This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare

Deliverable 2.5
"Secure Architecture and Safety Evaluation"

Work Package 2
"Cyber Risk Assessment and Mitigation"

Disclaimer

Copyright message

Document information

Grant Agreement Number: 826078			**Acronym: FeatureCloud**	
Full title	Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare			
Topic	Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures			
Funding scheme	RIA - Research and Innovation action			
Start Date	1 January 2019	**Duration**	60 months	
Project URL	https://featurecloud.eu/			
EU Project Officer	Christos Maramis, Health and Digital Executive Agency (HaDEA)			
Project Coordinator	Jan Baumbach, University of Hamburg (UHAM)			
Deliverable	D2.5 – Secure Architecture and Safety Evaluation			
Work Package	WP2 - Cyber Risk Assessment and Mitigation			
Date of Delivery	**Contractual**	31/12/2023	**Actual**	19/01/2024
Nature	Report	**Dissemination Level**	Public	
Lead Beneficiary	SBA			
Responsible Author(s)	Rudolf Mayer (SBA)			
	Richard Röttger (SDU)			
Keywords	Cyber security, risk assessment, cyber risk mitigation, KPIs, platform architecture, privacy, secure design, cloud security, federated learning			

History of changes

Version	Date	Contributions	Contributors (name and institution)
V0.1	30/11/2023	First draft	Anastasia Pustozerova (SBA) Tanja Šarčević (SBA) Daryna Oliynyk (SBA) Daniela Martinez (SBA) Rudolf Mayer (SBA)
V0.2	08/12/2023	Comments / edits	Rudolf Mayer (SBA) Richard Röttger (SDU)
V0.3	15/12/2023	Final draft	Anastasia Pustozerova (SBA)
V1.0	19/01/2024	Final version	Rudolf Mayer (SBA) Nina Donner (concentris) Jan Baumbach (UHAM)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Table of Content

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

1 Table of acronyms and definitions

AI	Artificial Intelligence
API	Application Programming Interface
App Store	Application Store
CA	Certificate Authority
CID	Consent identifier
concentris	concentris research management gmbh
CVSS	Common Vulnerability Scoring System
D	Deliverable
DDoS	Denial-of-Service
DP	Differential Privacy
DP-SGD	Differentially private stochastic gradient descent
GDPR	General Data Protection Regulation
GND	Gnome Design SRL / Egnosis
HTTPS	Hypertext Transfer Protocol Secure
KPI	Key Performance Indicators
LSB	Least Significant Bit
ML	Machine Learning
MS	Milestone
MUG	Medizinische Universität Graz
OWASP	Open Web Application Security Project
PAML	Privacy-Aware Machine Learning
Patients	In this deliverable, we use the term “patients” for all research subjects. In FeatureCloud, we will focus on patients, as this is already the most vulnerable case scenario and this is where most primary data is available to us. Admittedly, some research subjects participate in clinical trials but not as patients but as healthy individuals, usually on a voluntary basis and are therefore not dependent on the physicians who care for them. Thus to increase readability, we simply refer to them as “patients”.
PIDs	Patient Identifiers
RI	Research Institute AG & Co. KG
SBA	SBA Research Gemeinnützige GmbH
SDU	Syddansk Universitet
SMPC	Secure Multi-Party Computation
TLS	Transport Layer Security
UHAM	University of Hamburg
UI	User Interface
UMR	Philipps Universität Marburg
WFs	Workflows
WP	Work package

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

2 Objectives of the deliverable based on the Description of Action (DoA)

The main objective of WP2 is the definition of a security and privacy architecture **based on the cyber risk assessment and legal requirements**.

"While the approach of the FeatureCloud project by design mitigates most major concerns regarding security and privacy, the underlying foundations of the platform to be developed need to be secured thoroughly, especially considering the diversity of the local execution platforms on the hospital sites. Important attacker goals include the theft of data on a local level, as well as the theft and manipulation of results, with the inclusion of possible insider attacks."

Objective 5 of this work package aims ***"to develop a security architecture based on the requirements derived from the risk analysis and the developed mitigation strategies"***.

The corresponding task in this work package is ***Task 5 Secure Architecture:***
"Based on the developments within this work package and the requirements derived from WP3, a secure architecture of the FeatureCloud Platform will be developed. This also includes the incorporation of the PAML-Layer and the mitigation mechanisms, as well as formulating the developed KPIs in a way to make them testable for real-world local executions platforms."

