# Towards a unified ontology of data stations and federated analytics hubs

## 1 The Personal Health Train architecture as a starting point

We take the Personal Health Train (PHT) architecture as our starting point [1]. Note that the architecture described here is based on the vantage6 platform, which we aim to generalize in this paper.

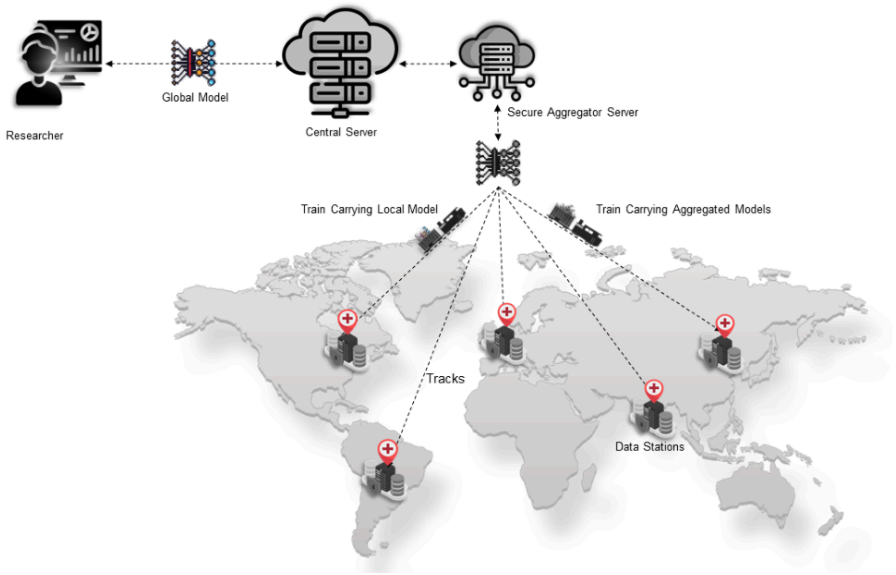



Figure 1: Overall architecture of a federated deep learning architecture adapted from Vantage6. The figure depicts a researcher connected to the central server, a secure aggregation server, trains carrying models, connected data stations, and the communicating tracks. Image source Choudhury A, Volmer L, Martin F, et al [1].

Figure 1 depicts the key components of the architecture, which are described in the infobox below. It is important to note that we will focus on a hub-and-spoke network topology (also referred to as client-server) for the federated analytics architecture. While there are many different network topologies choice for implementing federated analytics, including peer-to-peer and tree-based hierarchical, we focus on hub-and-spoke primarily because of a better fit for the healthcare domain in terms of security and privacy requirements. While peer-to-peer architecture is more cost-effective and offers a high capacity, it has the disadvantages of a lack of security and privacy constraints and a complex troubleshooting process in the event of a failure. By choosing the client-server network topology, we assume that Data Consumers gain access and use the platform through a central server, which also functions as the main point of control for the federated analytics network.

## 2 Central coordination server

Located at the highest hierarchical level and serves as an intermediary for message exchange among all other components. The components of the system, including the users, data stations, and Secure Aggregation Server (SAS), are registered entities that possess well-defined authentication mechanisms within the central server. It is noteworthy that the central acts as a coordinator rather than a computational engine. Its primary function is to store task-specific metadata relevant to the task initiated for training the deep learning algorithm. the central coordination server is equivalent to the hub in the event broker topology pattern.

## 3 Secure Aggregation Server (SAS)

A specialized station that contains no data and functions as a consolidator of locally trained models. The aggregator node is specifically designed to possess a Representational State Transfer (REST)–application programming interface (API) termed as the API Forwarder. The API Forwarder is responsible for managing the requests received from the data stations and subsequently routing them to the corresponding active Docker container, running the aggregation algorithm. Note that the SAS fulfills a distinctly different function than the central coordination server, although both component are often run on the same physical infrastructure at the central organization which oversees the federated network. The SAS is equivalent to the server-side in the client-server federated analytics pattern [2].

## 4 Data Stations

Devices located within the confines of each data holder's jurisdiction that are not reachable or accessible from external sources other than the federated learning network. The data stations communicate with the central server through a pull mechanism. Furthermore, the data stations not only serve as hosts for the infrastructure node but also offer the essential computational resources required for training the deep learning network. The infrastructure node is the software component installed in the data stations that orchestrates the local execution of the model and its communication with the central server and the SAS. The primary function of the data station is to receive instructions from both the SAS and the central server, perform the computations needed for training the CNN algorithm, and subsequently transmit the model weights back to the respective sources.

