Data Protection Impact Assessment (DPIA) for vantage6

Version history

Version	*Date*	*Authors*
2.1	November 12, 2020	Gijs Geleijnse, Daan Knoors, Arturo Moncada-Torres, Ferdinand Impens, Marilyn Maas, Jaap van Ekris
3.0	May 30, 2023	Ellen Bosma, Gijs Geleijnse, Frank Martin, Bart van Beusekom, Daan Knoors
3.1	August 21, 2023	Gijs Geleijnse

Participants in version 3

- Daan Knoors – Clinical Data Scientist (Cancer Registry of Norway)
- Gijs Geleijnse – Sr Clinical Data Scientist (IKNL)
- Jan Nygard – Innovation Manager (Cancer Registry of Norway)
- Eva Polk – Project Manager Blueberry (IKNL)
- Ellen Bosma – Legal counsel (IKNL)
- Frank Martin – Scientific Programmer (IKNL)
- Glendanique Minguel – Security Officer (IKNL)

Reviewer

- Merle Hafkamp (Data Protection Officer IKNL)

Contents

No table of contents entries found.

1. Introduction

General

Vantage6 is an open-source infrastructure developed by IKNL, eScience Center, Maastro and other partners. Vantage6 enables parties to gain insights from sensitive data (individuals, patients, citizens) from different sources, without transferring the data or inspecting items from individuals within the datasets. This is done through the application of privacy enhancing technologies (PETs), including federated learning (FL), secure multi-party computation (MPC), homomorphic Encryption (HE) and differential privacy (DP). These technologies enable analysis of data, while protecting the sensitive information of individual data subjects. Each technology brings its own form of complexity to the analysis and often a mix of them is required to get the most effective result. This usually depends on the research question you would like to answer, the actors involved, the type of data, the analysis methods, computational resources, and presence of other available safeguards.

This DPIA examines the privacy impact of the vantage6 infrastructure: how does the vantage6 infrastructure relate to the GDPR and which privacy risks can be identified as well as measures that can be applied to mitigate these risks. It is also assessed whether and, if so, where in the vantage6 infrastructure personal data is processed within the meaning of the GDPR and to what extent PETs could lead to anonymity in a legal sense. The roles as defined in the GDPR of the actors within the vantage6 infrastructure are assessed and other GDPR obligations are discussed (including storage limitation, data minimization, privacy by design and default).

This DPIA is intended to be independent of specific collaborators, algorithms and data sets. The risk of exposing personal data is partially dependent on the specific requirements of the project at issue, e.g. which data will be used or which algorithms will be executed and how often? This document describes the risks for the general use case and makes no assumptions on project specifics. It can serve as a starting point to evaluate the risk of a specific project in which vantage6 is intended to be used.

For algorithms used with sensitive data within the vantage6 infrastructure, a separate privacy impact analysis may be required. Depending on the size (e.g. number of patients) and modality (imaging, clinical features) of the data in combination with the algorithm (e.g. summary statistics, regression, deep learning), risks of sharing identifiable data outside the organizations should be assessed.

PETs

PETs are technologies that embody fundamental data protection principles by minimizing personal data use, maximizing data security, and/or empowering individuals. Data protection law does not define PETs. The concept entails many different technologies and techniques. The European Union Agency for Cybersecurity (ENISA) refers to PETs as: ‘software and hardware solutions, i.e. systems encompassing technical processes, methods or knowledge to achieve specific privacy or data

protection functionality or to protect against risks of privacy of an individual or a group of natural persons.'[1]

PETs are available for a variety of purposes (e.g. secure training of AI models, generating anonymous statistics and sharing data between different parties). Homomorphic encryption (HE) provides strong security and confidentiality by enabling computations on encrypted data without first decrypting it. Secure Multi Party Computation (MPC) provides data minimization and security by allowing different parties to jointly perform processing on their combined data, without any party needing to share all of its data with each of the other parties. Also in MPC, data is only shared in encrypted format. Federated learning (FL) trains machine learning models in distributed settings while only sharing aggregate data with each party. Differential Privacy (DP) is a mathematical framework that quantifies and limits the amount of information that can be learned about data subjects with the release of aggregate information.[2]

A paradigm shift to decentralized analysis

Traditionally, when a researcher wants to analyze data from different sources, these datasets need to be requested, prepared and shared by each dataholder to the researcher. This means that patient-level data leaves the respective organizations and is brought together on the machine of the researcher.

In recent years, concerns around ensuring patient privacy have increased, making organizations more hesitant to share record-level data with third parties. On the other hand, to progress our knowledge on healthcare in general and cancer in particular, there is an increasing need to combine both horizontally as well as vertically partitioned data.

The Personal Health Train

The Personal Health Train (PHT) is the initiative to provide a solution to patient-level data sharing concerns. The PHT is a paradigm to enable analyses of distributed data from multiple organizations, without patient-level data leaving the organizations. This privacy-by-design paradigm enables researchers to conduct their analyses, while not accessing or "seeing" individual patient records. By keeping data at the source, no copies of the datasets are generated that are shared with third parties. It enables data custodians to remain in control of the access to their datasets, while enabling analyses at-scale.

vantage6 is the open source implementation of the PHT (1, 2) (Djura Smits, 2022). Following the metaphor of the Personal Health Train, we identify:

1. **Stations:** locations where data is hosted that is made available for analyses using the PHT. Data providing organizations can host a station themselves or they can work with an external party to host the data (e.g., a cloud provider).

2. **Rails:** the technical infrastructure that connects the stations. vantage6 is the infrastructure that implements authentication and authorization, such that the right parties are connected in the right way.

3. **Trains:** statistical analyses on the data stations. An analysis script is composed of multiple trains (e.g., containing descriptive statistics, collecting information for tables and figures as well as more advanced regression and machine learning analyses).

[1] ICO guidance doc
[2] Idem

4. **Journey:** A full study involving one or more stations with dedicated datasets connected via the rails with a researcher (Client), who can send a predefined selection of trains to these stations.

Data partitioning

We distinguish two forms of analyses using data from multiple organizations and/or databases (Fig. 1). These vary in the manner in which the data is distributed, namely:

- **Horizontally-partitioned data,** where two or more organizations record similar data items but for different data subjects.. An example is the combination of data from multiple cancer registries that cover different geographies and patients. Combining cancer registry data allows for inter-geographical comparisons and creates a large patient volume. The latter is particularly relevant for the research on rare cancers. As databases contain data from different patients, matching identifiers between databases is not a concern for horizontally partitioned data.

- **Vertically-partitioned data**, where data items for a group of individuals are distributed across several databases. For example, the data items on cancer patients in a cancer registry and the data items recorded by an insurance provider. For vertically partitioned data, the identifiers of the patient records should be matched across databases.

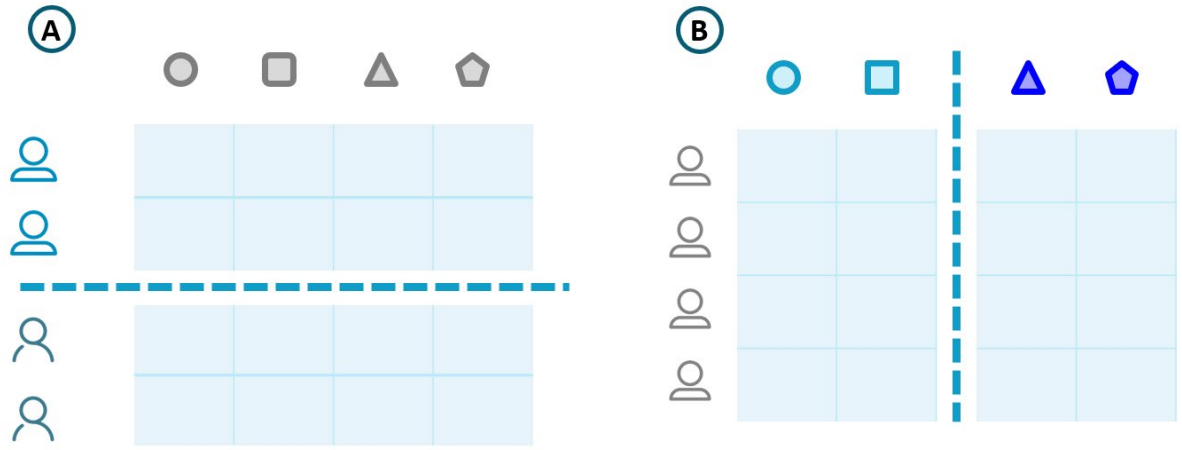


Figure 1. (A) Horizontally-partitioned data contains records from multiple organizations with the same features from different patients (e.g. cancer registry data from the Netherlands and Czech Republic). (B) Vertically-partitioned data contains records with different features with the same patients (e.g. cancer registry data from the Netherlands and socio-economic data on these patients from CBS Statistics Netherlands). Image adapted from (1).

2. Description of vantage6

For a journey on the PHT using vantage6, we distinguish the following computer architecture (Fig. 2):

- **Client:** a computer of a researcher, epidemiologist, or other professional requesting insights via a *journey*

- **Station:** a (virtual) machine where one or more datasets for a participating organization is stored and made available. With each journey, a dedicated dataset is associated.

- **Central server**: a machine where the journey is managed, and communication between stations is orchestrated and computations are performed on non-identifiable data and statistics received from the stations.

For a detailed description of version 3 of vantage6, we refer to the paper[3].

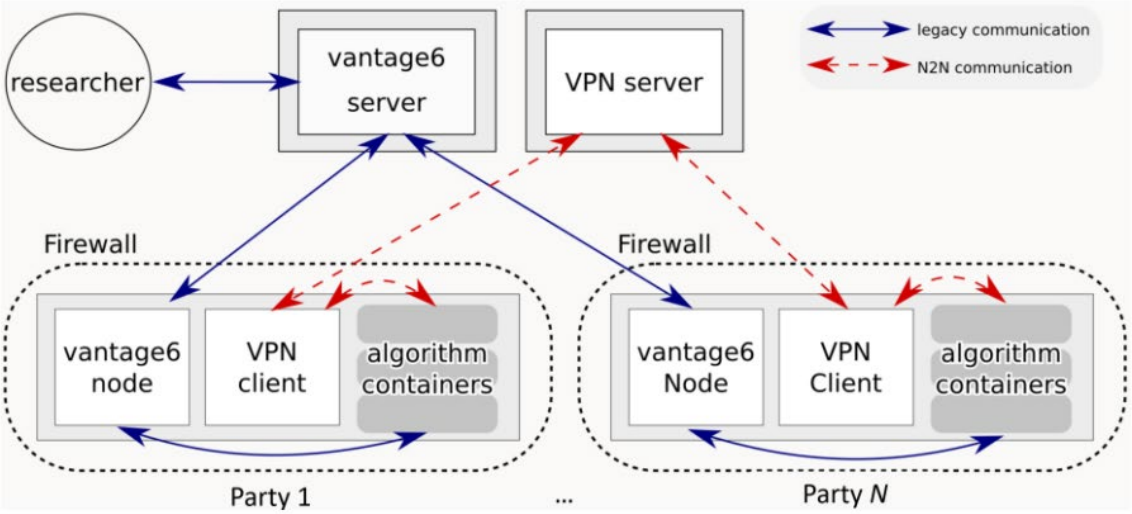


Figure 2. The architecture of vantage6 version 3 as described in Smits et al.

The vantage6 server consists of the following components:

- **EduVPN**, enabling the use of advanced algorithms (such as MPC algorithms) that require node-to-node communication. EduVPN provides an API for the OpenVPN server, which is required for automated certificate retrieval by the nodes. Like vantage6, it is an open source platform. This component is optional and not required in collaborations where no node-to-node communication is required. [4]
- **RabbitMQ (Message Queue)**, enabling the server to handle multiple requests at the same time. This is important if a server has a high workload. This component is also optional.[5]

[3] https://vantage6.ai/documents/15/smits2022improved_xYljLTd.pdf
[4] https://old-docs.vantage6.ai/installation/server/eduvpn
[5] https://old-docs.vantage6.ai/installation/server/rabbitmq

- **The Docker Registry**, a repository providing storage and versioning for Docker images. The installation of a (private) Docker registry is done when a collaboration wants to securely host a collection of algorithms.[6]

In practice, multiple organizations may be involved:

5. The client's organization ("*Client or Data requesting Party*")
6. The organization providing the PHT service and managing the central server ("*Central Server Manager*")
7. The *Central Server Provider*, the party hosting the central server on behalf of the central server management (e.g., a cloud provider).
8. The organizations hosting the data stations - whether or not - on behalf of the data providing organizations ("*Providers of PHT stations*")
9. The data providing organizations ("*Data providing organizations*")

An example of such a set up for a journey can be found in Fig. 3.

The roles in vantage6 correspond to the ones defined in the note by Bontje (3) (Fig. 4).