3 Executive Summary

This deliverable focuses on comprehensively addressing the security aspects of the FeatureCloud Platform architecture. It systematically evaluates the progress made since previous work and conducts an in-depth analysis of the current security and privacy status across various FeatureCloud components. The key aspects covered in this deliverable include:

- In the section Secure Architecture, we discuss the secure architecture of the FeatureCloud platform, emphasizing the rigorous risk assessment undertaken during its development. The architecture incorporates security standards, including communication encryption and secure protocols for aggregation (Secure Multi-Party Computation, SMPC). We also provide recommendations for securing local and networked systems, Docker containers, and storage of sensitive data, addressing aspects like authentication, encryption, and adherence to regulations like GDPR.
- Further, we discuss the Key Performance Indicators KPIs set in Deliverable D2.2 "KPIs and metrics for local execution platforms" and which actions were taken in place to satisfy the KPIs.
- The Blockchain section is dedicated to the analysis of privacy and security of the consent management system provided by FeatureCloud.
- We provide a systematisation of threats and defences integrated into the Privacy-Aware Machine Learning Layer (PAML-Layer) on the FeatureCloud Platform. We provide detailed analysis of different privacy threats, including attacks on models and data confidentiality as well as unauthorised data and model access. We provide an overview of the mitigation techniques and defences integrated into the FeatureCloud platform. Most importantly, we provide guidelines for the usage of different mitigation techniques, including the combination of multiple mitigation techniques, the order of application of different mitigation techniques, and recommendations for setting the parameters.

4 Introduction (Challenge)

This deliverable primarily focuses on addressing the significant challenge of systematically organising and enhancing the security aspects across all components of the FeatureCloud platform. The key objective is to ensure that each component and aspect of the overall system and platform is secured, aligning with contemporary privacy and security standards. To achieve this, we answer the following questions:

- How does the FeatureCloud platform ensure the security of the architecture?
- How does the FeatureCloud platform ensure the security and privacy of the Blockchain and consent management component?
- How does the FeatureCloud platform meet the security and privacy KPIs set in the Deliverable D2.2 “KPIs and metrics for local execution platforms”?
- How does the FeatureCloud platform incorporate PAML-Layer? Which defence and mitigation mechanisms are integrated into FeatureCloud to ensure the privacy of the sensitive data?

5 Results

5.1 Secure architecture

In the course of an ongoing discourse, a thorough risk assessment was conducted (see Deliverable D2.1), leading to the development of a secure FeatureCloud architecture. The primary objective was to align with stringent security standards, encompassing crucial elements such as communication encryption, and a secure aggregation protocol realised via Secure Multi-Party Computation (SMPC). An important aspect of FeatureCloud architecture is that Docker containers are kept separate and cannot communicate with the outside. It is also a crucial aspect that the FeatureCloud platform needs to be deployed in each participating hospital, and a subsequent limitation is that the platform operators do not have direct control over the system outside of the Docker environment. For these host systems, we can only provide guidelines and offer support in securing them, but the eventual responsibility and control lies with the administrators of these host systems. One critical recommendation thus is, if feasible from a data management point of view, to isolate the local host system for the FeatureCloud platform from other systems within the host organisation (e.g. the hospital).

On the local system level, threats can be caused by weak organisational and technical policies, e.g. by the protection of physical access to computing resources, to the authentication methods and policies (e.g. password strength and rotation policies, multi-factor authentication, …), security patching practices, or enforcing of encryption of storage devices. Recognising the critical importance of adhering to robust security measures, administrators of these local systems are strongly recommended to adhere to standard secure practices as outlined in task 1 of this work package, and described in deliverable D2.1.

For the feature cloud platform itself, we further followed design guidelines aiming at considering security aspects from the beginning. During the design phase, the Open Web Application Security Project (OWASP) Application Security Verification Standard (ASVS)[1] provides guidelines for technical security controls and also provides a list of requirements for secure development. Our internal development processes diligently followed these guidelines, to ensure that the FeatureCloud platform is designed with security in mind.