## 5 Trains

A Docker image that encompasses several components bundled together: the machine learning model that needs to be trained on local data; the aggregation algorithm used for consolidating the models; and a secondary Python Flask API known as the Algorithm API for facilitating the communication of these models.

## 6 Tracks and Track Provider

Refers to the various infrastructure components establish coordination among themselves through the use of secure communication channels commonly referred to as the “tracks.” The communication channels are enabled with end-to-end encryption. The responsibility for the maintenance of the infrastructure, including the hosting of the central coordinating server and the specialized SAS, lies with the track provider. The track provider is additionally accountable for the maintenance of the “tracks” and aids the data providers in establishing the local segment of the infrastructure known as the “nodes.”

## 7 Data Providers

Hospitals and health care organizations that are responsible for curating the pertinent datasets used for training the deep learning network. The responsibility of hosting the data stations within their respective local jurisdiction lies with the data provider. They exercise authority over the data as well as the infrastructure component called the node.

##Researcher Responsible for activating the deep learning algorithm and engaging in the authentication process with the central coordinating server using a registered username and password. This allows the researcher to establish their identity and gain secure access to the system, with their communication safeguarded through end-to-end encryption. The researcher can then assign tasks to individual nodes, monitor progress, and terminate tasks in the event of failure. Importantly, the researcher’s methodology is designed to keep the intermediate outcomes of the iterative deep learning training process inaccessible, ensuring that the ultimate global model can only be obtained upon completion of all training iterations, thereby mitigating the risk of unauthorized access by malicious researchers to the intermediate models and providing a security mechanism against insider attacks.

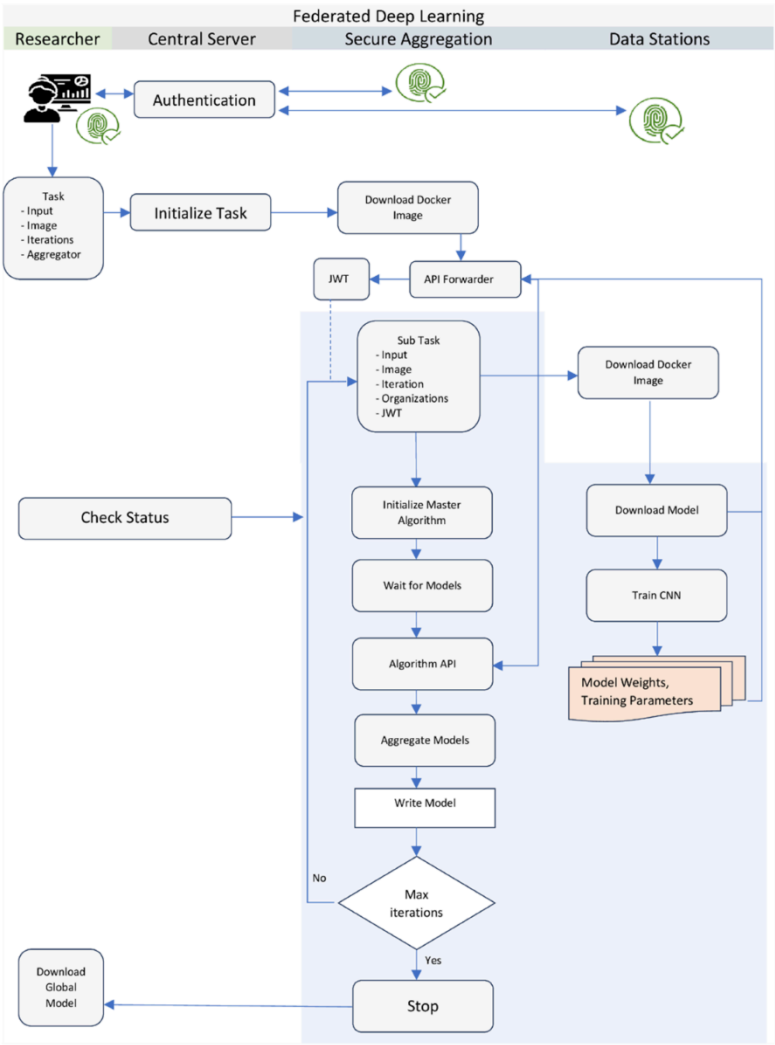



Figure 2: Illustration of federated machine learning training workflow.