	Role in (3)	Role in vantage6
Data requesting site	Opdrachtgever van de PHT trein	Data requesting party / Client
PHT Domain	Aanbieder van de PHT trein (provider of the PHT train)	Central Server Manager
	Aanbieder van PHT station	Provider of PHT station
Data provider site	Aanbieder van PHT data	Data providing organization

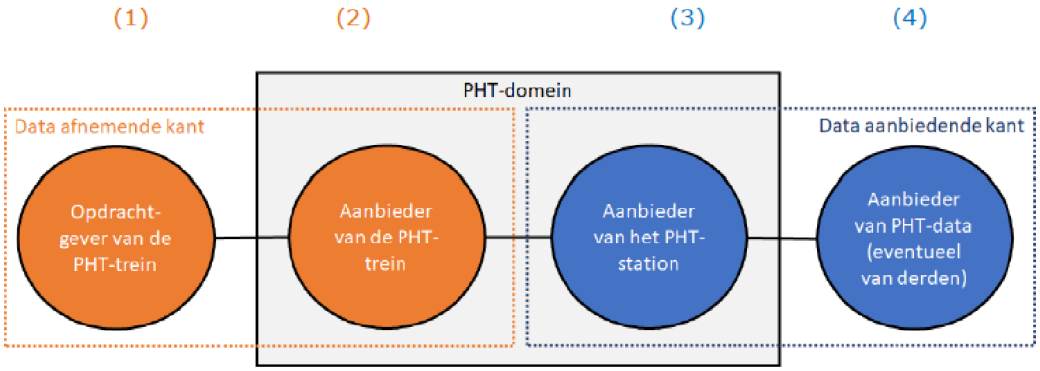


Figure 4. Roles as defined in "privacyaspecten van de personal health train" (3)

3. Trains in vantage6: Federated Learning and Multi-Party Computation

In vantage6 trains can use a variety of different PETs but to allow for computation on distributed data, it at least requires either a form of Federated Learning and/or Secure Multi-Party Computation.

[6] https://old-docs.vantage6.ai/installation/server/docker-registry

Federated Learning

Federated learning is typically applied for horizontally-partitioned data (i.e. organizations provide data from disjunct cohorts of patients, yet providing the same characteristics/items). Federated learning is based on the mathematical principle of splitting a computation into (a) parts at the stations and (b) a central part. The stations share sub-computations with the central server.

For example, let's suppose one is interested in the average age of all patients in a cohort

$$Average = \frac{\sum_{i=1}^{n} age(i)}{n}$$

Here, $age(i)$ is the age of patient i in a group of n patients.

Now suppose that the group of n patients are now divided over multiple organizations that do not want to share the age of individual patients. A federated algorithm to compute the average age, would require that each station computes the following:

1. $\sum_{i=1}^{m} age(i)$ – The sum of the age of all patients in the cohort at the station
2. m – the number of patients in the cohort at the station

The central server now collects these statistics from all participating stations. To compute the average over the entire group, it combines the sums of the ages (1) and divides this by the total number of patients in each cohort (2).

This principle of splitting computations into a central- and a station-part is illustrated here by a simple example, but can be applied in complex computations as well (4–8).

Note that the technique of splitting computations as explained above only works for horizontally partitioned datasets.

For vertically partitioned data, federated learning can be used to approximate centralized calculations for some tasks. At the moment of writing, one algorithm (for logistic regression (1)) was included in the vantage6 library. However, not all types of analyses can be implemented using federated learning, particularly when an algorithm cannot be mathematically separated.

Multi-Party Computation

Similar to Federated Learning, Secure Multi-Party Computation (MPC) enables various organizations to perform a joint analysis without the need to share raw sensitive records. However, instead of mathematically decomposing an algorithm, MPC relies on a toolbox of cryptographic techniques that allows several different parties to jointly compute functions on encrypted data. This form of encryption makes it safe to share the data among parties, while still supporting specific mathematical operations. After completing the computation, only the final result can be decrypted and the participating parties determine who is allowed to view the outcome of the computation.

Let's use the same example as above to calculate the average age of some participants through an MPC protocol. For this, we can use a secure sum algorithm.

Let us assume that there are 3 registries (A, B and C having respectively a, b and c patients) that want to know the total number of patients of all registries (N=a+b+c) but do not want to share their number of patients with the other registries. However it is accepted that the other registries learn the average of the other two registries. An algorithm that solves this problem is show in Fig. 5:

1. A generates a random number R
2. A add this number to its patient count x1=a+R
3. A shares x1 with registry B
4. B adds its patient count x2=x1+b
5. B shares x2 with registry C
6. C adds its patient count x3=x2+c
7. C shares x3 with registry A
8. A subtracts the random number N=x3–R
9. Optionally A shares the result N with B and C

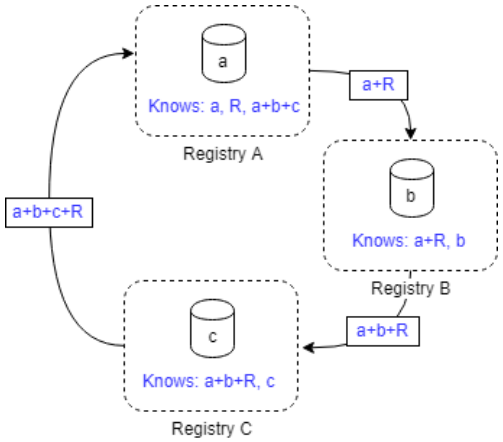


Figure 5. Example algorithm for MPC.

Having calculated the number of participants securely (i.e. the denominator of the average age formula), one can follow a similar process to obtain the sum of all ages in each registry.

Observe that, through this example we show how we can still share and do computations on encrypted numbers that do not reveal anything about their true values. With a bit more complexity this scenario extends itself to full databases, including varying column types and dimensions. Contrary to federated learning, MPC is a bit more flexible and can be used for both horizontally and vertically partitioned data. It has technical guarantees to perform all computations securely and thus keep all input and intermediate results encrypted. Only the final result is revealed and thus this prevents any inferences being made on intermediate output. Hence, the security properties provide superior privacy protection, but do come at the cost of added algorithm complexity, computation time and communication rounds, i.e. many MPC protocols require that trains need to pass by stations multiple times. Therefore, MPC solutions are more difficult to develop and interpret and often require a custom implementation for specific use cases to achieve the required efficiency.

Train Certification

In vantage6, a *train* is an analysis script implemented in a Docker container. Docker is a technology to execute a script on a machine without installation of additional software packages outside of the container. If a programmer creates an analysis in a certain programming language (e.g., version 3.1.5 of the language R), Docker creates a virtual machine, i.e., it compiles and executes the analysis script as was generated on the machine of the programmer.

In order to transform a software script to a vantage6 train, a Docker container of the script is created. The Central Server Manager will send this Docker container to the data station in order to execute the analyses as was defined in the journey.

To certify that the Docker container corresponds to the corresponding script, vantage6 makes use of Docker Notary (https://docs.docker.com/notary/getting_started/). Docker Notary allows us to verify the author of the Docker container. At the moment it is the *de facto* technology to implement this functionality.

3

3.1 vantage6 in Practice: Adoption in Workflow and Processes

To use vantage6, we identify two steps: (1) the deployment of vantage6 at the organization and (2) the process of using vantage6 for a single study/journey.

Deployment of vantage6

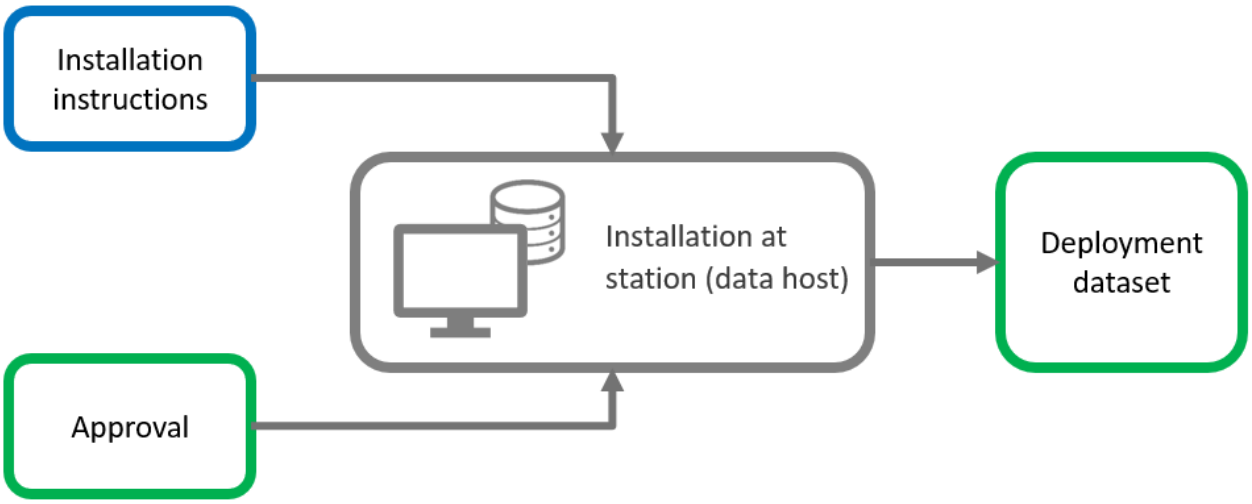


Figure 6. A data providing organization will be working with the PHT service provider (in blue) for the technical installation of the vantage6 software. This will either be done within their own IT infrastructure or at a Data Hosting organization. Installation will take place after receiving the approval necessary. vantage6 is ready for usage once a dataset from the organization is (and may) be made available within the vantage6 infrastructure.

Approval may not only include approval for local usage and installation, but also a contract with other data providers providing a framework to facilitate studies using their respective data.

Usage vantage6 for a single journey

Once all participating organizations are prepared to partake in any journey using vantage6, researchers ("Clients" in gray) can utilize vantage6 to conduct their studies. We visualize this process in Fig. 7 - from study idea to execution.

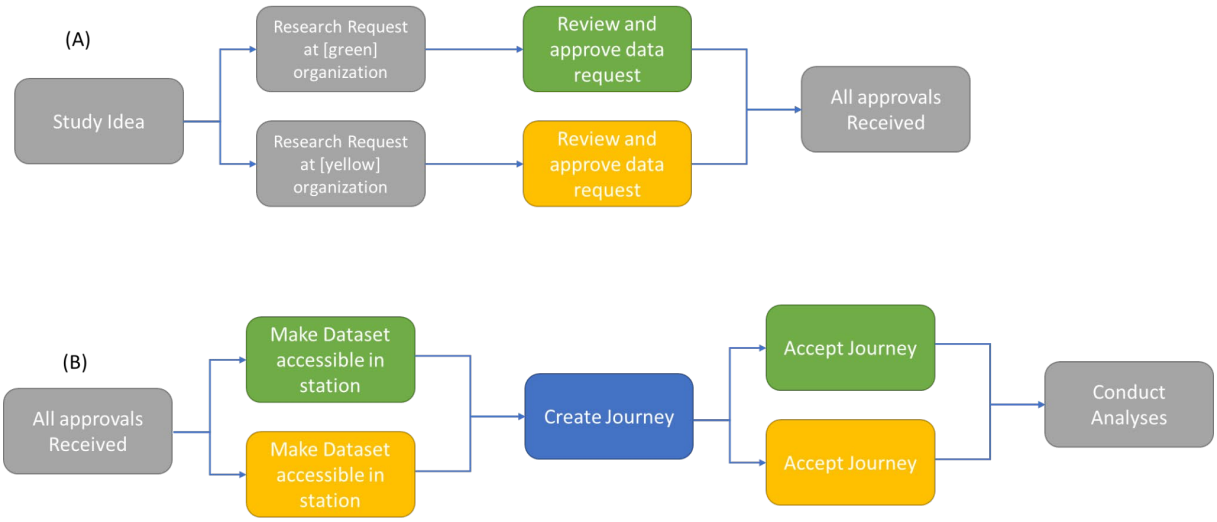


Figure 7. (A) Seeking approval according to data usage/research request processes at all participating organizations. In this example, we assume two (green and orange). (B) From Approval to Analysis on vantage6. The organization managing vantage6 and the central server manager creates a "Journey" (defining trains, privacy aspects, stations, dataset, users) that is to be accepted by the data providing organizations.