[1] https://owasp.org/www-project-application-security-verification-standard/

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

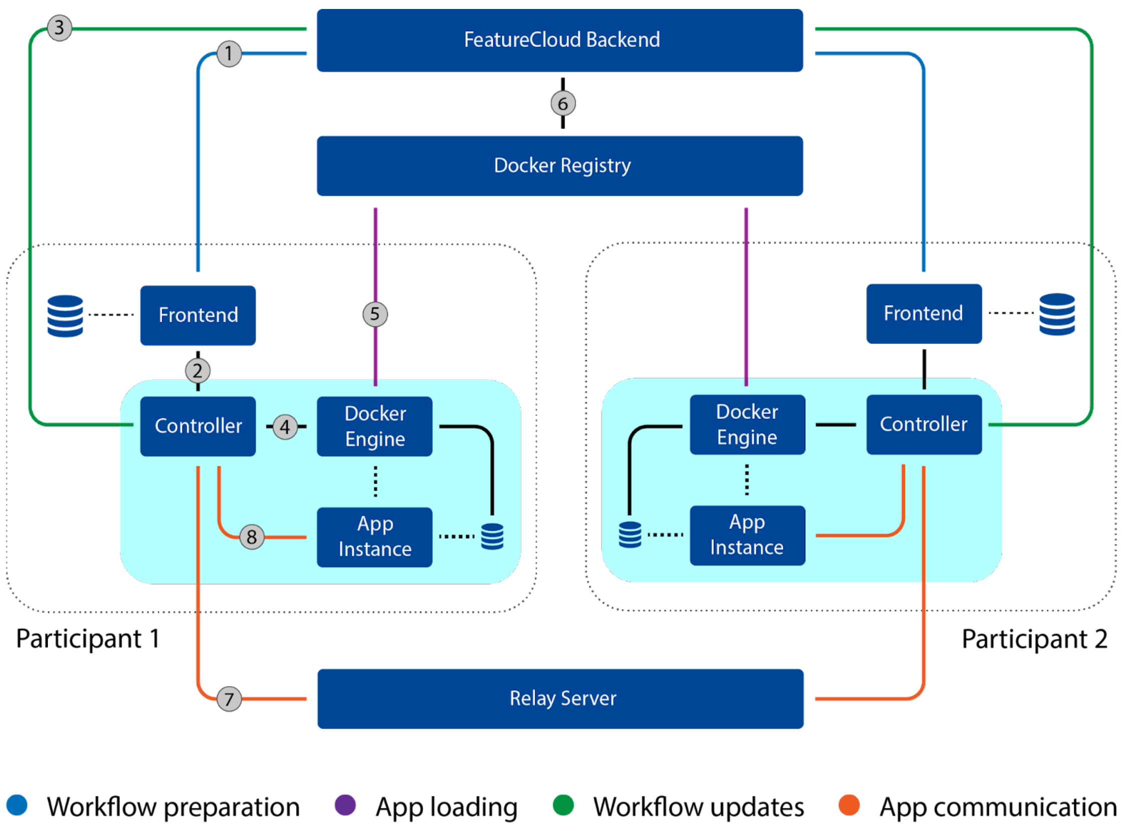

Figure 1: FeatureCloud Platform architecture

The final architecture of the FeatureCloud Platform is presented in Figure 1. It comprises various components, including a FeatureCloud global backend, a Frontend web application, a local Controller (deployed at every data holder's, i.e. participant's, site) for workflow execution, and a Relay Server for federated algorithm communication. The Frontend uses the FeatureCloud Backend API for AI Store features and project and workflow management, while the Controller manages local workflow execution using Docker containers. The Relay Server ensures secure communication for federated algorithms (see more in Deliverable D7.2 "App store ready and extendible by developers").

For security, Docker isolates app implementations contributed by external developers, preventing access to other local system resources besides the provided input data. Apps communicate with the Controller through an HTTP-based API, acting as web servers and waiting for queries instead of actively sending data. The configuration allows the use of a private Relay Server instead of direct peer-to-peer communication. All communication uses encryption.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

