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

Daniel Kapitan
Health-RI, Dutch Hospital Data, Eindhoven University of Technology
daniel@kapitan.net

Jack Broeren
Health-RI

Jeroen Beliën
Health-RI, Amsterdam UMC

Niels Bolding
Health-RI

Stefan van der Loop
Cumuluz

Yannick Vinkesteijn
Dutch Hospital Data
y.vinkesteijn@dhd.nl

Joep de Ligt
Hartwig Medical Foundation
j.deligt@hartwigmedicalfoundation.nl

Abstract We describe …

1 Designing and implementing health data platforms is notoriously difficult

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 heterogeneity and complexity of such systems. As a starting point - and to frame the scope of this paper - consider the trade-offs along the following two design criteria.

First, we distinguish health data platform for primary or secondary health availability [1]. It is well known data systems have different design an 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 M. Kleppmann and C. Riccomini [2]).

Table 1: Distinguishing characteristics between 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

A second design criterium 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.

To complicate matters further, current efforts to design and implement the European Health Data Space (EHDS) in fact aims to support primary and secondary use in one go [3]. 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 Simpl Programme ([4]) 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 ([5]), a joint action that prepares the ground for the harmonised implementation of the secondary use of health data in the EHDS.
- The Data Space Blueprint v2.0 by the Data Spaces Support Centre ([6]) that serves as a vital guide for organizations building and participating in data spaces.

To illustrate the current state of affairs, Table 2 lists known examples of existing health data platform architectures along these two trade-offs.

Table 2: Broad categorization of health data platforms

	primary	secondary
centralized	openHIE [7], Digizorg, Nordics	kapseli, Mayo, …
decentralized	RSO Zuid Limburg, Twiin portaal, …	many federated analytics research networks such as x-omics programme and EUCAIM

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

2 A confluence of developments towards federated data systems

We observe a confluence of various trends into what we call federated data systems (FDS). The need for FDS is exemplified by the increasing number of research consortia that are using FDS in a wide diversity of domains. There are, however, a number of issues that need to be addressed to bring FDS to the highest technological readiness level (TRL) that is required for large-scale operations at the level of a country and even in a networks-of-networks as is required for the EHDS.

One of those issues pertains to the concept of 'data stations', which is the focus of this practice paper. Data stations are conceived as the foundational building block with which FDS are constructed, by and large inspired as how the Internet came into existence. As of today, different architectural patterns and solution designs are emerging, as for example:

- Fair Data Points (FDP) as data stations containing metadata and indexes, aimed at supporting localization of data [insert references]
- Data stations in the sense of centralized federated learning networks, which is currently one of the most widely used solution designs for federated learning [8], [9]
- Graph-based data stations such as the Fair Data Cube [10] and the Swiss Personalized Health Network [11]
- Specifically in the Dutch context:
  - the definition of a data station for primary health data availibility by Cumuluz [add reference]
  - the definition of data stations by the Zorginstituut [12]

At the same time, the computer science and engineering has developed couple of new concepts and technologies that make it easier to implement FDS:

- **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 on a single data station [13], [14]
- The **composable data stack** provides a way to unbundle the venerable data base into loosely components, thereby making it easier and more practical to implement FDS,

thereby opening up a transition path towards more modular and robust architectures [15], [16]

- Federated machine learning (or federated learning in short) has matured as a means for training of predictive models, most notable through weights sharing of deep learning networks [9]
- The increasing need for privacy-enhancing technologies (PETs) additionally fuels the development of FDS-related technologies, where technologies such as secure multiparty computation (MPC) are now sufficiently mature to be used on an industrial scale [17], [18]

As noted by Perreira et al. (2023), however:

> The requirement for specialization in data management systems has evolved faster than our software development practices. After decades of organic growth, this situation has created a siloed landscape composed of hundreds of products developed and maintained as monoliths, with limited reuse between systems. This fragmentation has resulted in developers often reinventing the wheel, increased maintenance costs, and slowed down innovation. It has also affected the end users, who are often required to learn the idiosyncrasies of dozens of incompatible SQL and non-SQL API dialects, and settle for systems with incomplete functionality and inconsistent semantics.

This paper rises to their call to take a principled, open source approach to FDS aimed at "…standardizing different aspects of the data stack (…) advocating for a paradigm shift in how data management systems are designed", focusing on FDS for secondary use in healthcare, that is, federated analytics and federated learning.

1. functional architecture and specification of FDS for healthcare that integrates various common practices and blueprints from the data engineering community; we explicitly aim to clarify the different perspectives, namely:

- requirements from primary vs secondary use, critically evaluating whether it is possible to support both with one solution design for a data station
- address various aspects of interoperability by explicitly detailing the architecture in using the layers pattern [19]
- address various solution designs for the extract-transform-load (ETL) mechanisms using the pipes and filters pattern [19]
- address various solution designs for federated learning for horizontal vs. vertically partitioned data and solution involving secure processing environments (SPEs) as defined in the EHDS

2. a reference implementation in the Python-Rust data stack that is quickly emerging as a new *de facto* standard for performant and reliable analytical processing;

3 Desiderata of federated data systems

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.

