Towards a unified ontology of data stations and federated analytics hubs

1 Federated analytics and the Personal Health Train architecture

The network topology choice for implementing FL can vary from client-server, peer-to-peer, tree-based hierarchical, or hybrid topologies. 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. The choice of network topology for this study is based on a client-server architecture, offering a single point of control in the form of the central server. We take the Personal Health Train (PHT) architecture as our starting point [1]. Note that the architecture describes here is based on the vantage6 platform, which we aim to generalize in this paper.

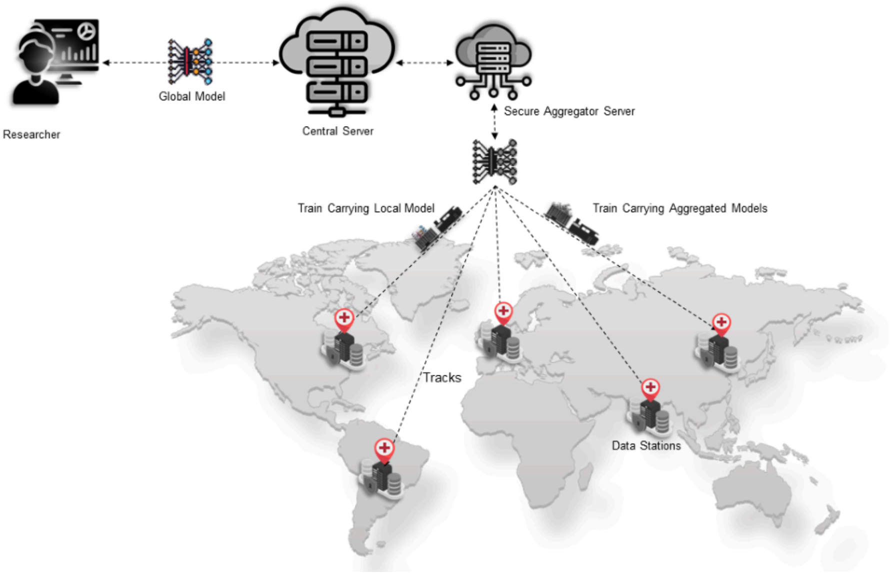

Figure 1: Overall architecture of a federated deep learning architecture adapted from Vantage6. The figure depicts a researcher connected to the central server, a secure aggregation server, trains carrying models, connected data stations, and the communicating tracks. Image source Choudhury A, Volmer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

Figure 1 depicts the key components of the architecture, which are described in the infobox below.

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.