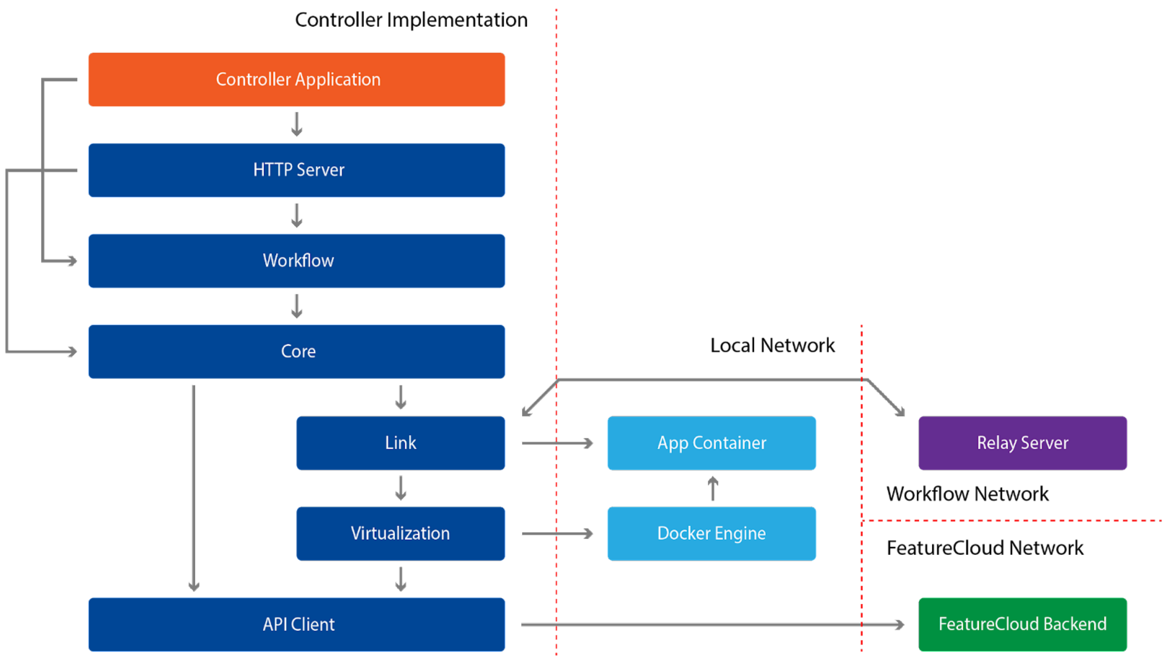

Figure 2: FeatureCloud Controller

The secure architecture of the FeatureCloud Platform in detail includes the following components:

- **Local systems**, hosting the "FeatureCloud Controller", e.g. in hospitals and their host's systems. As local systems still largely stay under the control of the local administrators, FeatureCloud
    - Provides recommendations and guidelines for (local) system-level architectures and security principles, e.g. strong authentication and authorisation to safeguard local system access and other rigid user management aspects, full-disk encryption, recommendations regarding update and upgrade policies, to the host systems. The recommendations for secure local system-level architecture include e.g. the NIST cybersecurity framework[2] or those from the Open Web Application Security Project (OWASP), including the OWASP Top Ten security threats[3], and the OWASP proactive controls[4] (see Deliverable D2.1 "Risk assessment methodology"), LINDDUN[5], STRIDE[6] and ENISA (ENISA 2021[7]).
    - Provides guidelines and standards for the FeatureCloud-specific systems deployed there, e.g. mostly the docker containers containing the FeatureCloud Controller (see Deliverable 8.7 "Report on Data Protection Impact Assessment").
    - Further recommends that local systems undergo regular security auditing and testing.
- Recommendations for **securing networked systems** include various measures, like encrypted communication. These measures are particularly crucial for ensuring the security of communication channels, specifically those connecting the FeatureCloud backend,

[2] https://www.nist.gov/cyberframework/framework
[3] https://owasp.org/www-project-top-ten/#
[4] https://owasp.org/www-project-proactive-controls/
[5] https://linddun.org/
[6] https://owasp.org/www-community/Threat_Modeling_Process
[7] https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

FeatureCloud frontend, and individual nodes hosting FeatureCloud controllers, as shown in Figure 1. In the context of the FeatureCloud architecture, a notable consideration is the deliberate avoidance of peer-to-peer communication among individual participants. This decision is partly motivated by challenges such as the local network restrictions that make direct peer-to-peer communication infeasible. Instead, messages within the FeatureCloud architecture are exchanged through a central relay server. As all network communication, also these messages are encrypted during transport.

- Recommendations for **storage of sensitive data** include various measures that are both imposed by industry standards and regulations (GDPR). One of the measures is defining the minimum retention policies to govern the lifecycle of sensitive data. These policies ensure that the data and metadata are not stored within FeatureCloud longer than necessary. The policies also imply the regulatory compliance with data disposal after the minimum retention period, which will be achieved by appropriate encryption techniques or secure deletion. These measures apply on all FeatureCloud-specific aspects of the system. This will be in the form of enforceable practices, e.g. within the FeatureCloud controller, or guidelines towards the app developers when it comes to recommendations on retention policies.

5.2 KPIs

The Key Performance Indicators (KPIs) and metrics designed in Deliverable 2.2 were crafted to be testable in real-world local execution platforms. The KPIs serve as systematic minimal requirements for assessing the security of the FeatureCloud platform and the individual applications in the FeatureCloud App Store. The metrics are instrumental in identifying potentially risky installations. Recommendations on where and how to monitor these KPIs were defined in Deliverable D2.2. Here we outline the overall characteristics of the KPIs and validate that the FeatureCloud platform meets the required criteria to guarantee the security and integrity of the platform.

KPI 1: No global service inside hospitals

Key Performance Indicator 1 emphasises the avoidance of global services within hospitals for security reasons. Running global services, accessible from the internet, on local hospital machines poses a security risk, making them vulnerable to masquerading and denial-of-service attacks. Hospitals typically prohibit global services due to IT policies, reinforcing the importance of this KPI. Instead, a socket server relays model parameter values between participating sites securely. Potential threats include information disclosure through reconnaissance, (Distributed) Denial-of-Service (DDoS) attacks, and exploits leading to data breaches.