3.1 Design principles of analytical data systems

These functional requirements lead us to the following design principles

- **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

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

At the lower, technical, levels we follow the rationale of 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)

3.3 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 partioned 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 mapppings [20]

4 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

Bibliography

[1] F. Cascini, A. Pantovic, Y. A. Al-Ajlouni, V. Puleo, L. De Maio, and W. Ricciardi, "Health Data Sharing Attitudes towards Primary and Secondary Use of Data: A Systematic Review," *eClinicalMedicine*, vol. 71, p. 102551, May 2024, doi: 10.1016/j.eclinm.2024.102551.

[2] M. Kleppmann and C. Riccomini, *Designing Data-Intensive Applications, 2nd Edtion (Early Release)*. O'Reilly, 2026.

[3] "European Health Data Space Regulation (EHDS)." Accessed: Jun. 09, 2025. [Online]. Available: https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space-regulation-ehds_en

[4] "Simpl Programme." Accessed: Jun. 09, 2025. [Online]. Available: https://simpl-programme.ec.europa.eu/

[5] "TEHDAS2." Accessed: Jun. 09, 2025. [Online]. Available: https://tehdas.eu/

[6] "Data Spaces Blueprint v2.0." Accessed: Jun. 09, 2025. [Online]. Available: https://dssc.eu/space/BVE2/1071251457/Data+Spaces+Blueprint+v2.0+-+Home

[7] "OpenHIE Framework v5.2-En." Accessed: Aug. 27, 2024. [Online]. Available: https://ohie.org/

[8] D. Kapitan, F. Heddema, A. Dekker, M. Sieswerda, B.-J. Verhoeff, and M. Berg, "Data Interoperability in Context: The Importance of Open-Source Implementations

When Choosing Open Standards," *Journal of Medical Internet Research*, vol. 27, no. 1, p. e66616, Apr. 2025, doi: 10.2196/66616.

[9] N. Rieke *et al.*, "The Future of Digital Health with Federated Learning," *npj Digital Medicine*, vol. 3, no. 1, pp. 1–7, Sep. 2020, doi: 10.1038/s41746-020-00323-1.

[10] X. Liao *et al.*, "FAIR Data Cube, a FAIR Data Infrastructure for Integrated Multi-Omics Data Analysis," *Journal of Biomedical Semantics*, vol. 15, no. 1, p. 20, Dec. 2024, doi: 10.1186/s13326-024-00321-2.

[11] "SPHN - Swiss Personalized Health Network (SPHN)." Accessed: Jun. 09, 2025. [Online]. Available: https://sphn.ch/

[12] "KIK-V x GERDA," Apr. 2024. [Online]. Available: https://populationhealthdata.nl/wp-content/uploads/2024/07/Whitepaper-GERDA-x-KIK-V_-databeschikbaarheid-door-hergebruik.pdf

[13] M. Raasveldt and H. Mühleisen, "DuckDB: An Embeddable Analytical Database," in *Proceedings of the 2019 International Conference on Management of Data*, Amsterdam Netherlands: ACM, Jun. 2019, pp. 1981–1984. doi: 10.1145/3299869.3320212.

[14] F. Nahrstedt, M. Karmouche, K. Bargieł, P. Banijamali, A. Nalini Pradeep Kumar, and I. Malavolta, "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*, Salerno Italy: ACM, Jun. 2024, pp. 58–68. doi: 10.1145/3661167.3661203.

[15] P. Pedreira *et al.*, "The Composable Data Management System Manifesto," *Proceedings of the VLDB Endowment*, vol. 16, no. 10, pp. 2679–2685, Jun. 2023, doi: 10.14778/3603581.3603604.

[16] "The Composable Codex." Accessed: Oct. 16, 2024. [Online]. Available: https://voltrondata.com/codex.html

[17] "The PET Guide," 2023. Accessed: Jan. 22, 2025. [Online]. Available: https://unstats.un.org/bigdata/task-teams/privacy/guide/

[18] "From Privacy to Partnership," Jan. 2023.

[19] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *A System of Patterns*, vol. 1. in Pattern-Oriented Software Architecture, vol. 1. Wiley, 1996.

[20] S. Zhang, R. Cornet, and N. Benis, "Cross-Standard Health Data Harmonization Using Semantics of Data Elements," *Scientific Data*, vol. 11, no. 1, p. 1407, Dec. 2024, doi: 10.1038/s41597-024-04168-1.