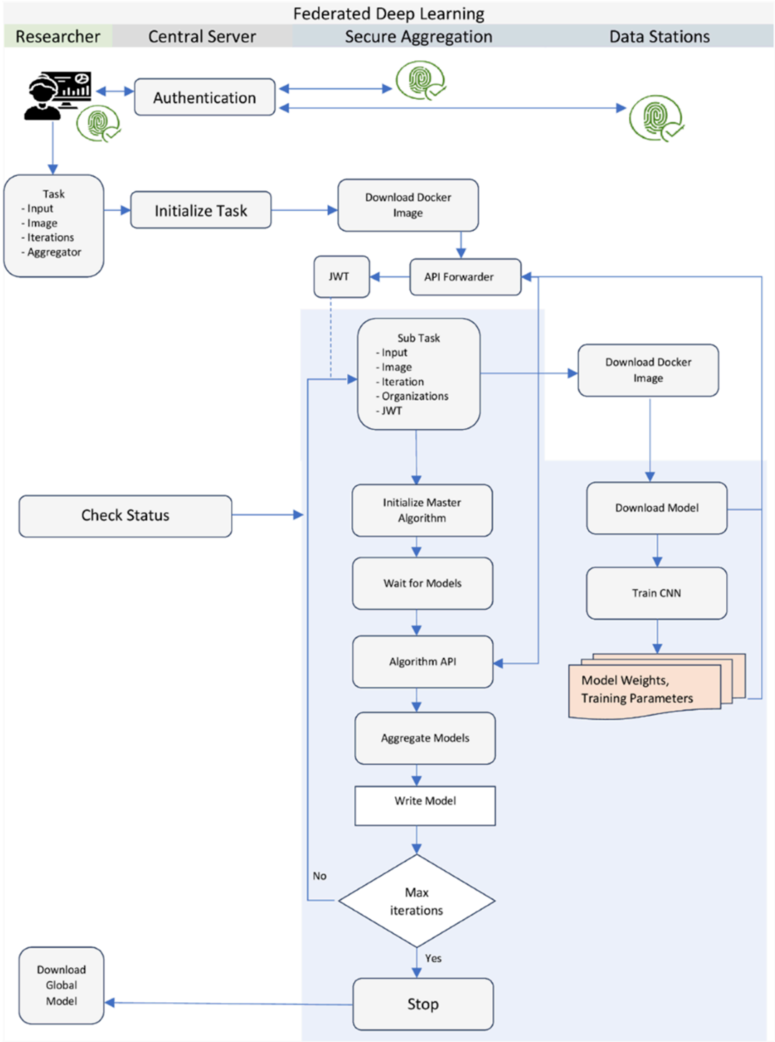

Figure 2: Illustration of federated machine learning training workflow.

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 Mapping the PHT architecture into the DSSC Blueprint 2.0

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.

8.1 Conceptual framework for interoperability

We follow Welten S, Arruda Botelho Herr M de, Hempel L, et al [3]:

- **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.

8.2 Ideas from TNO

Four layers 1. SSI wallet voor IAA functions 2. Control plane (Eclipse Dataspace protocol) 3. Analytics data plane 4. Algorithm jobs

The last two are considere *event-based collaborative analytics* 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