Mitigation involves a security-by-design approach, restricting hospital services to local accessibility only. FeatureCloud ensures encrypted communication on specific ports when managing the workflow. The App Instance running inside a hospital is started on a private docker network by the controller, thus has no access to the internet (see Figure 1) and prevents containers from opening any connection. We ensure by design that no system components on a hospital machine provides a global service, therefore meeting the requirement of the KPI 1.

KPI 2: Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) serves as a freely available industry standard to evaluate the severity of security vulnerabilities in computer systems. It aims to allocate severity scores to vulnerabilities, enabling responders to prioritise actions and allocate resources based on the perceived threat level. In Deliverable D2.2 “KPIs and metrics for local execution platforms”, we set the threshold for achieving this KPI of 3.9. The CVSS score can be calculated by local system administrators in the data holders’ institution using the tool at https://www.first.org/cvss/.

To access the CVSS score of the FeatureCloud platform components such as the client and controller, we consider example scenarios and calculate the score for each of them. It should be noted that a re-calculation of the CVSS score has to be done after any change in the FeatureCloud system. The current CVSS score for FeatureCloud platform is calculated for several attack scenarios including local and remote settings.

Local Scenario
In Deliverable D2.2 “KPIs and metrics for local execution platforms” we proposed a scenario where a medical doctor has access to the used frontend of FeatureCloud. The attacker can thus steal the information via a USB stick that a doctor stocks into a local machine. In addition to this specific scenario, we extend the consideration to any type of breach into a local machine and re-evaluate the CVSS score. For the current state of the FeatureCloud platform results in the following vulnerability values:

- Attack Vector (AV): Local (L)
- Attack Complexity(AC): High (H)
- Privileges Required (PR): High (H)
- User Interaction (UI): Required (R)
- Scope (S): Unchanged (U)
- Confidentiality (C): Low (L)
- Integrity (I): None (N)
- Availability (A): None (N)

The corresponding vector string is CVSS:3.1/AV:L/AC:H/PR:H/UI:R/S:U/C:L/I:N/A:N. This scenario leads to a CVSS score of 1.8 (low), which is lower compared to D2.2 (CVSS score for Local scenario was 4.0) and our KPI is achieved by having a CVSS below 3.9. We were able to improve the score from 4.0 down to 1.8 on the current KPI due to the implementation of anonymisation techniques which allow us to mitigate the risks from attacks on confidentiality, i.e. protect the private data on the local machines.

Remote scenario: Masquerading

- Attack Vector (AV): Network (N)
- Attack Complexity(AC): High (H)
- Privileges Required (PR): None (N)
- User Interaction (UI): Required (R)
- Scope (S): Unchanged (U)
- Confidentiality (C): Low (L)
- Integrity (I): Low (L)
- Availability (A): None (N)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

This scenario amounts to a CVSS score of 2.1 (low) with vector string: CVSS:4.0/AV:N/AC:H/AT:N/PR:N/UI:A/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N. We were able to achieve the score satisfying the KPI (below 3.9) due to implemented in FeatureCloud mitigations like: Strong authentication and authorisation mechanisms, Encryption of data storage and communication, Minimisation of data exchange and usage of privacy enhancing techniques like DP, SMPC (see Section 5.4). Moreover, by design FeatureCloudensures that neither the coordinator nor a participant can access personal data of other participants.

Remote scenario: Eavesdropping

- Attack Vector (AV): Adjacent (A)
- Attack Complexity(AC): High (H)
- Privileges Required (PR): None (N)
- User Interaction (UI): Required (R)
- Scope (S): Unchanged (U)
- Confidentiality (C): None (N) (because of HE/SMPC)
- Integrity (I): Low (L)
- Availability (A): None (N)

This scenario amounts to a CVSS score of 2.6 (low) with vector string: CVSS:3.1/AV:A/AC:H/PR:N/UI:R/S:U/C:N/I:L/A:N

KPI 3: Privacy Requirements for Federated Algorithms Working on Patient Data

Privacy Requirements for Federated Algorithms Working on Patient Data (KPI 3) focus on the secure sharing of information between hospital platforms, particularly model parameters. In federated learning, only model parameters have to be shared between different parties, however an adversary or malicious app may try to retrieve more information about the data, than just model parameters. To address this problem, the KPI introduces a measurement inspired by big-O notation from computational complexity theory, evaluating how the amount of exchanged data scales with the amount of patient data.