Figure 2 depicts the workflow of the federated learning training process, where each of the components described above works in a coordinated manner to accomplish the convergence of a machine learning algorithm. The training process begins with the researcher authenticating with the central server. Upon successful authentication, the researcher specifies the task details, including a prebuilt Docker image, input parameters, number of iterations, and the identity of the SAS. The task is then submitted to the central server, which forwards it to the connected nodes. The SAS is the first to receive the task request. It downloads the specified Docker image from the registry and initiates the master algorithm. The master algorithm orchestrates the training at each data station node through the central server. The central server then forwards a subtask request to all the data stations. Like the SAS, the data nodes download the same Docker image and initiate the node part of the algorithm. The node algorithm runs the learning process on local data for the specified number of epochs. After each training cycle, the node algorithm sends the local model weights to the SAS.

The blue shaded section of Figure 2 shows the details of the security mechanisms used in vantage6. The SAS verifies the JWT signature of each received model and forwards

the request to the Algorithm API. The Algorithm API extracts the weight and metadata information of the models. Once the SAS receives all the required locally trained models for that cycle, it initiates the FedAvg algorithm to consolidate the models and create an intermediate averaged model, which is stored locally. This completes the first iteration of the training cycle. For the second and subsequent iterations, the data stations request the SAS to send the intermediate averaged model weights from the previous iteration. The SAS validates these requests and sends the model weights to the data stations, which then use them for further training on their local data. This cycle of training and averaging continues until the model converges or the desired number of iterations is reached.

At the end of the training process, the SAS sends a notification to the researcher indicating the successful completion of the task. The researcher can then download the final global model from the server. It is important to note that during the training iterations, the researcher or other users of the infrastructure do not have access to the intermediate averaged models generated by the SAS. This design choice prevents the possibility of insider attacks and data leakage, as users cannot regenerate patterns from the training data using the intermediate models.is conducted. key components of the architecture:

# 8 The PHT architecture in terms of EIRA building blocks

The European Interoperability Reference Architecture (EIRA) provides a framework for describing interoperable digital solutions. The info box below describes the concepts of Architecture Building Blocks, Solution Building Blocks and Solution Architecture Template that are used for this purpose.

# 9 Architecture Building Block (ABB)

Based on the TOGAF definition [3], an Architecture Building Block is an abstract component that captures architecture requirements and that directs and guides the development of Solution Building Blocks. An ABB represents a (potentially re-usable) component of legal, organisational, semantic or technical capability that can be combined with other Architecture Building Blocks. An Architecture Building Block describes generic characteristics and functionalities. Architecture Building Blocks are used to describe reference architectures, solution architecture templates or solution architectures of a specific solutions.

# 10 Solution Building Block (SBB)

Based on the TOGAF definition [3], a Solution Building Block is a concrete element that defines the implementation and Introduction to the European Interoperability Reference Architecture v3.0.0 27 European Interoperability Reference Architecture (EIRA©) v3.0.0 fulfils the required business requirements of one or more Architecture Building Blocks. On the technical view, a Solution Building Block is a specific product or software component and may be either procured or developed.

# 11 Solution Architecture Template (SAT)

A solution architecture template (SAT) is a specification containing including a sub-set of Architecture Building Blocks of the EIRA© and some optional Solution Building Blocks. It focuses on the most salient building blocks needed to build an interoperable solution addressing a particular business capability involving business information exchange.