Following the flow-chart, we envision the following steps:

1. A Client (researcher or user) has a study idea.
2. They file data usage requests to the designated bodies responsible for each of the data sets. The data usage requests defines the journey:
    a. Which data providing organizations are involved?
    b. Which datasets are requested (from all organizations)?
    c. Which trains (analyses) are to be conducted on these datasets?
    d. Who (besides the Client) should have control to execute the analyses?
3. These organizations all review the research/data usage requests
4. After the data request has been accepted by all participating organizations, the Client requests them to make the dataset accessible in their Stations. After doing so, these datasets cannot be accessed/analyzed by any outside party yet.
5. The PHT service provider (blue) defines the journey according to the specification in the data request. It associates the datasets made available in the stations with the journey. Moreover, the trains are selected that can be used (and can only be used) to analyze the data. Lastly, the user(s) of the system are identified and logins are created.
6. Before being able to conduct any analysis, all data providers are required to accept the journey. The specification of the journey is shared with these organizations and they are invited to review and compare with the original data request. The approvals are logged both locally at the station as well as at the central server.
7. After all data providing parties have granted permission, the user(s) can execute their research by running the trains as defined in the journey.

For vertically-partitioned data, we assume at this moment that patient IDs between the organizations have been matched. The datasets will be disconnected from the station after the time defined in the data request.

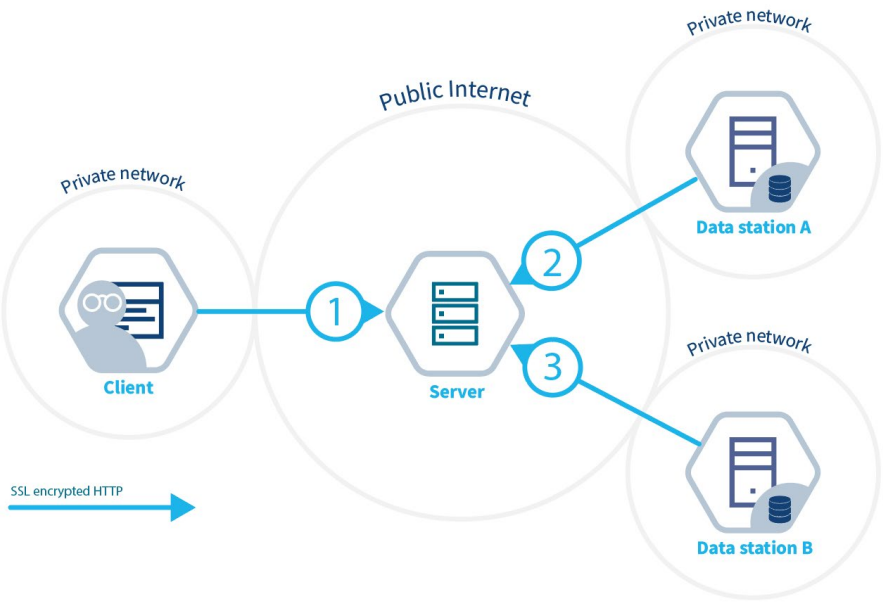
4. Personal data

The personal data processed within the vantage6 infrastructure depends on the nature of the data provided by the parties involved and the associated categories of data.

5. Data Processing

The journey as identified on page 14 will hereinafter be translated focusing on the data flows in order to identify in which steps personal data within the meaning of the GDPR are processed.
The data processing operations (in Dutch: *gegevensverwerking*) which can be identified within the vantage6 infrastructure are as follows:

1. A collaboration involving stations, accepted algorithms and privacy settings (e.g. subsets can only be created if size difference is at least 2) is defined and accepted by all parties (*no processing of personal data involved in this step*).
2. The client selects an **algorithm** (i.e. the corresponding docker container) **and parameters** in line with what it agreed to in the data requests (for example: descriptive statistics, imputation of missing values, a regression) and sends the algorithm to the central server (*no processing of personal data involved as no operations have yet been performed on the local data*);
3. The data stations within the collaboration are identified and logins are created (*no processing of personal data*);
4. (The node of) the data station receives the **algorithm and parameters** (e.g a logistic regression with parameters “age”, “sex” and “stage”) from the central server (*no processing of personal data*)
5. (The node of) the data station executes the **algorithm with the specified parameters** on the local data (*processing of personal data as local (patient level) data is involved* (hereinafter also referred to as: input data);
6. (The node of) the data station sends the **computation results** back to the central server (*processing of personal data depends on the algorithm, privacy settings and data*)
7. In case of an iterative algorithm: the central server receives the aggregated computation results and processes them. If the end criterion of the iteration is not reached (e.g. number of rounds or convergence), the central server will update the algorithm + parameters and send these back to the nodes to repeat step 4 to 6. When the stopping criteria is reached, the central server receives and stores the **aggregated computation results** and sends the **aggregated computation results** to the client (*processing of personal data depends on the algorithm, privacy settings and data - with appropriate privacy settings, only aggregated data is processed*);
8. In case of a non-iterative algorithm: the central server sends the **aggregated computation results** to the client *(processing of personal data depends on the algorithm, privacy settings and data - with appropriate privacy settings, only aggregated data is processed*)



Which data flows are covered by the GDPR?

The data flows have been identified above (steps 1 - 8). In order to determine whether the GDPR applies to (certain of) these data flows, it must first be assessed whether personal data within the meaning of the GDPR are processed or whether anonymous data is concerned (outside scope GDPR).

With regard to steps 1 - 4 as described above (p. 15), it can be established that no personal data is processed since the local (patient level) data is not yet involved. In step 5 it is clear that personal data are processed as operations are performed on patient level data (input data). Important to note is that this is done locally at the respective data-holding organization.

It is more complicated to assess whether personal data are processed in the context of steps 6-8: more specifically, can the **computation results** sent by the data station to the central server in step 6 contain personal data (in the definition of the GDPR)? And: can the **aggregated computation** results received, stored and sent by the central server to the client contain personal data? To what extent it is likely that these computation results reveal information about individuals in the underlying patient data? While aggregate statistics certainly feel safer, they are still susceptible to privacy attacks. For instance, a differencing attack aims to single out an individual's value through a combination of aggregate statistics. Therefore, one needs to evaluate the associated risks on a per-algorithm basis and implement the necessary safeguards to prevent the algorithm from leaking more information than intended.

Can PETs (from a legal point of view) lead to anonymization of personal data?

Due to different approaches and interpretations, legal uncertainty currently exists on how PETs relate to the GDPR and whether the application of certain (or a combinations of) PETs can ensure the data to be considered anonymous in a legal sense (and therefore outside the scope of the GDPR). As seen below, the majority of authors who have conducted research hold the view that the output *could* still contain personal data (this will be discussed in more detail below).

In order to understand the method of interpretation and the different approaches, (the scope of) the concept of personal data in a legal/GDPR-sense is discussed and the way to assess whether data qualify as personal data or anonymous data.

GDPR: personal data

The GDPR only applies if personal data are processed, whereas non-personal data fall outside its scope. Pseudonymised data is considered to still qualify as personal data. The broad definition of **article 4(1) GDPR** reads as follows:

> "Personal data means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person."

Recital 26 GDPR tries to offer a test to differentiate between personal and non-personal/anonymous data:

> "To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments."

In practice, the answer to the question whether data qualifies as personal data or anonymous data depends on the approach followed, arising from a difference in the interpretation of art. 4(1) GDPR and recital 26 GDPR: **the absolute approach and the relative approach**. The different approaches determine the perspective the controller has to take into account in the assessment provided by recital 26 GDPR.

The **absolute approach** requires that to classify data as anonymous, no remaining risk for re-identification is acceptable. This means that if the data are identifiable for one party (for example the holder of the original dataset) the data are considered identifiable for each party, irrespective of whether it is impossible or not for that party to identify the individual. The absolute approach is followed by the Article 29 Working Party (currently: the European Data Protection Board, EDPB) in its opinion on anonymization techniques from 2014[7], which opinion is still quoted by the EDPB.[8] Rulings from national authorities (for instance Austria and France) follow the absolute approach as well.

The **relative approach** accepts that there is always a remaining risk of re-identification. The encrypted data shall only be personal data for a controller or processor who has the decryption key. The judgment of the Court of Justice of the European Union (CJEU) in Breyer vs. Germany supports this approach. In that judgment the court - in essence - held that in order to determine identifiability it needs to be assessed whether a party has means that can reasonably be used to identify the individual. The court states that this is in any case not the case if it would require an excessive effort or if identification is prohibited by law.

In its recent judgment of 26 April 2023 the CJEU confirmed the approach provided in Breyer and emphasized that in order to determine whether the data constitutes personal data, it is necessary to look at the perspective of the receiving/processing party: does the information transferred to/processed by that party relates to identifiable person(s)? This confirms that the question of

[7] The WP29 opinion suggests that data resulting from personal data remains personal data as long as the original data set is preserved. EDPB refers to 2014 opinion in guidelines on consent May 2020 and in COVID-19 guidelines.

[8] By still insisting on Opinion 5/2014 the EDPB seems to ignore that in 2016 (Breyer) the CJEU gave a different test to decide whether data are anonymous or not.

applicability of the GDPR is not whether data is identifiable to a party (someone) (absolute approach), but whether the data is identifiable to a specific party, namely the processing party.

It seems there are sufficient arguments to successfully argue that the interpretation of the 'identifiability test' of recital 26 as provided in CJEU Breyer 2016 and confirmed in the judgment of CJEU of 26 April 2023 is currently leading.[9]

Summarizing: to assess whether data is considered identifiable the aforementioned judgements provide the following test (to be performed from the perspective of every processing party):

1. **Does the party who processes the data have means that can reasonably be used to identify the individual?**

    a. Does identification require a disproportionate effort in terms of time, costs and man-power, so that the risk of identification appears in reality to be insignificant?
    →if so, then the party concerned is considered not to have means that can reasonably be used to identify; or
    b. Is identification prohibited by law?
    →if so, then the party concerned is considered not to have means that can reasonably be used to identify.
    c. Consider all objective factors including; costs of and amount of time required for identification and the available technology at the time of the processing and technological developments.

Briefly said, it must be assessed for every processing party within the infrastructure whether identification requires a disproportionate effort in terms of time, cost and man-power (and other objective factors), so that the risk of identification appears in reality to be insignificant. The outcome will depend on the context/circumstances of the specific case: e.g. the data, the specific (combination of) PETs applied, the algorithms used, privacy settings, the nature of the receiving party.

In the literature (academic papers), most adhere to the view that it cannot be ruled out that the output of the computations contain personal data. Some indicate that it depends on the technique used, but in those cases it is not clear which factors and which (combination of) techniques would then lead to anonymity in the sense of the GDPR.

The identifiability test to be performed on a case-by-case basis

The aforementioned test (a, b, c) is to be performed on every data flow (see page 14 step 1-8) in the context of setting up a project using vantage6. It will depend on the circumstances (e.g. the specific PETs applied, algorithms used, privacy settings, the receiving party) whether the computations are considered to contain personal data.

The use of the 'ICO-guidance document' on the use of PETs and the guidance on anonymization (providing practical guidance) is encouraged when setting-up a specific project / infrastructure and assessing the identifiability of individuals to which the underlying data relates.[10]

[9] With regard to the status of opinions, recommendations of administrative agencies such as the EDPB can be referred to Groos and Van Veen who state: "*While admitting the various interpretations of the rule of law under legal scholars, one of its pillars is that in the end the court decides and not an administrative agency. In that sense it is somewhat disappointing that the EDPB never reconsidered its Opinion 5/2014 in the light of this decision [Breyer vs Germany], assuming that it could ignore the criticism in the literature*."

[10] Chapter 2 Draft anonymisation, pseudonymisation and privacy enhancing technologies guidance, Chapter 2: How do we ensure anonymisation is effective? p. 11 and further, and chapter 5: Privacy-enhancing technologies (PETs).

This DPIA is intended to be independent of specific collaborators, algorithms and data sets. The risk of exposing personal data is partially dependent on the specific requirements of the project at issue, e.g. which data will be used or which algorithms will be executed and how often, what are the privacy settings? This document describes the risks for the general use case and makes no assumptions on project specifics. It can serve as a starting point to evaluate the risk of a specific project in which vantage6 is intended to be used. Which technology or combination is the most effective depends on the context/circumstances, the type of data, the actors involved and other available safeguards.

Papers that conclude that the specific PET assessed processes personal data in certain phases of the PET (or at least conclude that this cannot be ruled out):

2023 Federated Machine Learning, Privacy-Enhancing Technologies, and Data Protection Laws in Medical Research: Scoping Review: "*The major identified problem is defining the GDPR status—personal or anonymized data—of which only the former is governed by the GDPR. We found that, in addition to the data themselves, the GDPR status of both local and global FL models is uncertain. Without DP and SMPC, local FL models should be considered personal data and, thus, need to be treated as such. Moreover, there is controversy as to whether DP and SMPC are sufficient to "anonymize" local models. Whether global models are personal data is also uncertain. Therefore, in general, it remains unclear whether FL achieves a level of privacy and security consistent with the requirements of the GDPR. Although FL systems do provide better security than centralized systems, they do not by themselves ensure a sufficient degree of anonymization and privacy to be considered GDPR compliant by design. Thus, even if global models are not to be considered personal data, the GDPR remains applicable to local models and model updates*."