The KPI examines whether the exchanged data's growth is proportional to the patient data size. An acceptable federated application should have a shared-to-raw-data dependency ideally represented by O(1), indicating a constant relationship. This criterion can be automatically tested by creating artificial patient data chunks of varying sizes and measuring the corresponding network traffic. To satisfy the current KPI we introduced the additional step during the certification process (see Deliverable D7.3 ”Federated machine learning apps running in app store”). Each application published in the FeatureCloud app store and going through certification will be checked if the exchanged data's growth is proportional to the patient data size.

KPI 4: Encryption Requirements for Patient Data and Network Traffic

FeatureCloud meets the Encryption Requirements for Patient Data and Network Traffic (KPI 4) by ensuring the secure exchange of model parameters over the internet. All data traffic, including connections between the Controller and Socket Server, Controller and Consent API, Global API and Frontend, Controller and Frontend, and App Container and App Frontend, adheres to the HTTPS protocol with Transport Layer Security (TLS). This encryption method guarantees user authentication, data integrity, and confidentiality, effectively preventing unauthorised access and

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

manipulation. For patient data encryption, FeatureCloud employs a highly secure approach offering symmetric and asymmetric (public/private key) encryption. Patient data stored in medical centres is securely encrypted using Advanced Encryption Standard with a key size of 256 bits for symmetric encryption, ensuring a high level of protection and meeting the KPI goal set in D2.2. For the asymmetric encryption component, we employ elliptic hash curves (ECC), specifically using the P-224 curve (see Deliverable D7.3 “Federated machine learning apps running in app store”). In our earlier Deliverable D2.2, the minimum size goal for a public key was established at 2048 bits, assuming RSA encryption, which is equivalent to he P-224 curve with ECC encryption[8]. ECC proves more efficient than RSA by offering robust security with shorter key lengths, resulting in quicker computations and reduced storage and bandwidth requirements.

5.3 Blockchain

FeatureCloud addresses the challenge of analysing distributed data across multiple locations, advocating for federated machine learning to maintain data privacy and security. However, strong trust assumptions in collaborating parties raise concerns, prompting the need for a post-auditing process to verify the integrity of results, especially when data manipulation by hospitals is a potential threat. Key challenges include determining if hospitals executed machine learning models faithfully, detecting the generation of fake data, identifying any omission of patient data due to bias, or usage of the patients' data without consent (see more in Deliverable D6.2 “Model for defining user rights in federated machine learning”). GDPR and national data protection laws may require hospitals to obtain patient consent and track processing activities.

To address transparency, audibility and immutability concerns, FeatureCloud utilises blockchain technology. The FeatureCloud platform provides a generic User Interface (UI) for managing patient consent that can be adopted and integrated into hospital platforms and the potential for federated collaboration (see Deliverable D7.5 “Project manager and patients can control access rights with integrated blockchain”). Each participant of the federated learning (e.g. medical institution) deploys its patient UI to collect and manage consents locally, ensuring patient privacy. The Patient UI enables patients to provide, modify, or withdraw consent for utilising their data in research. Patients' consent within FeatureCloud is managed through paper or digital consent processed and hashed before submission to the blockchain system. The blockchain system tracks patients' data usage across different workflows, enabling external auditors to verify consented data in past federated collaborations. The controller, responsible for hashing consent, requires a specific file format (e.g., JSON or PDF). The patients' UI, deployed on hospitals' platforms, facilitates consent creation, updating, and revocation, with commitments submitted to the ledger.

Hospitals register local patient identifiers (PIDs), linked to patient-signed paper consents, and submitted to the ledger. Multiple PIDs can be created for a patient to prevent linking different consents. Hospitals inform patients about future research, creating a consent identifier (CID) linked to the PID. Patients can submit, update, or revoke consents, with commitments automatically submitted to the blockchain. For using data in an ML study, the corresponding consent is hashed and submitted to the FeatureCloud Ledger as part of the workflow. The hospital's IT personnel filter consented data, providing it to the project coordinator, who uploads it for WF execution. The controller hashes data and uploads it to the ledger. At the end of a workflow, the controller stores a

[8] https://web.archive.org/web/20150627183730/https://www.nsa.gov/business/programs/elliptic_curve.shtml

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

copy of data, consent(s), the workflow configuration, and results locally for auditing, proving that the study used only consented data.