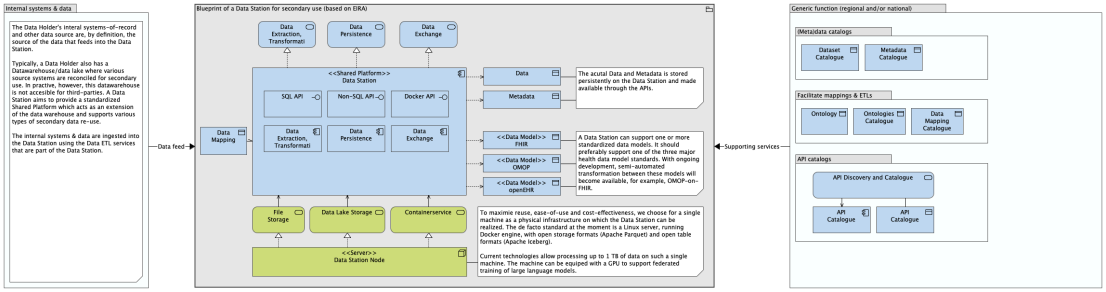
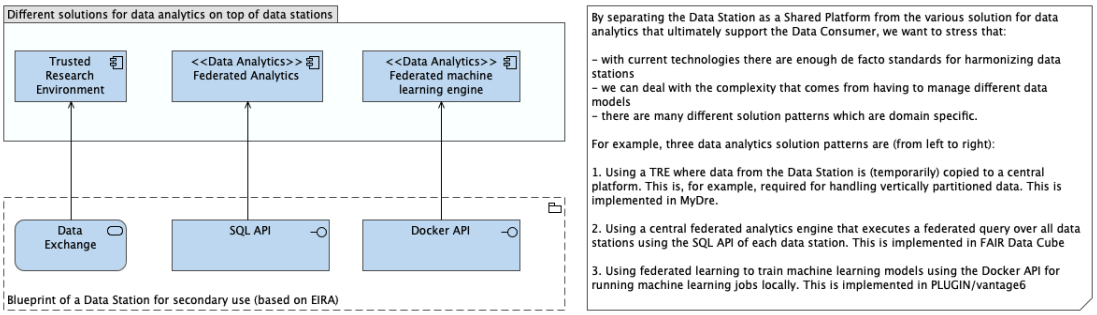



Figure 3: Blueprint of data stations for federated analytics in terms of EIRA ABB.

Each of the application components are described in EIRA with a unique identifier, as for example the *Data Extraction, Transformation and Loading*. To navigate the blueprint, refer to the online Archimate model [TO DO: add hyperlink].

Note that the blueprint explicitly accounts for hybrid SPEs, where data stations feed into a centralized Trusted Research Environment (TRE)




# 12 Defining layers of interoperability

We want to move toward open standards and specifications of the blueprint to move towards interoperability. This requires i) defining the layers; ii) defining the open standards and specification *within* each layer; and iii) defining the interface *between* layers. To put the issue into context, we first present existing frameworks for assessing interoperability that are relevant to data stations

## 12.1 EIRA LOST framework


Figure 4: The LOST layers of EIRA.

## 12.2 Interoperability within the DSSC Blueprint

To arrive at consistent conceptualization of data stations and trains, we want to relate the PHT architecture to the key components as defined in the DSSC Blueprint 2.0 (DSB2). The technical building blocks of DSB2 make an important distinction between between a control plane and a data plane. The control plane is responsible for deciding how data is managed, routed and processed. The data plane is responsible for the actual moving of data. For example, the control plane handles the identification of users and the handling of access and usage policies. The data plane handles the actual exchange of data. This implies that the control plane by nature can be standardised to a high-level, using common standards for identification, authentication, etc. The data plane can be different for each data space, as different types of data exchange take place. Some data spaces focus on the sharing of large datasets, others on message exchange, and others take an event-based approach. There is no one-size-fits-all, although there are some mechanisms (especially in the data interoperability pillar) which can assist in making sure different data planes work together.