2022 ICO guidance, Privacy-enhancing technologies (PETs): "*Are PETs anonymisation techniques? PETs and anonymisation are separate but related concepts. Not all PETs result in effective anonymisation, and you can achieve anonymisation without using them. At the same time, PETs can play a role in anonymisation, depending on the circumstances. For example, you can configure differential privacy methods to prevent information about specific individuals being revealed or inferences about them being made. However, the purpose of many PETs is to enhance privacy and protect the personal data you process, rather than to anonymise that data. This means that: • many PET use-cases still involve personal data; and • when you deploy such techniques, you still need to meet your data protection obligations*."

2021 Data protection by design in AI? The case of federated learning: "*Considering, on the one hand, the potentially broad interpretation of the notion of identifiability, and, on the other, the possibility of updates leaking the underlying training data as well as their (theoretical) vulnerability to property inference attacks, it cannot be excluded that, in certain specific settings, these updates may qualify as personal data. Should that be the case, the controller(s) responsible for the processing operations on these data will also have to ensure that the processing of model updates complies with the GDPR*." […]

"*As is usually the case with privacy preserving technologies, when considered in isolation, federated learning is no silver bullet. Although it can, under certain circumstances, help facilitate compliance with some data protection principles, it does not, as such, exempt organizations from the GDPR's application, especially if the raw training data qualifies as personal data*."

2021 Multi-Party Computation in the GDPR: "[…] *MPC only protects data during the computation but not the computation's output. We show that the output of an MPC could still be personal data, even in the relative approach. […] "Notwithstanding that MPC will protect the input data during the computation, one must be careful about the output. It could result in a transfer of personal data*."

2021 Musketeer: Benefits and challenges of federated learning under the GDPR: "*Also in this case, the key question is whether these updates can qualify as personal data. Our view is that this cannot be excluded, especially in light of (i) the potentially broad interpretation of the concept of personal data in general, and identifiability in particular, and (ii) the possibility of updates leaking information or being amenable to property inference attacks.*"

6. Processing Purposes

The purpose of processing the data via the vantage6 infrastructure depends on the specific collaboration / project for which vantage6 is applied. In general, this will involve gaining insights based on various data sources for (scientific) research or statistics.

7. Parties Involved

The **client** and the client's organization, i.e. the researcher or party benefiting from the statistics and research gained using the vantage6 infrastructure. The client files the data request. Generally, the client / data requesting party qualifies as a **controller** determining the purposes and means of the data processing.

The **vantage6 service provider** and **central service manager**.
The organization that provides the vantage6 service and manages the central server. By design, the central server only accesses results from computations of algorithms according to the agreements in the collaboration. For a discussion whether personal data is processed by the central server, we refer to the previous section. Based on this analysis, we conclude we qualify for the moment the Service provider as a "**Processor**".

The hosting party utilized by the **PHT Service Provider**. The status of the hosting party depends on the status of the PHT service provider.

The **data providers**, typically managing sensitive personal data

Generally, the data providers qualify as a '**Controller'**. They determine the purposes and means of the processing of personal data through PHT.

The organization hosting the PHT station for the data provider.
This organization generally qualifies as the '**Processor**'.

We assume that the trains are constructed and deployed according to the basic principle of the personal health train: no patient identifiable data is shared between parties. This is established by sharing either aggregated data (federated learning) or encrypted data (multi-party computation) with the central server.

4 7.1 Personal Health Train and Interpretation of GDPR

The report assumes that trains are accepted that share encrypted data with the service provider. The GPPR Article 29 working group has stated its opinion (5/2014) on several encryption

technologies. This working group assessed these techniques did not meet the three criteria for effective anonymization: person traceability, ability to connect data, and deductibility of personal details.

For the implementation of the Personal Health Train as discussed here, we argue that we *can* put measures in place that do meet these three criteria if a collaboration agrees on the right combination of algorithms, data and privacy settings.

9. The data stations are responsible for only accepting trains that do not disclose privacy sensitive data and guarantee effective anonymization. An on-going effort is to establish trust in these trains and empower organizations to review trains in a meaningful way.

10. Multi-Party Computation is a novel encryption paradigm that builds upon some of the 5 techniques as reviewed by the Article 29 WG. The techniques in development at the moment are “encryption with secret key” and “homomorphic encryption”.

11. Additional measures are put in place to guarantee anonymity of data.
    - Data minimization – no data is to be analyzed and placed on the data station that is not strictly required for the research question to be addressed
    - Random selection: instead of including all patients in a cohort, a random subsample can be used for analysis making it harder to infer which patients were included and which were left out of the analysis.
    - Differentially privacy: calibrated randomness can be added to an algorithm or query that processes sensitive data according to the definition of differential privacy, which provides mathematical guarantees that the output of the algorithm is resistant to any form of attack that attempts to infer which individuals are present in the input data.

We believe that these measures will further drive the discussion on the role of PETs and vantage6 in regard to the GDPR. We therefore consider this DPIA a living document that will be revisited on a regular basis.

8. Processing Locations

The locations where data is processed are as follows:

1. At (the data host of) the data provider
2. At the vantage6 data station - at the central server
3. At the PHT service provider

Using Federated Learning trains:
Trains are certified to only share aggregated statistics with the central server
No processing of individual patient data takes place outside the data station
The data providing parties are responsible for accepting trains on their stations. They will verify whether the train indeed does not share any identifiable information.

Federated learning is therefore suited for international collaborations, with data providers outside the EER. As no patient-level is shared across borders or organization, the GDPR is not applicable for as no sensitive data is processed outside the data stations. Of course, each data provider should adhere to GDPR when processing data.

Using Multi-Party Computation trains:

12. Trains will share encrypted data with the PHT service provider
13. Processing of encrypted patient-level data takes place at the PHT service provider, yet the service provider is unable to identify individual patients due to the state-of-the-art encryption techniques applied.
14. MPC techniques enable privacy as no single organization can decrypt data collected at the PHT service provider.

9. Techniques and Methods of Data Processing Operations

Trains are implemented to provide the functionality of statistical packages that are commonly used in data analysis projects.

The PHT service provider manages and certifies the trains, while the data providers are required to accept a journey including the trains required.

15. It is the responsibility of the PHT service provider to ensure that the train used in the journey is the same as specified at the moment when data providers accept the journey
16. The PHT service provider will make information available to review the functionality of the trains and test them in a controlled environment (e.g. with fake/synthetic data)
17. The data provider will accept the trains based on this information.

10. Retention Periods

In the data request, two periods will be defined:

18. The period in which the data will be available in the PHT station
19. The period the dataset will be retained at the organization as defined in the data request

A. Assessment of lawfulness of data processing

11. Legal Basis

It is important that all parties involved in a journey have a justification of lawfulness. Besides the criteria mentioned in article 6 GDPR (lawfulness) all parties involved need to also have an exception following article 9 GDPR to be able to process special categories of personal data (sensitive data).

For now, it is known that the PHT will be used in settings using sensitive data. It can be considered that for these situations article 6 under f GDPR can be invoked:

> *"processing is necessary for the purposes of the **legitimate interests** pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child."*

Mostly also article 9 sub 2 under j can be invoked:

> *"processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) based on Union or Member State law which shall be proportionate to the aim pursued, respect the essence of the right to data protection and provide for suitable and specific measures to safeguard the fundamental rights and the interests of the data subject."*

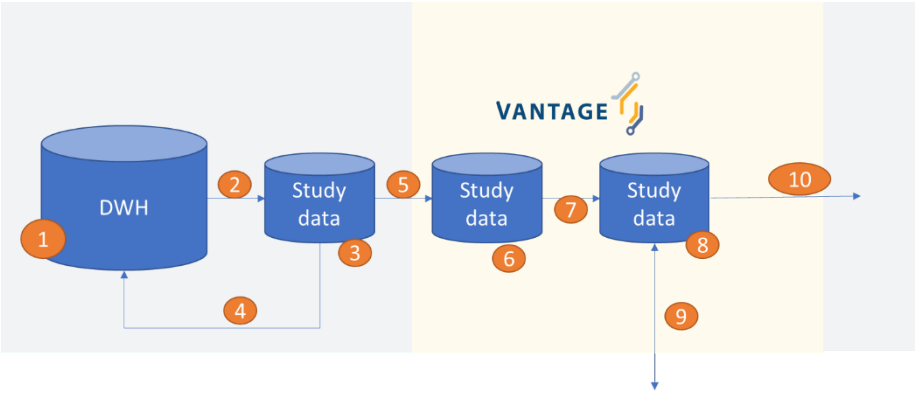
12. Special Categories of Personal Data

vantage6 aims to enable epidemiological research available in a privacy-preserving manner. The data involved may contain sensitive information.
It can contain personal data, genetic data and/or data concerning the health of individuals.

B1 Description and assessment of the risks for the data subjects

13. Risks

For this analysis, we deem only the steps 5 - 10 of Fig. 8 relevant, as the other steps are not specific to the PHT.



Ref. no.	Step	Risk type	Risk
1	5	Loss of confidentiality	Unsecure file transfer to Data Station
2	7	loss of confidentiality	Data provider accepting a journey not reflecting the data request
3	6,8	Loss of confidentiality	Hack on Data Station
4	9	Loss of confidentiality	Use of malicious Docker image after failed certification
5	9	Loss of confidentiality	Use of malicious Docker image after hack on the PHT service provider
6	9	Loss of confidentiality	Use of Docker image of malicious train accepted by data provider
7	9	Loss of confidentiality	Use of very small data set such that aggregated data contains identifiable data
8	9	Loss of confidentiality	Authentication not sufficient allowing undesired access to other party
9	9	Unauthorized or unlawful disclosure and/or processing	Client (e.g. a researcher) may use data otherwise than stated in the data request (e.g. commercial application) – risk is not specific to PHT
10	all	Unauthorized or unlawful disclosure and/or processing	Interception when data is transferred from one location/system to the other. (e.g. man in the middle attack)

11	5	Unauthorized or unlawful disclosure and/or processing	Too much data in dataset (e.g. dob delivered rather than age)
12	n.a.	Unauthorized or unlawful disclosure and/or processing	Lack of governance structure
14	n.a.		One of the nodes is slow or gets disconnected – research cannot be performed.

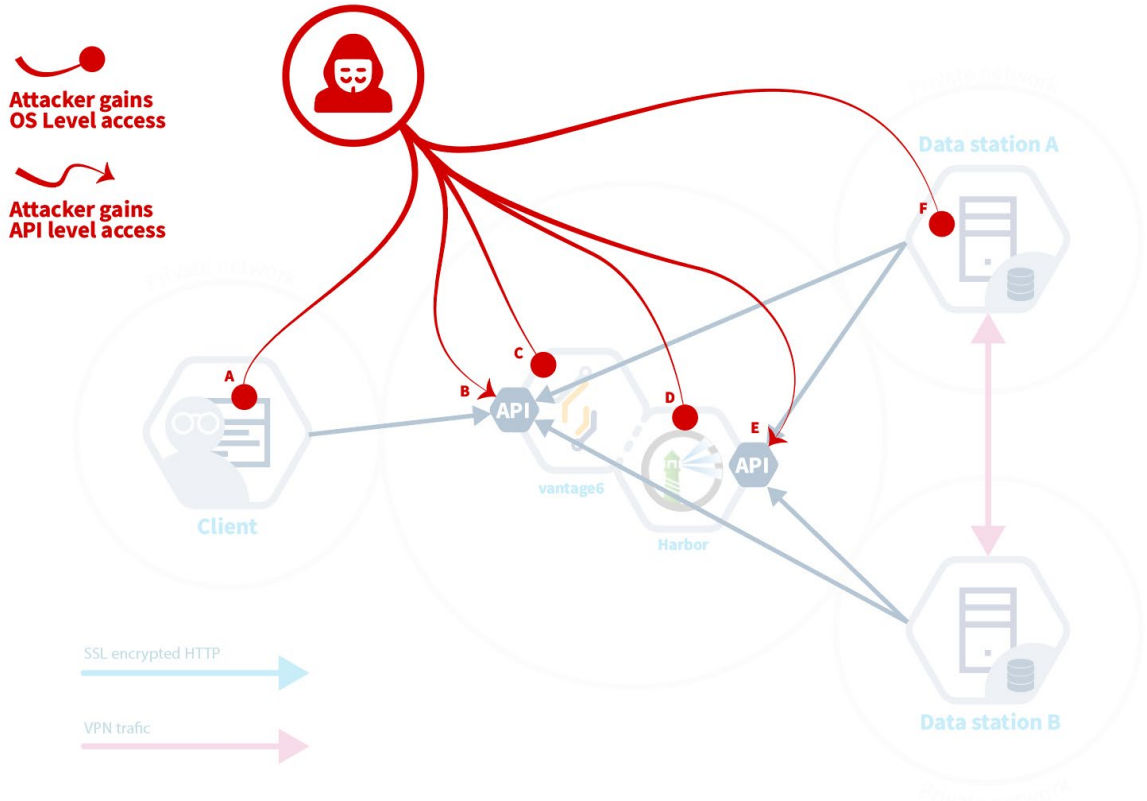


Figure 6 – Possible initial attack vectors on the vantage6 infrastructure. We can roughly categorize the attack vectors in two types. (1) Attacks aimed to gain OS level access, and (2) attacks aimed to gain API (application) level access. The attacker gains access to: A) the OS of the client computer. B) the API of the vantage6 server. C/D) the OS of the vantage6-server and depending on the network setup also to the Harbor registry. E) the API of the Harbor registry, (F) the OS of the data-station.

