590. <https://doi.org/10.1109/TKDE.2023.3270101>
36. AbouZaid A, Barclay PJ, Chrysoulas C, Pitropakis N (2025) Building a Modern Data Platform Based on the Data Lakehouse Architecture and Cloud-Native Ecosystem.

Discover Applied Sciences 7(3):166. <https://doi.org/10.1007/s42452-025-06545-w>

37. Mazumdar D, Hughes J, Onofre JB (2023) The Data Lakehouse: Data Warehousing and More. <http://arxiv.org/abs/2310.08697>. Accessed 11 Jan 2024
38. Zhang S, Cornet R, Benis N (2024) Cross-Standard Health Data Harmonization Using Semantics of Data Elements. Scientific Data 11(1):1407. <https://doi.org/10.1038/s41597-024-04168-1>
39. (2024) OpenHIE Framework v5.2-En. <https://ohie.org/>. Accessed 27 Aug 2024