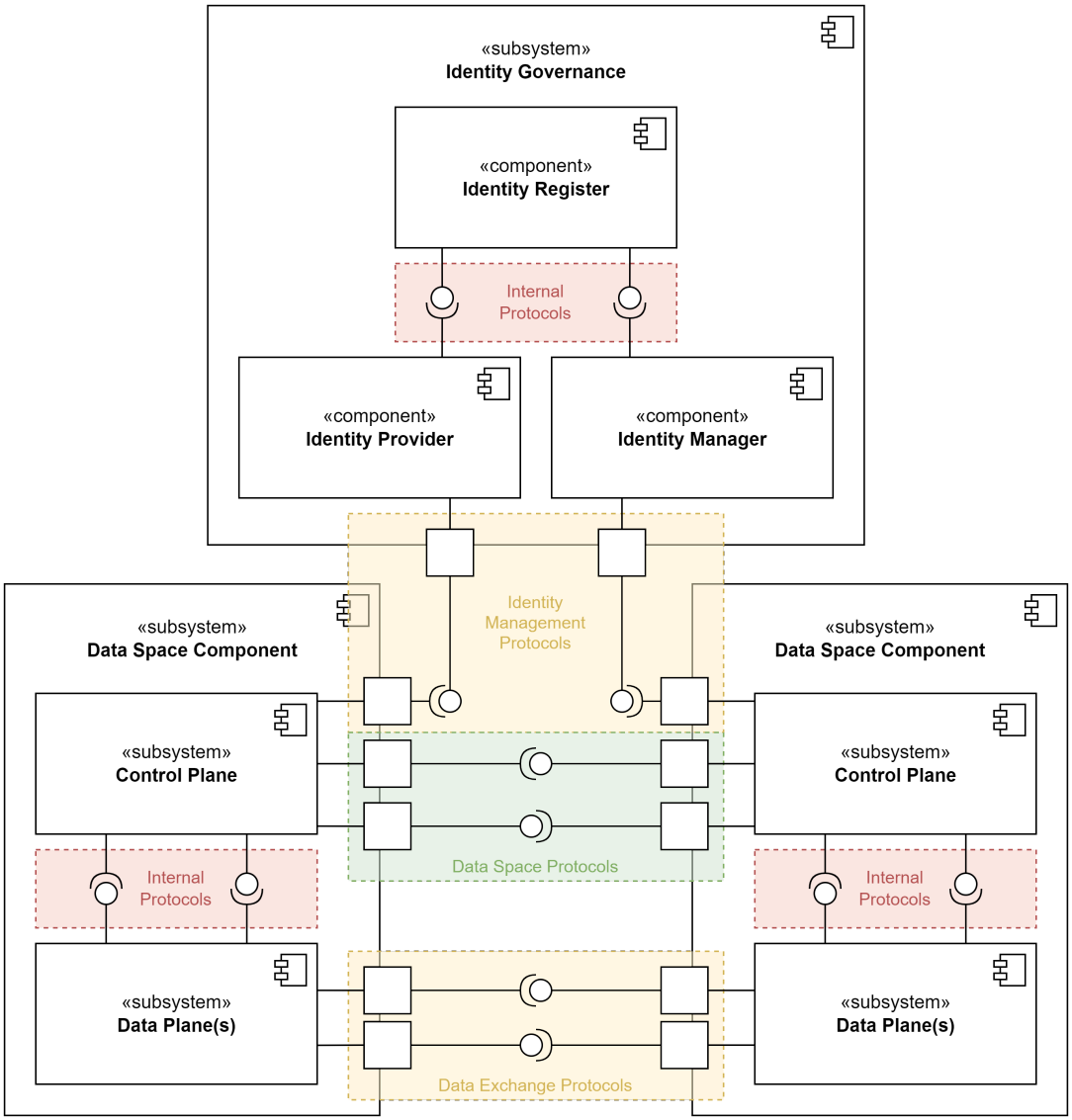



Figure 5: Conceptual overview of interoperability as defined in DSSC Blueprint 2.0.