Risk	Impact	Measures	Hazard	Impact
Server API Access				
Credentials of user or root user are exposed	Attacker can log onto the server as user or root	Two-factor authentication can be enforced for all vantage6 logins	Unlikely	Moderate
Comprised Operating System at Client				
Credentials of user are obtained	attacker can log onto the client	Two-factor authentication can be enforced for all vantage6 logins. Sessions expire after 48	unlikely	moderate

		hours (default), with new login necessary		
Attacker obtains private key	attacker can inspect all previous algorithms and output	Two-factor authentication can be enforced for all vantage6 logins. Sessions expire after 48 hours (default), with new login necessary		

Risk	Impact	Measures	Hazard	Impact
Comprised Operating System at Server				
Attacker manages to login to the server	Read and modify database records, stop and/or uninstall the server	Two-factor authentication can be enforced for all vantage6 logins. Sessions expire after 48 hours (default), with new login necessary. API keys and passwords as these are stored in hashed form, so not accessible by the attacker	unlikely	moderate
Comprised Docker Registry				
Attacker manages to login to the server with the Docker Registry	The attacker can add a new docker container or remove an existing one	If the policies of the collaboration are defined, new/added docker containers will not be part of the collaboration and not be accepted. If a Docker Container is removed, the user cannot use the algorithm, but this does not pose a data protection risk.	unlikely	Low
Comprised EduVPN Server				
Attacker manages to login to the eduVPN server	The attacker can view the messages sent between parties in the collaboration.	End-to-end encryption can be applied such that the attacker cannot inspect the content of the messages exchanged.	unlikely	Low
Attacker manages to login to the eduVPN server	The attacker can shut down the VPN network.	No algorithms can be executed, but no data is leaked.	unlikely	Low

Comprised RabbitMQ Server				
Attacker manages to login to the server with the RabbitMQ server	The attacker can shut down the server	Only metadata is shared, no sensitive data is at risk	unlikely	Low
Comprised data station/node				
Attacker manages to log into the system hosting the patient-level data	Attacker can access the data source and can read patient level data	The risk of a successful attack is very low, as the node only requires a single outgoing port to operate. A hack can also take place inside the local network (subject of different DPIA) or the attacker is an employee of the organization (measures out of scope for this DPIA)	unlikely	Moderate
Algorithm-based Risks				
A malicious algorithm is added to the collaboration	algorithm does not perform (only) according to its specifications, but would (also) leak data	Code review of the algorithms (and verification of its authors) should be part of the process of accepting a collaboration. Reuse of algorithms across collaborations will build trust	unlikely	moderate

B. Description of measures planned

14. Measures

Ref. no.	Step	Risk type	Measures	Hazard	Impact
Loss of confidentiality					
1		Unsecure file transfer to Data Station	The data is stored on a secured server, making use of all modern web security standards including safe file transport between the data stations and servers.	unlikely	moderate
2	7	Data provider accepting a journey not reflecting the data request.	Will result in an unpredictable outcome or the train (algorithm) will not run on the dataset. The client responsible for the study will notice the discrepancy and take actions as the study aim cannot be achieved. In the current way of working, the PHT central server manager is responsible for the definition of the journey. The data providing organizations will review the trains before accepting the journey. As all peers (i.e. all data stations) review the journey, the implementation of the journey is not dependent on one reviewer from one organization, but is a shared effort and responsibility.	unlikely	moderate
3	6,8	Hack on Data Station	To use vantage6 on a data station, Docker and the vantage6 software need to be downloaded from the internet (vantage6.ai). The responsibility for downloading a correct version of the software is with the data providing organization. As it is open source, other, compatible versions yet with undesired functionality may be published on the internet. However, the source code of the installed software can always be inspected and reviewed. The data is stored on a secured Azure server, making use of all modern web security standards. Trusted users review usernames and passwords Future: disable accounts that are not used for 30 days. Log logins and notify the Data Protection Officer when suspicious logins occur. Log files of vantage6.ai will be shared with data Station organizations to review data traffic. Authentication, encryption, and security policy will be published and reviewed by the security officer. Said policy will be regularly updated and reviewed. With the PHT, we address this problem by placing the sensitive data on a secure server including a firewall. In PHT projects today, we use limited datasets. Log files of vantage6.ai will be shared with Node organizations to review data traffic. Authentication, encryption, and security policy	unlikely	moderate

			will be published and reviewed by the security officer. Said policy will be regularly updated and reviewed.		
4	9	Use of malicious Docker image after failed certification	In the current way of working, the coordinator (in the role of PHT central server manager) is responsible for the definition of the journey, including the selection of Docker containers. The data providing organization defines the Docker containers that are accepted on their stations. If a container is accepted that is not certified, this container may conduct analyses or induce communication that is not specified. This behavior can be observed also when analyzing synthetic data. Data partners may therefore wish to first evaluate the behavior of Docker containers on synthetic data, such that no sensitive data is exposed at the first usage of the container.	unlikely	minor
5	9	Use of malicious Docker image after hack on the PHT service provider	See 4 use certificates and standard safety-measures on their infrastructure and monitors where applicable.	unlikely	minor
6	9	Use of Docker image of malicious train accepted by data provider	Each data provider (station) is responsible for their own infrastructure. However, in the current way of working the study cooridnator is responsible for the definition of the journey, including the selection of Docker containers. Future: when other parties make algorithms available, the central server manager will (a) review the code by 2 data scientists, (b) publish the review on the GitHub page where the code is stored and (c) test the data communication using the algorithm on synthetic data to detect possible data leaks	unlikely	minor
7	9	Use of very small data set such that aggregated data contains identifiable data	Data requests need to be evaluated as they are today for “normal” requests. If a data set is too small, then take corresponding measures and establish a minimum number of patients to process in the collaboration’s privacy settings.	unlikely	minor
8	9	Authentication not sufficient allowing undesired access to other party.	The central server manager hosts the authorization of users and thereby the access. No access is granted before all the necessary legal steps have been taken between the partners.	possible	moderate
Unauthorized or unlawful disclosure and/or processing					
12	n.a.	Lack of governance structure	Current measures: file for separate data requests at participating data providers and make all software open source to provide full transparency.	unlikely	minor

			Future measure: identify (semi-)trusted third party to play the role as Central Server manager. and define contracts between data providers and Central Server Manager. A workflow should be defined and coordinated to execute studies with multiple data providers (stations) in order to adhere to the applicable data protection, ethics and privacy measures.		
14	n.a.	One of the nodes is slow or gets disconnected – research cannot be performed.	Measures are not necessary, this will result in delay or postponing of the study. This is not different from the normal procedures when performing scientific studies.	unlikely	moderate
15.	NA	Algorithm-based risks of identification	- Add differential privacy in order to create noise to the output of the algorithm. Accuracy of the result is compromised, in particular in analyses with small data sets - Perform k-anonymity filter. For the data items to be considered (quasi) identifying, at least k subjects with the same value (e.g. age, gender, date of diagnosis) should be included in the data set to be processed by the algorithm	unlikely	moderate
16.	NA	Correct use of algorithms	- If a setup is chosen in which the central server is also the party where the data station is located, this will lead to possible identifiability of data because the data station also receives data from other stations, which poses a privacy risk. Will be particularly relevant in studies with vertically partitioned data, MPC and secret sharing, certain protocols mainly MPC. - Solution: each party must be independent. Data station and central server must not reside at the same party.	unlikely	moderate

C. References

1. Arturo Moncada-Torres, Frank Martin, Melle Sieswerda, Johan van Soest, Gijs Gelijnse. VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange. AMIA Annual Symposium Proceedings, 2020 (in press).

2. IKNL. vantage6.ai – Privacy preserving federated learning [Internet]. 2019 [cited 2020 Jan 24]. Available from: https://www.vantage6.ai

3. Nina Bontje. Privacyaspecten van de Personal Health Train Aandachtspunten voor de verdere ontwikkeling. 2018.

4. Jones EM, Sheehan NA, Masca N, Wallace SE, Murtagh MJ, Burton PR. DataSHIELD – shared individual-level analysis without sharing the data: a biostatistical perspective. Norsk Epidemiologi [Internet]. 2012 Apr 13 [cited 2019 Jan 22];21(2). Available from: http://www.ntnu.no/ojs/index.php/norepid/article/view/1499

5. Jiang W, Li P, Wang S, Wu Y, Xue M, Ohno-Machado L, et al. WebGLORE: a Web service for Grid LOgistic REgression. Bioinformatics. 2013 Dec 15;29(24):3238–40.

6. Lu C-L, Wang S, Ji Z, Wu Y, Xiong L, Jiang X, et al. WebDISCO: A web service for distributed cox model learning without patient-level data sharing. Journal of the American Medical Informatics Association. 2015 Jul 9;ocv083.

7. Lee J, Sun J, Wang F, Wang S, Jun C-H, Jiang X. Privacy-Preserving Patient Similarity Learning in a Federated Environment: Development and Analysis. JMIR Medical Informatics. 2018 Apr 13;6(2):e20.

8. Chang K, Balachandar N, Lam C, Yi D, Brown J, Beers A, et al. Distributed deep learning networks among institutions for medical imaging. Journal of the American Medical Informatics Association. 2018 Aug 1;25(8):945–54.

9. Marte van Graafeiland, Nina Bontje. Toepassing van de Personal Health Train in de zorg Verdiepend onderzoek. Pels Rijcken; 2020.

10. Veeningen M, Chatterjea S, Horváth AZ, Spindler G, Boersma E, van der Spek P, et al. Enabling Analytics on Sensitive Medical Data with Secure Multi-Party Computation. Stud Health Technol Inform. 2018;247:76–80.

11. Djura Smits, B. v.-T. (2022). An Improved Infrastructure for Privacy-Preserving Analysis of Patient Data. Proceedings of the International Conference of Informatics, Management, and Technology in Healthcare (ICIMTH), (pp. 144-147). Athens, Greece.

B. Security and Privacy Note

C.Introduction

The aim of this document is to provide a structured way to determine the risk when using the vantage6 infrastructure for your projects. Vantage6 is a privacy-enhancing analysis infrastructure that allows collaborators to semi-automatically deploy algorithm networks [1]. It is designed to minimize -but does not eliminate- the risk of exposing record-level data while analysing data without bringing the data together in a central location.

The risk of exposing data is partially dependent on the requirements of your project, e.g. which data will be used or which algorithms will be executed and how often. This document describes the risks for the general use case and makes no assumptions on project specifics. It is a starting point to evaluate the risk of your specific project.

This document is structured as follows. First, the remainder of this section introduces the basics of risk models and the basics of the vantage6 infrastructure. Section 2 discusses the risks of breaches in the vantage6 infrastructure and their implications. Next, section 3 lists which attacks are generally possible for federated algorithms and how the risks related to them may be minimized. Finally, in section 4, we provide guidelines to help identify risks for a specific project and how to mitigate them as much as possible. These guidelines should aid in creating a security document specific to your project.

1. Risk Model

In a Data Protection Impact Assessment, risks around events that impact the protection of sensitive data are assessed. Here, a risk may be defined as:

$$risk = likelihood \times impact$$

where likelihood is the probability the event occurs, and impact is the severity of the consequences when the event occurs. The impact and likelihood are scored between 1-5. Impact is scored from Negligible to Severe and likelihood is scored from Rare to Almost Certain (Figure 1).

Likelihood and Impact Model

			Likelihood				
			Rare	Unlikely	Moderate	Likely	Almost Certain
			1	2	3	4	5
Impact	Severe	5	5	10	15	20	25
	Major	4	4	8	12	16	20
	Moderate	3	3	6	9	12	15
	Minor	2	2	4	6	8	10
	Negligible	1	1	2	3	4	5

Figure 1 – Risk matrix to classify the risk for certain events. Scores equal or greater than 15 are considered high risk, between 4-15 are considered medium risk, and below 5 is considered low risk.