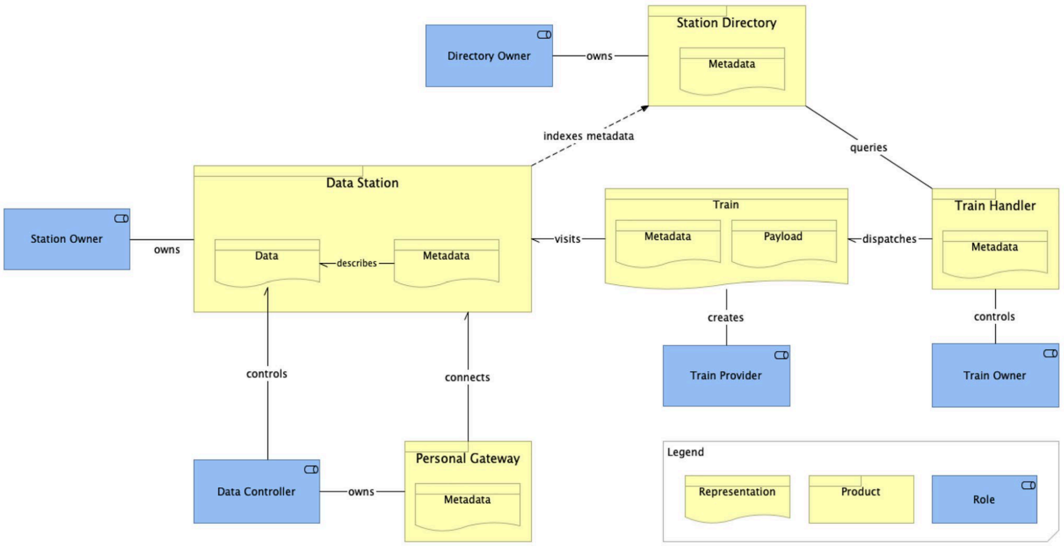

Figure 3: Main roles and components

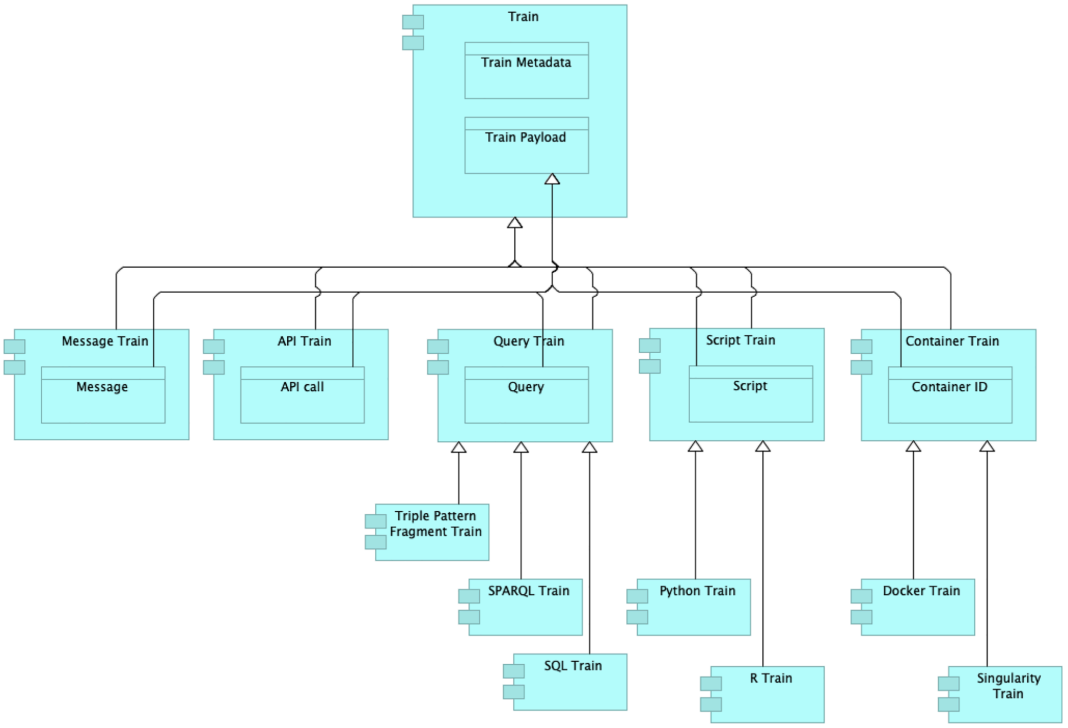

Figure 4: Train types

High-level overview of the Personal Health Train (PHT) architecture

In their paper Silva Santos LO Bonino da, Ferreira Pires L, Martinez V, Moreira J, Guizzardi R [4] continu 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.

9 Mapping PHT to the DSSC Blueprint 2.0

To arrive at consistent conceptualization of data stations and trains, Table 1 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 1: Mapping the key concepts from the PHT architecture [4] to the concepts of the DSSC Blueprint 2.0.

Component PHT	mapping to DSSC Blueprint 2.0 concepts
• Data Station	• Data Space Building Block
• Data • Metadata	• Data Product
• Data Controller	• Data Rights Holder
• Station Controller	• Data Product Provider
• Personal Gateway	• included in Participant Agent Services
• Station Directory	• included in Federation Services • Catalogue provisions and discovers offerings of data and services in a data space
• Directory Owner	• Common Intermediary provides federation services that are common to all participants of the data space
• Train	• Value Creation Services
• Train Provider	• Service Provider
• Train Handler	• specialization of Data Space Component that realizes the Train Value Creation Service
• Train Owner	• included in Service Provider as most generic role • concept of Intermediary (specialization of SP) is closer to definition of Train Owner

10 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

11 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 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 [5], Digizorg, Nordics	kapseli, Mayo, …
decentralized	RSO Zuid Limburg, Twiin portaal, …	many federated analytics research networks such as x-omics programme and EUCAIM

12 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. 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. 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

4. 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

5. (2024) OpenHIE Framework v5.2-En. https://ohie.org/. Accessed 27 Aug 2024