5.3.1 Blockchain Security and Privacy

In the FeatureCloud context, patient data and consent, which are considered personal data, are used for research purposes, with hospitals acting as data controllers and processors This raw health-related data remains stored locally off-chain and does not leave the participant site. FeatureCloud ensures compliance with GDPR, collecting consent from patients in written or digital form for utilising their data in various studies. Consent contains identifiable information such as the patient's name, address, and the scope of the consent. These consents, along with the raw data, are stored off-chain, to ensure confidentiality of private data. Blockchain transactions contain only blinded commitments of consents and ML studies that do not reveal personally identifiable data, only hashed data and pseudonymous identifiers, ensuring a high level of protection against deanonymization. Re-identification risks are deemed reasonably unlikely due to large input spaces, diverse formats, and secure local storage.

The patient ID (PID) serves as a local identifier specific to the participant, ensuring that each hospital has a unique identifier for the same patient, and these identifiers remain unlinkable. The use of public keys or certificates, which may be involved in committing changes to patients' consents, is carefully managed to prevent linkability. Employing single-use identifiers further enhances the network's inability to link separate consents to the same patient. The commitment to privacy extends to the off-chain data, which remains securely stored locally within each participant. The implementation of a random blinding factor stored securely off-chain adds an additional layer of protection against deanonymization, contributing to the robust privacy framework of FeatureCloud.

A digital certificate contains identifying information about an entity, keying material, issuer, subject, and additional attributes. Its utility lies in demonstrating ownership of an identity, contingent on trust in the issuing entity (a certificate authority). Moreover, digital certificates play a crucial role in ensuring secure communication with other entities. The cryptographic elements outlined in these certificates are employed for digitally signing and verifying the authenticity of transactions. The FeatureCloud platform incorporates a built-in Certificate Authority (CA) component that manages X.509 certificates. A certificate authority (CA) is a trusted entity issuing digital certificates. In FeatureCloud, each participant can have its own certificate authority, thus issuing its own certificates (identities) to its patients, doctors or admins (see more in Deliverable D6.4 “Prototypical implementation of phase 2 and evaluation results”).

To ensure secure communication, FeatureCloud employs TLS encryption. All the smart contract functions are provided as REST services directly accessible using valid x509 certificates and their corresponding private keys, or through a web api for non-tech savvy users. The web application interface enables patients managing their consents, by interacting with the smart contract functions. The web api is protected by a login/password mechanism, and upon successful authentication a Json Web Token (jwt) is provided for subsequent interactions. Each participant independently hosts its dedicated web API and REST services, protected by a firewall. The smart contracts incorporate access controls to restrict access to on-chain data or smart contract functions to specific users.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

5.4 Privacy threats in federated learning. Defences, mitigations and PAML-Layer integrated into FeatureCloud.

Table 1: Privacy threats and integrated into FeatureCloud mitigation techniques. A green check mark indicates that a mechanism mitigates the corresponding attack. A yellow asterisk indicates that a mechanism mitigates corresponding threats to some extent or in specific conditions.

		Data Anonymisation	Synthetic Data	Data Fingerprinting	SMPC	Local DP, Hybrid DP	Central DP	LSB Sanitisation	Model Watermarking
Data privacy attacks	Identity disclosure, Membership disclosure	✓	✓			*			
	Attribute disclosure	*	*						
	Unauthorised data re-distribution	*	*	✓					
Privacy attacks on models	Inference from local models	*	*		✓	✓			
	Inference from global models	*	*			✓	✓		
	Data exfiltration	*	*					✓	
	Model Stealing								✓
	Cost	Utility loss	Utility loss	Utility loss	• Computational overhead • Communicational overhead	• Possible computational overhead • Utility loss	• Possible computational overhead • Utility loss	Utility loss	• Utility loss • Computational overhead

Featurecloud enhances privacy by design utilising federated learning as a core of the platform. There are, however, other privacy risks coming from the storage of sensitive or pseudonymised data and residual privacy risks in federated learning. To address these threats, we incorporate a privacy-aware machine learning (*PAML*) layer in FeatureCloud, and provide several further mitigation mechanisms that can address residual privacy risks in federated learning.

From Table 1, one can observe that for each threat, the FeatureCloud platform provides one or two mitigation options. Data anonymisation techniques and synthetic data, as provided by the PAML layer, protect data from identity and membership disclosure; this is also achieved as well by Local DP via input perturbation. To some extent, data anonymisation and synthetic data can mitigate risks of attribute disclosure, as the ability of an attacker to perform attribute disclosure from sanitised data is reduced. Also by design, they mitigate to some extent other threats like unauthorised data redistribution, inference from local and global models and data exfiltration, as not the original data is utilised. Data Fingerprinting provides a reactive defence against unauthorised data redistribution, thus allowing tracing of unauthorised data usage. To prevent data leakage through the machine learning models, we employ Secure Multiparty Computation (SMPC), which addresses confidentiality of the local models, and Differential Privacy (including Local DP, Hybrid DP, and Central DP: see more in the section "Differential Privacy"), which can protect the local and global models. The Least Significant Bit (LSB) sanitization and certification process are used to mitigate risks against data exfiltration attacks. To trace unauthorised model re-distribution and model stealing, the FeatureCloud platform provides model watermarking functionality.