Prevention is defined as taking the appropriate measures to reduce the *likelihood* of an event. For example, enabling two-factor authentication (2FA) makes it less likely that a hacker gains access.

Mitigation is about reducing the *impact* when the event happens. For example, by encrypting stored data, the impact of unauthorized access would be lower.

2. Vantage6 Infrastructure

Vantage6 uses a client-server and peer-to-peer network model [2], which is shown in Figure 2. The traffic between the central server and the clients and data stations is all SSL encrypted HTTP (HTTPS). The peer-to-peer networking between data stations proceeds over a VPN network. This component is optional but required by some algorithms.

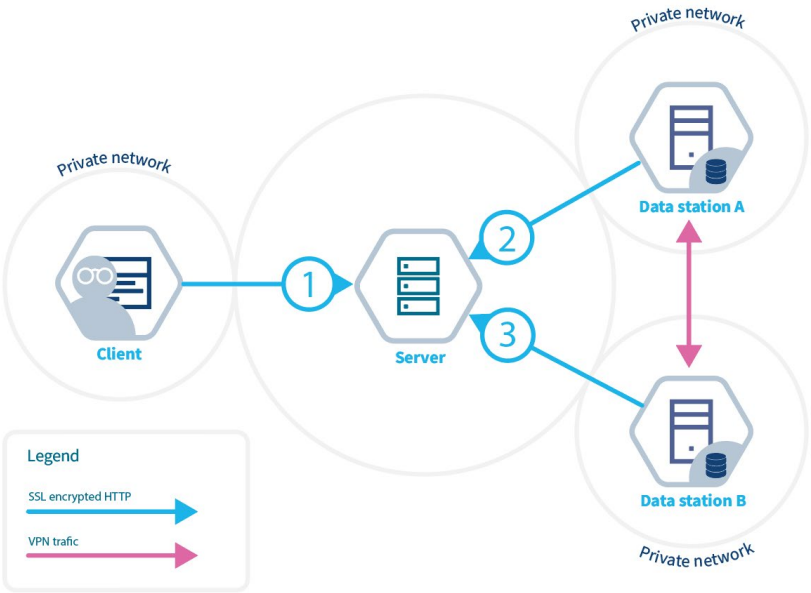


Figure 2 – High-level overview of the vantage6 infrastructure. The VPN connection is an optional feature of the infrastructure and is only required for certain algorithms. The client and data-stations connect with the server and typically only require outgoing port 443 to be open. The node collects information from the server through a pull mechanism. In other words, the

node is always the initiator of the communication and there is no way for the server to connect to the nodes. This protects the node as there are no entry points to the data-station that a potential attacker can use. The server is likely to be a public endpoint to which the other components can connect.

There are three major types of components in the vantage6 network:

(1) **Client**. The client can be a user or an external application that connects to the server to initiate an analysis. Client applications that are provided by the infrastructure developers are the Python client, the user interface (UI), and the R client. Users can also connect to the API of the vantage6 server directly, or they may even create their own client application using any programming language.

(2) **Data station**. The data station is connected to the local data source containing the privacy sensitive data. It is responsible for executing the algorithm and returning the results to the server.

(3) **Server**. This should at least contain an instance of the vantage6 server. The vantage6 server is the central hub that receives computation requests and stores their results. It also manages organizations, collaboration and users.

3. Additionally, there are optional components that may be required in a specific project. The first is a Docker registry, which is a place to store algorithm software securely. Secondly, an EduVPN instance, which is required to enable the peer-to-peer network feature. Finally, a RabbitMQ service to improve performance of the vantage6-server in case of high workloads. Encryption

Communication between data stations and between data stations and clients go through the vantage6-server. Task input and their result can be end-to-end encrypted. In this case, they are stored encrypted at the central server and can only be read by the intended receiver. In the current version of vantage6 the end-to-end encryption is between organizations. So all nodes and users within a single organization use the same private key to receive messages. In a future release of vantage6 this will most likely be handled at data station/client level.

4. Vertical and Horizontal Data Partitioning

In vantage6, we distinguish two types of data partitioning: horizontal and vertical [1]. When the data is horizontally partitioned, the data stations collect the same data items for different sets of subjects. In case of the vertical-partitioned data, the data stations collect distinct data items for the same set of subjects.

Vantage6 supports both cases, but they require different types of algorithms. In the horizontal case, usually FL algorithms are used and in the vertical case, MPC is typically used.

5. Federated Learning and Multi Party Computation

Currently we distinguish two types of algorithms: Multi-Party Computation (MPC) and Federated Learning (FL). In Federated Learning, algorithm mathematics are separated in a federated and central part (also commonly called the aggregator). For example, we want to compute the average of vector $\vec{x} = \langle x_1, \dots, x_n \rangle$. In a central analysis this would be:

$$\underline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

In the federated analysis the vector $\vec{x}$ would be distributed amongst at least two parties. Lets consider two parties A and B: $\vec{a} = \langle x_0, \dots, x_k \rangle$ and $\vec{b} = \langle x_{k+1}, \dots, x_l \rangle$ where $n = k + l$. Then the average is computed as:

$$\underline{x} = \frac{1}{k_{\smile party\ A} + l_{\smile party\ B}} \left(\sum_{i=1}^{k} a_i^{\smile party\ A} + \sum_{j=1}^{l} b_j^{\smile party\ B} \right)$$

Now only the number of observations k, l and the sum of each vector need to be shared. These can then be combined to a global average as shown in the equation above. This process can be repeated for a variety of algorithms (e.g. GLM, CoxPH), however mostly for the horizontal case. In some cases it is not possible to decompose the algorithm in a FL algorithm, this is almost always the case in the vertical scenario. For these scenario's Multi-Party Computation can be a solution [3].

6. Algorithm Containers

The infrastructure enables users or other applications to build MPC and FL networks. Data stations execute tasks in order to participate in these networks. For example if we look at the average example from section 1.3, all participants compute the number of elements and the sum of the vector of interest. These tasks are predefined and stored as (algorithm) containers in the Docker registry. The nodes can retrieve and then execute these containers to compute the required results.

Containers can be viewed as an cross-platform (Windows, Linux, etc.) executable package that contains everything to run the task. This includes the code, their dependencies, runtime, system tools and libraries. These containers are easily shared and executed on different hardware and operating systems.

7. Algorithm Data Flow in vantage6

In section 1.2, the three main components of vantage6 are explained and in section 1.3 a simple federated algorithm to compute a global average is explained. In this section, it is explained how algorithms are created and executed within the vantage6 infrastructure.

The simple average algorithm can be schematically displayed (Figure 3). There is a central task that is responsible for aggregating and a subtask that handles the computation of the partial result. In case of the average, the partial result is the number of observations and sum of the vector. In vantage6, the central task is also responsible for orchestrating the algorithm. In other words, the central task is responsible for creating the subtasks and collecting their results. Tasks and subtasks are run within a container.

The schematic representation of the algorithm can be projected on the infrastructure, shown in Figure 3.

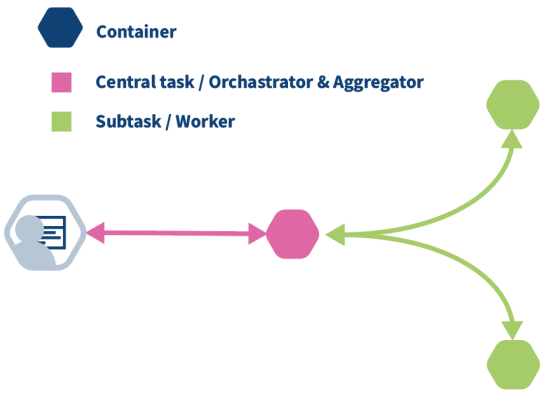


Figure 3 - A schematic representation of the federated average algorithm. The central part, initiated by a user, is responsible for orchestration and aggregation of the partials. The subtask is responsible for computing the length of the vector and the sum of the vector. All the tasks and subtasks are run within containers. Note that for more complicated algorithms, there may be a more complicated flow: for instance, there may be multiple iterations of the subtask, or algorithms may communicate over a peer-to-peer network.

Figure 4 – Data flow projected on the infrastructure for a simple non iterative algorithm with an orchestration and aggregation part. The task is initiated by the client, which stores a task record at the server. The client also assigns which of the data stations should start this task – note that the

central task is run on a data station and not on the central server. In this case, the central task is executed by Data station A. The central task creates subtask (orchestration) records on the server, which are then picked up by each of the data stations participating in a task. When the subtasks are completed, their partial results are stored at the server. Finally, they are combined by the central task which uses them to compute the global result (i.e. aggregation). The global result is then stored at the server from where the client can access this global result. The partial and global results are stored at the central server until the user deletes them. The central task requires all partial results to compute the global result. Therefore all the partial results are also stored at the data-station that handles the central task. Note that in the current version of vantage6 these are kept indefinitely, which is useful for debugging but might not be desired in a production use case.

In general, we separate three types of data transfer:

(1) Record level data: data used for computation
(2) Aggregated data: outputs of algorithms
(3) Metadata: task description, task status, etc.

Data description	**Component(s) that have this data**	**Data type**
Task metadata. This data allows vantage6 to execute the task. It consists of a name, description, reference to a container image and input data. It also includes timestamps when certain steps have been executed in the algorithm.	Client Server (vantage6-server)* Data station	Metadata
Algorithm containers. Contains the algorithm code and all its dependencies.	Server (Docker registry) Data station	Algorithm
Aggregated data. Output from the algorithm containers. E.g. model beta's, aggregated statistics.	Data station Server (vantage6-server)* Client	Aggregated data
Patient data. Sensitive data typically record level data of patients	Data station	Patient data

Record level data is never shared between the components and always remains at the data station. Aggregated data is shared between algorithm containers and the client. Metadata is mainly used in the system to initiate tasks. The types of data are summarized in Table 1 and the flow of the different types of data is summarized in Figure 4.

Table 1 – Which data of the analysis is stored where. (*) End-to-end encrypted and is therefore only readable by the receiver. For example the task input is only readable by the node that needs to execute the task.

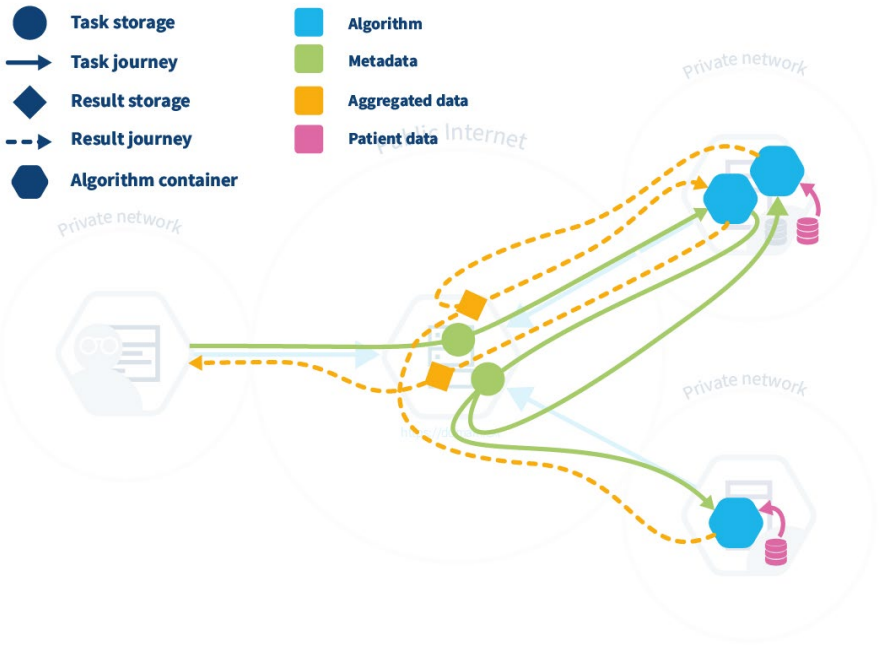


Figure 5 – Algorithm data flow per type of data for a typical simple vantage6 algorithm. Record(/patient) level data is never shared between the components. Note that the storage icons in this figure refer to storage in a relational database, however everywhere a journey starts or ends this data is also stored.

It is important to note that the infrastructure does not verify that the output of the algorithms is aggregated data. This is the responsibility of the algorithm. Therefore, it is important to review and validate the algorithm before allowing it to run on your data station.

8. Collaborations and Policies

In vantage6 a collaboration consists of one or more organizations. Within a collaboration certain agreements (policies) with regard to the infrastructure need to be made, for example:

- Which algorithms are allowed, and what are the privacy settings (e.g. minimal number of patients after selection)
- Is the communication encrypted, see section 1.7
- Execution policies (e.g. which users and organizations can initiate the algorithm)

Once the policies are accepted by all parties, the data-station owners can enforce these rules locally. This way, a change at the server does not expose the data-stations to altered policies. In other words, the data-station owner controls what policies are enforced at their node. This also means that if the collaboration agrees on a new policy, effort of the data station owner is required.