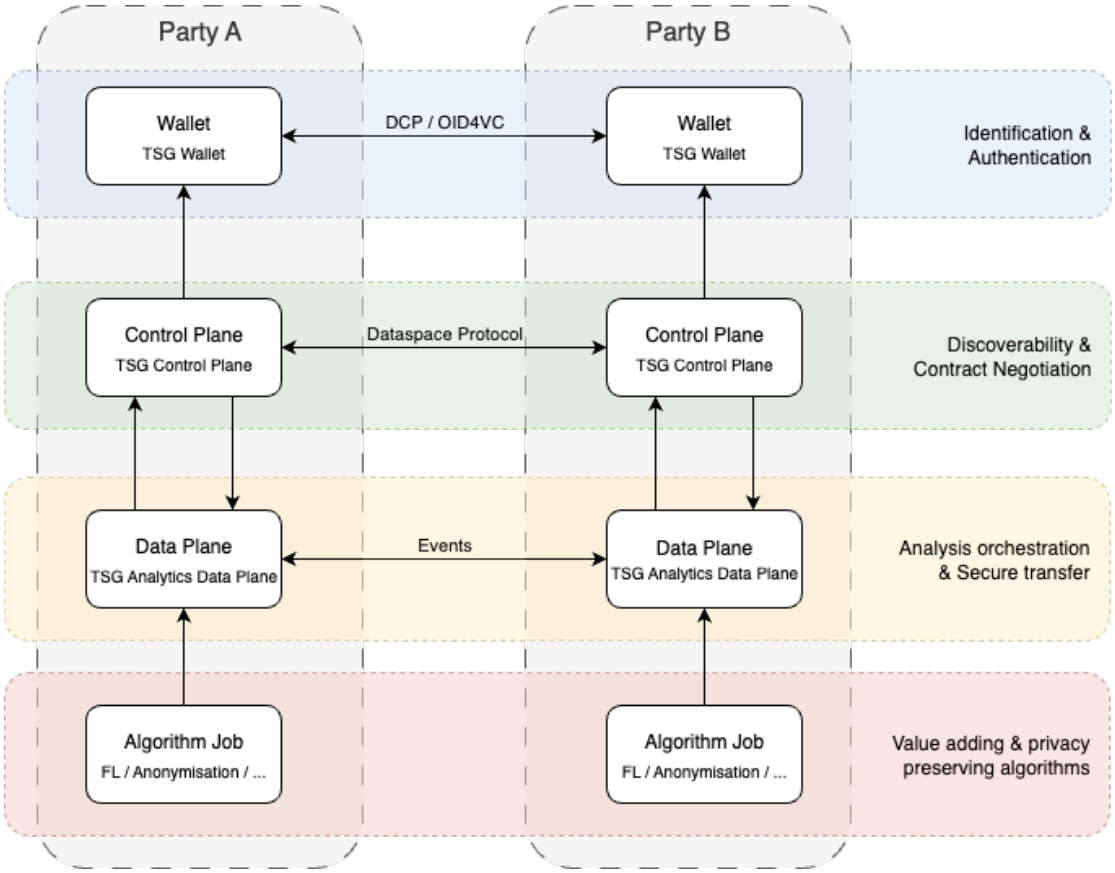



Figure 6: Simplified view of the four layers required for interoperability of data spaces. Image credit: Maarten Kollenstart, TNO.

- Communication within first three are envisaged to be done with did:web
- "Any Docker that emits events can be supported"
- "We can disregard Simpl Open"
- "Data plane = data station"
- Data space protocol and decentralized ID will be proposed to ISO as a standard

## 12.3 Layers of interoperability of PHT

Welten et al. [4] have defined five layers:

- **Layer 0 - Data integration.** The foundational step in our multi-layered approach involves harmonizing data across different infrastructures. This layer focuses on aligning and integrating the different data formats, structures, and standards from various data sources into a unified format such that it can be seamlessly processed by the analysis train.
- **Layer 1 - Assigning (globally unique) identifiers to stations.** In order to transfer trains between infrastructures, it is necessary to establish a method for identifying the station unambiguously across infrastructural borders. This is essential to ensure the correct routing of trains between the infrastructures and stations.
- **Layer 2 - Harmonizing the security protocols.** The PHT infrastructures were developed with different requirements regarding the security protocols and the encryption

of the train. Therefore, we formulate an overarching security protocol that aligns with infrastructure-specific requirements.

- **Layer 3 - Common metadata exchange schema.** By employing distinctive station identifiers (Layer 1), we establish the initial building block of a shared communication standard. As the security protocol also requires metadata for proper functioning (e.g., exchange of public keys), our third objective is to create a common set of metadata that facilitates technical interoperability and also extends to a first foundation for semantic compatibility. This layer primarily merges the metadata items from Layers 1 and 2 into a machine-readable format.
- **Layer 4 - Overarching business logic.** After we have established all the preliminaries mentioned above, we need to develop the actual business logic to transfer trains between the infrastructures from a technical perspective based on the route defined by the identifiers (Layer 1).

?!Do we need to define standards at the algorithm job layer? I think this is in fact less important, aside from the topology that you use for aggregation.

## 13 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

## Bibliography


1. Choudhury A, Volmer L, Martin F, et al (2025) Advancing Privacy-Preserving Health Care Analytics and Implementation of the Personal Health Train: Federated Deep Learning Study. JMIR AI 4(1):e60847. https://doi.org/10.2196/60847
2. 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
3. The TOGAF® Standard, Version 9.2. https://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html. Accessed 7 Jul 2025
4. 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