D. Infrastructure Risks

In this section, the risks of attacks on the vantage6 infrastructure are described. It describes how attackers may try to obtain data, what data they obtain if the attack succeeds, and how the risk may be reduced.

The most important attack on the vantage6 infrastructure is an attack aimed at disclosing record-level data from the data stations. This could be attempted via several attack vectors (Figure 5).

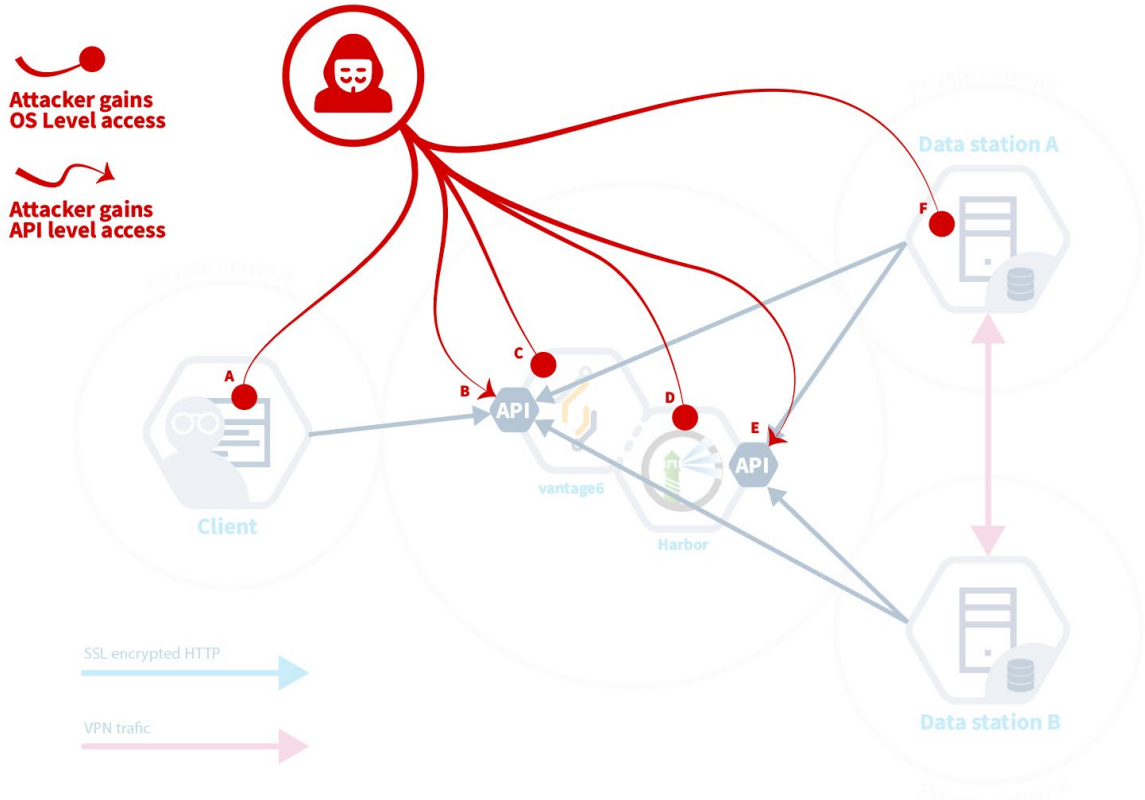


Figure 6 – Possible initial attack vectors on the vantage6 infrastructure. We can roughly categorize the attack vectors in two types. (1) Attacks aimed to gain OS level access, and (2) attacks aimed to gain API (application) level access. The attacker gains access to: A) the OS of the client computer. B) the API of the vantage6 server. C/D) the OS of the vantage6-server and depending on the network setup also to the Harbor registry. E) the API of the Harbor registry, (F) the OS of the data-station.

These attack vectors can be grouped as:

(1) Gain access to operating systems hosting the vantage6 components (client, server, etc), through: A, C, D, F.
(2) Gain access to the vantage6 server API, through B.
(3) Code injection into algorithms, through D or E.
(4) Use (intermediate) results and to reconstruct or derive record level data, through B or C.

Note that once the attacker gains access to some component, he or she might be able to gain access to a secondary component more easily. In the next subsections we will describe each of the attack and possible secondary attack possibilities.

1. Vantage6 Server API Access

Access credentials can be obtained with brute force methods or they may be leaked. A hacker who obtains API access can execute certain operations on the server, depending on the permissions of the compromised user account.

- Modify users, organizations and collaborations
- Send tasks and read their results (after replacing the public key)
- Read results of past analyses, if they also have access to the private key or if the task was not encrypted
- View the usernames of other vantage6 users and their permissions. This may help them attack user accounts with different or higher permissions.

In the worst-case scenario, a root user (i.e. which has all permissions) is compromised. In this scenario, the attacker is able to send tasks to any collaboration but is limited to algorithms that are allowed on each collaboration or node.

If two-factor authentication is enforced in combination with a strong password policy such a breach is unlikely. Also, the attacker is unlikely to gain access to other components from this attack vector.

2. Compromised Operating System

Here we will describe what the risks are of gaining access to the Operating System (OS) of machines running vantage6 software. Breaches could occur due to:

- A vulnerability in the OS. For example, the attacker might use the Log4J vulnerability to obtain credentials [4].
- Human error. For example, a user has a weak password or stores their password in plain text.
- Poorly secured system. For example, a machine with no firewall or with open ports that need not be open.

The machines running the following software are vulnerable to an attack:

- Client
- Vantage6 server
- Docker registry

- EduVPN
- RabbitMQ
- Data station

In the next sections, we investigate the implications of breaches in any of them.

2.1 Client

This is usually the machine of a researcher that has a user account on a vantage6 server. Other possibilities are for instance a server running another application that creates vantage6 tasks.

Depending on the permissions this client has, it can manage users, organizations, collaborations, create tasks and obtain their results. If an attacker gains access to the system, they may:

- Obtain the username and password of the vantage6 user whose machine they have taken over. Note that this is only the case if the user has stored their username and password in plain text on their machine.
- Take over an active connection with the server using a refresh and/or access token.
- Obtain the private key of the organization. This key may be used to read previous results and inputs at the vantage6 server

If two-factor authentication (2FA) is enabled at the vantage6 server, the loss of a username/password combination does not directly lead to access to the vantage6-server API. This would require the attacker to also obtain access to the second device and their security app.

If an active session to the vantage6 server is ongoing, the attacker gains access to the API (see section 2.1 for the implications). If the attacker also obtains the organization's private key that is used in the vantage6 infrastructure, or if the collaboration is not end-to-end encrypted, they can read all previous algorithm results and their input. Depending on the algorithm, this may provide a way for an attacker to reconstruct record level data, see section 3.

In vantage6, by default, sessions expire after 48 hours. An attacker may refresh the token to extend their session. If the token is not refreshed in time, the attacker would have to re-authenticate – which ends the attacker's session unless they have user credentials. It is possible to lower the session expiration time to reduce the likelihood that an attacker takes over an active session.

2.1 Vantage6 Server

The vantage6 server should be approachable by clients and data stations. To attain this, the server typically has a public IP address, though other configurations are possible. When an attacker gains access to the OS of the vantage6-server they may:

- Read and modify database records
- Stop and/or uninstall the server

Modifying database records can result in a broken database, rendering the server unusable.

The attacker might also gain access to the API, since they may obtain the server secret which is used to generate secure API keys. They can use this to create a root user account or to access the API impersonating a node or algorithm.

The attacker cannot read existing API keys and passwords as these are stored in hashed form. They are also not able to read any previously created results, unless the task was not encrypted or if the attacker also has access to a private key of the participant of whom the results are to be red.

Even though the server usually has a public or semi-public interface, such a breach is very unlikely if the server has been deployed in a secure manner and is properly maintained. Attacking the server would require breaking through several layers of security. For example, a server administrator should configure the SSH port to only be available through a tunnel or at least be limited to be approachable by certain IP addresses.

In case the Harbor registry, VPN server or RabbitMQ is hosted at the same (virtual) machine, the hacker also gains access to these, see section 2.2.3, 2.2.4 and 2.2.5.

2.1 Docker registry

Algorithms may be stored in a Docker registry. In case a hacker gains access to this system there is no immediate threat to the parties involved. He or she may:

- Remove algorithms
- Shut down de service
- Upload new algorithms

If the attacker removes an algorithm it can no longer be retrieved for computation. Similarly, algorithms can no longer be retrieved if the attacker shuts down the service completely. While this may be disruptive, it does not lead to data leakage.

Assuming that the nodes are configured with the proper policies, the upload of a new algorithm to the repository should not pose a threat.

2.1 EduVPN server

The EduVPN service is required for the peer-to-peer communication between algorithms. This feature is not required by all algorithms. The traffic between the algorithm containers (running at different nodes) is directed through the EduVPN instance. The traffic between the algorithm containers and the EduVPN instance is encrypted using TLS. However, an attacker inside the EduVPN server would not be hampered by this as the data is unencrypted there. This would enable the attacker to read the messages between the parties. The impact of this breach can be reduced by only using algorithms with an additional end-to-end encryption layer. In that case, no data would be leaked: the attacker would only be able to see that messages are being sent.

The attacker may also shut down the EduVPN server. When this is done, it will not be possible to execute algorithms that rely on peer-to-peer communication. However, no data is leaked.

2.1 RabbitMQ server

RabbitMQ is an optional component of the server. It is only required when multiple instances of the vantage6 server are run. Running a single server instance may be sufficient for small collaborations, but for collaborations with many parties and/or many tasks, scaling is usually needed.

Only metadata is shared through RabbitMQ, for example a status message or node configuration details. Therefore, a hack could only cause disruption (e.g. by shutting down the service) and not leakage of any sensitive data.

2.1 Data station/Node

The node has access to the local data source which is required for the computations. If a hacker gains access he or she can:

- Access the data source and read record level data
- Stop the node
- Gain a node API key to access the server
- Gain access to the VPN network

Depending on the data source type it might be very easy to obtain record-level data. For example, in the case of a CSV or Excel file, the hacker only needs to locate the file on the system. If a more complex data type is used, OMOP for example, the attacker needs some basic SQL knowledge to

obtain the data. The attacker may also be able to change or compromise the data, leading to incorrect results of any subsequent algorithm executions.

Each node has an API key to authenticate to the vantage6 server. This key has limited permissions on the API. For example, the node has access to the tasks assigned to them and some metadata regarding the collaboration it operates in.

The VPN network is optional (see Section 2.3). If enabled, an attacker could gain access to this network. This would open possibilities to hack algorithm containers from other data stations in the network. These containers have access to their local data, potentially exposing other parties at risk too. Such a hack is highly unlikely as the attacker first must gain access to the node, take over the VPN network, wait for an algorithm network to be setup, obtain their ports and IP's through the API and finally there needs to be a vulnerability in the algorithms containers itself that the attacker can make use of.

When a node is set up properly, a hacker is very unlikely to gain access. The node only requires a single outgoing port to operate, making it nearly impossible to hack from the outside. A more likely scenario is that the attacker already infiltrated another machine in the local network of the data station or is an employee of the organization hosting the data station.

3. Code Injections into Algorithms

Vantage6 uses Docker images to store and distribute algorithms. An attacker can try to inject and hide a malicious piece of code into an algorithm image and upload it to the registry. Potentially privacy-sensitive data could be leaked to the vantage6 server or to another party (most likely the hacker himself) in case VPN is enabled. The attacker would not be able to copy the record-level data over the internet, because the algorithm containers do not have internet access. The attacker would need to:

- Create an algorithm
- Inject malicious code
- Upload it to the registry (requires access to a registry)
- Execute the algorithm (requires access to a collaboration)

Additionally, the data stations should allow this algorithm to be run – they can define policies to define which algorithms are allowed and which are not.

When using an algorithm in your collaboration you need to trust it. You can do this by trusting the author or executing your own code review. Doing your own code review can be challenging as the attacker might have hidden the code very well. In section 5, a solution is described to make this process easier by limiting the code to be reviewed.

E. Algorithm-based Risks

This section describes which risks are potentially relevant when using federated algorithms. These risks are not specific to vantage6, but apply to any FL infrastructure. Also, we only consider attacks here that lead to leakage of record-level data. Attacks via algorithms that lead to aggregated data leaks or interfere with analyses (e.g. modelling) are outside the scope of this document.

Several different types of attack are possible within a FL network that would allow someone to reconstruct record-level data from aggregated data. Several recent academic publications [5, 6, 7, 8] describe a variety of attack methods. In this document we only consider attacks that could lead to reconstruction of patient-level data:

- Reconstruction
- Differencing
- Deep Leakage from Gradients (DLG)
- Generative Adversarial Networks (GAN)
- Model Inversion
- Watermark attacks

This list has been carefully constructed but it is not exhaustive. There might be types of attack that have not been discovered, have not been made public or have not yet been found by us. Note that privacy-enhancing technologies in general and federated-learning systems in particular are active topics of scientific research.

Not all of the described attack methods are relevant to every project. It depends on the algorithms and the type of record-level data being used in a project to determine which of the risks apply. Also, some algorithms might have a small risk for a certain attack type whereas other algorithms run a larger risk.

The research question determines both which data and algorithms are required. To minimize the risk, it is advised to limit the number of algorithms and only authorize usage of those that are required to answer the research question. Each algorithm's output inherently leaks some

information about the record-level data. Hence, limiting the number of authorized algorithms reduces the chance that a combination of aggregate statistics could be used to single-out an individual's value in the underlying data.

On the other hand, a larger number of data subjects in the record-level data at each data station reduces privacy risks as well. In this case, algorithm results (Aggregate data) will be based on larger groups of people and thus less dependent on specific individuals being part of the data, reducing the risk of them being singled-out.

Since the risk depends heavily on the research question, which determines the types of algorithms and data that are required, it is not feasible to give a single answer to what the risk of a research project using vantage6 is. Instead, we give general information below on how the impact and likelihood of security breaches in federated analyses can be limited.

2.1 Impact

The potential impact of breaches in a federated analysis is that record-level data are leaked. The severity of the impact is then largely determined by which data are used in the project.

Depending on the research question, there are several ways in which the potential impact may be reduced:

- Add noise to the dataset (e.g. through differential privacy). Adding a calibrated amount of noise to each computation that is approximately equivalent to any possible individual datapoint in the underlying dataset to mask whether they were part of the computation. This comes at the cost of the accuracy of the final result, especially when the dataset is small [9]. Some algorithms are more sensitive to this than others.
- K-anonymity filters. K-Anonymity is achieved if there are at least *k* individuals for every set of quasi-identifiers. For example, 2-anonimity (*k*=2) on a data column containing the disease type of patients means that every disease type occurs at least twice. Larger *k* leads to stronger levels of privacy, as individuals can hide in the crowd.

2.1 Likelihood

There are several factors that may influence the likelihood of leaking record-level data:

- Role of the attacker in the analysis (participant/aggregator)
- Knowledge needed by the attacker
- Scale of the project

- Access to the system
- Which algorithms are used

Each type of attack requires a specific position in the system. In an algorithm, there are two positions an attacker can fulfill:

(1) The participant is someone that provides data (i.e. provides a node) but is also capable of asking questions to the system (i.e. uses the client).
(2) The aggregator is the party that combines the partial results to a global result. This can be an iterative procedure to find the optimal solution for the global model. It is also possible that the aggregator is a participant in the same computation.

Differencing-, model inversion- and watermark attacks may be executed by any participant. The aggregator may also execute these attacks, and additionally they may perform reconstruction-, DLG-, and GAN attacks. The aggregator position can be fulfilled by anyone in the system as it is not dependent on local data. Therefore, choosing a trusted party to fulfill the aggregator position would reduce the likelihood of such an attack happening. In that case, DLG and GAN can only be executed by the trusted party.

For some types of attacks, knowledge of the vantage6 system and algorithm are required, and sometimes they also require systems to be built around vantage6. In other cases (e.g. in a differencing attack), executing an algorithm twice can already lead to record-level data leakage (e.g. when only a single patient has been added to the dataset between two algorithm runs). Again, the likelihood of these attacks depends on the research question, as that determines which data and algorithms are available.

Only registered users assigned to the collaboration are participating. This reduces attack likelihood since access is restricted to the participating organizations and each participant is known by name and organization. As a result, the level of trust and accountability within a collaboration (system) increases. If there are more participants in a collaboration, the likelihood of including a malicious party obviously increases.

The likelihood of attacks also decreases if the attack can be traced more easily. Some, but not all attacks are traceable. Logging in vantage6 enables system administrators to view what every participant has requested and contributed to the analyses. When a participant purposely destabilizes the convergence of the model, logging can expose them. On the other hand, linkage attacks are hard to trace if the malicious participant has obtained additional information from

outside the system (e.g. the attacker already knows that only one female is included and then requests the average age per sex).

In conclusion, the likelihood of an attack that compromises patient record level data depends on the trust in your collaboration partners (e.g. you have collaborated before, they are from respected institutions, etc) and the algorithms that you use to answer your research question, as not all attacks are possible with any algorithm.

F. Examples of Attacks

This section is to give some intuition for possibilities to attack in order to obtain privacy-sensitive data.

1. Example: Internal Attacker

If set up properly, it is impossible to reach the node from an external network. Therefore an attacker either is part of the organization (but unauthorized) of the data station or gained access to the internal network through another system.

In order to gain access to the sensitive data, The attacker should achieve all of the following:

1. Gain access to a machine which can reach the data station machine. The number of machines that can reach this machine should be extremely limited as a firewall should only allow certain IP addresses.
2. Find the access credentials for the data station. This should be a private key. In case the hacker has hacked a machine, the private key might be at the hacked machine. However, these private keys should be password protected.
3. In the unlikely (due to human error) event that the attacker obtained a readable private key they still need a password to access the machine.
4. Once the attacker gains access to the data station, they can access the data. Depending on the type of data source they need to perform some additional work. In a CSV file the data can directly be read, but in the case of a (external) relational database the attacker needs to query it. The data source might be password protected, but the access credentials are not encrypted stored at this machine (as we need them at runtime). Even though this might delay the exposure it is unlikely to stop the attacker at this point.

2. Example: API breach

The attack targeted the API and managed to gain access to the API. Either by stealing user credentials and the device used for two factor authentication, or more likely (but still very unlikely) making use of an exploit in the API. The attacker needs to:

- Gain access to an account with sufficient privileges to execute algorithms in the targeted organization or collaboration
- If the node has been properly set up, the attacker can only send pre-approved algorithms. They are limited by these algorithms but might be able to obtain patient level information:
    - by executing smart queries and/or when less secure algorithms are accepted in the collaboration
    - by placing themselves in the aggregator position and using one of the attacks described in section 3 . However this only works if the collaboration accepts algorithms which are sensitive to these types of attack

3. Example: Leaked Credentials

See section 4.2, as this leads to an API breach.

4. Example: Simple Reconstruction Attack

The attacker attempts to reconstruct patient level data from aggregated results. The attacker is most likely someone who has authorized access to the collaboration but could also obtain access by other means.

The hacker needs:

- One or more algorithms he or she can abuse
- Policies and/or exploits which allow for some data to be leaked
- Permission to initiate tasks in the targeted collaboration

For example, the attacker might create a task to compute descriptive statistics on population N and subsequently create a second task that works on population N-1, thereby exposing the contribution of the singled-out patient.

5. Example: Aggregator Attack

There are two main scenario's in this case, the attacker:

(1) is part of the collaboration

(2) is *not* part of collaboration but gained access to the collaboration, see section 4.2.

In case (1), the attacker needs to obtain the aggregator position. Whether the aggregator is set to a trusted party, a random party or may be chosen by the initiator, is defined by the collaboration policies.

In case (2) the attacker might be able to assign the aggregator role to himself. He then would also need:

- A machine to run a vantage6-node, configured with the API. He or she is able to generate a new API key through the API using his illegal obtained credentials
- Depending on the algorithm he or she is going to use a valid dataset. This could be extra complex in case complex databases are used like OMOP.

For both case (1) and (2), they need:

- He needs to start the analysis
- That the collaboration approved an algorithm which is sensitive to aggregator attacks
- an external (malicious) application to reconstruct the patient record level data. Depending on the algorithm/reconstruction method this could be extremely complex

G. Write a project-specific risk analysis

The previous sections describe the general risks for a research project using vantage6. This section specifies which prevention and mitigation techniques you can employ to reduce the risks, depending on your research question.

- Consider which algorithms are needed for the collaboration. Each algorithm has its own risks (Section 3). Look into the risks associated with these algorithms. Limit the set of algorithms to minimize the potential attack vectors.
- Only a minimal set of data items required for the collaboration should be used. Apply generalization where possible. For example, use the age in years rather than the date of birth. Suppress values that occur infrequently or are unique to individuals.
- Consider disabling the optional peer-to-peer feature. It is only needed if the algorithms require direct communication with other algorithm containers. Typically it is required for vertically-partitioned algorithms (Section 1.4) or when an external FL library is used. This

may typically be checked in the algorithm documentation, algorithm code or with the algorithm developer.

- If one or more of the algorithms you intend to use is vulnerable to attacks from the aggregator position, consider a neutral/trusted party for the central part of the algorithm.
- Consider protecting the data by adding noise or anonymity filters (Section 3.1.1) at the cost of slightly inaccurate output or granular information loss.
- Consider if certain policies are needed to protect the record-level data. For example, ensure algorithm results never end up with less than X patients to do the analysis on.

Next, make sure that all best practices are used to deploy the server (all its components):

- Ensure the machines are kept up-to-date. For example, use cloud services that update machines automatically
- Limit the SSH port to be only reached by the people who need it.
- Close all unneeded ports
- Consider whitelisting IP addresses of all users and data stations. This is only possible if they are stable over a long period of time.

Ensure that the server is configured in the safest way:

- Enable 2FA. This makes it much more difficult for attackers to gain access to user accounts.
- Use encrypted collaborations.
- Give user accounts only the permissions they need, and not more.

Make sure node administrators in your project follow best practices in the node configuration:

- Whitelist only the algorithms that are used in the collaboration. Use the hash of the algorithm images you trust for extra security: an attacker that obtains write access to your Docker registry may overwrite your image with a malicious image but can't overwrite the hash.
- Close all unneeded ports, in principle only outgoing port 443 is required.
- Node machines should only be approachable by node administrators. It is best if the node machine can only be connected to from machines known to belong to node administrators.

Ensure the algorithms developed for your project consider security in their implementation:

- Algorithms using peer-to-peer communication should encrypt the VPN traffic to prevent it from being read if the EduVPN server is compromised.

Then, there are relevant factors outside the technical domain:

- Identify your collaborators and their trust level. Long-standing relations with respected institutions have higher trust levels than collaborations where anyone can join.
- Make your users aware that they should pick strong passwords and store those passwords in an appropriate password manager (instead of plain text).
- Stimulate users to protect their private key with a password

H. Future work

Vantage6 is still in development and therefore we still can reduce some risks by implementing new features. Some risk reduction features on the roadmap:

- As a mitigation measure we are currently investigating the use of a fresh token pattern [10]. This would mean that some operations on the API require a fresh token. For example changing a password or updating user permissions. A fresh token is only obtained after initial authentication, using a refresh token to obtain a new access token leads to a non-fresh token. Thereby limiting the possibilities of an attacker when an active session is seized.
- To limit the amount of code that needs to be reviewed, we are working on an algorithm build service. This service is responsible for packing and uploading the algorithm to the registry. This way it is very difficult for the hacker to hide the malicious code, as the only place where he can put code is in the algorithm package itself.
- The VPN network will be split per collaboration (or per task) to ensure that if an attacker gains access to the VPN network by limiting the number of exposed nodes.
- Traffic in the VPN network will be automatically end-to-end encrypted, no longer relying on algorithms to do this.
- Encryption is handled at user and node level instead of organization level. This mitigates when a private key is leaked as the items this key can decrypt is limited for this specific user or node.

I. References

[1] A. Moncada-Torres, “VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange,” in *AMIA Annual Symposium Proceedings*, 2020.

[2] D. Smits, B. van Beusekom, F. Martin, L. Veen, G. Geleijnse and A. Moncada Torres, “An Improved Infrastructure for Privacy-Preserving Analysis of Patient Data,” *Advances in Informatics, Management and Technology in Healthcare,* pp. 144-147, 2022.

[3] TNO, “Multi Party Computation,” TNO, 2022. [Online]. Available: https://www.youtube.com/watch?v=WRU_nUeqVu8.

[4] A. S. Foundation, “CVE,” CVE, [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2021-44228. [Accessed 2023].

[5] R. Gosselin, “Privacy and Security in Federated Learning: A Survey,” *Applied Sciences,* 2022.

[6] V. Mothukuri, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems,* 2021.

[7] N. Truong, “Privacy Preservation in Federated Learning: An insightful survey from the GDPR Perspective,” *Computers & Security,* 2021.

[8] J. Zhang, “Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges,” *Security and Communication Networks,* 2022.

[9] E. e. a. Bagdasaryan, “Differential Privacy Has Disparate Impact on Model Accuracy”.

[10] “Flask JWT Extended documentation,” [Online]. Available: https://darksun-flask-jwt-extended.readthedocs.io/en/stable/token_freshness/. [Accessed 6 3 2023].