Each defence and mitigation strategy impacts the privacy of the data and models, aiming to protect their confidentiality. At the same time, each of the protective techniques results in additional costs, which users have to acknowledge. Most of the approaches have a direct impact on the utility of data or/and machine learning models. For every considered defence, besides SMPC, users have to acknowledge this privacy-utility trade-off. When using SMPC, additional computational and communication overhead should be acknowledged. With some Differential Privacy approaches

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

(e.g. DP-SGD), one can also experience additional computational overhead. The application of model watermarking also requires additional computational resources. For other defences, the computational overhead is not significant.

Table 1 provides an overview of the privacy threats considered, and tools integrated into FeatureCloud to mitigate these. Later in this section, we discuss the threats in detail and provide details on the defences and mitigation that can be used against these threats. We describe the mitigations integrated into FeatureCloud and provide guidance on their usage to ensure that users of the FeatureCloud platform apply mitigations in a meaningful way. Further, in Figure 3, we demonstrate at what stage of data analysis different attacks can be possibly executed. Besides the attacks, we also show at which stage and in which sequence defences and mitigations against privacy attacks should be applied.

5.4.1 Threats to privacy in shared data

Identity disclosure, also referred to as re-identification, is generally considered the strongest form of disclosure. It means that an attacker can associate an individual with a specific record. It is often achieved via a record-linkage attack, where the partial information is linked to other data available to the attacker, e.g. on the so-called quasi identifiers such as birthdate, ZIP code, etc. The data that is linked to also includes the identification.

Attribute disclosure is a type of disclosure where an attacker learns (exactly or approximately) the value of one or more attributes of an individual that are contained in the targeted dataset. For example, an attacker might learn the medical diagnosis, or the salary, of an individual in a dataset, given some background knowledge on the individual. Attribute disclosure might be achieved even without uniquely associating an individual to a specific record in a dataset.

In **membership disclosure,** the attacker infers whether an individual is contained in the dataset. Unlike other disclosure settings, it does not directly release sensitive attributes from the dataset, i.e. it does not directly unveil information contained in the dataset, making this generally the weakest form of considered disclosures. However, an attacker might infer meta-information from the membership of an individual in the dataset, e.g. if this dataset is a medical dataset containing information about patients carrying a certain disease.

Mitigations against privacy attacks on data: Data Anonymisation and Synthetic Data

K-anonymity (Samarati and Sweeney, 1998) is an anonymisation technique that is used to ensure data privacy of personal, sensitive data. This technique protects against re-identification by obfuscating information that can be used to link individual records. In particular, it ensures that quasi-identifiers of at least *k* records in a dataset are identical. The records sharing the same quasi-identifiers are called an *equivalence class*. Common approaches to achieving *k*-anonymity are generalisation, suppression and microaggregation. Generalisation decreases the level of granularity of an attribute. Suppression removes a value entirely from the dataset. Microaggregation perturbs the data by partitioning it into groups of *k* records and replacing the values in the partition with an aggregated output (e.g., using the mean or median). More details on *k*-anonymity were discussed in Deliverable D2.3 “Working PAML-Layer with low distortion”.
Data anonymisation techniques generally transform *personal data* into *fully anonymised data*, i.e. contain information which does not relate to an identified or identifiable natural person. This allows further data processing as the anonymised data is not subject to the same restriction placed on the

processing of personal data under the General Data Protection Regulation (GDPR), where, for example, appropriate consent needs to be given[9].
As shown in Table 1, *k*-anonymity mitigates identity disclosure on a high level, as no single record could be linked to a single individual, as well as membership disclosure, since the data is generalised[10]. Furthermore, the risk of attribute disclosure is generally reduced; for a general protection against this risk, however, extended privacy models such as *l*-diversity and *t*-closeness need to be applied. k-anonymity also significantly reduces the risk of inferences from the downstream local or global models, since any model represents an additional generalisation of the underlying data.

Cost of anonymisation: *k*-anonymity implies modifications on data that inevitably reduce data utility. If the extended privacy models are applied, the utility loss increases. Hence, there is a trade-off between privacy and utility that needs to be met when applying anonymisation techniques. Although there are no industry standards for choosing an appropriate *k* and it completely depends on the context, as well as the legal assessment, one can assess the information loss via various metrics, such as granularity, record-level squared error, etc., for an acceptable range of *k*-s to achieve a good trade-off (Šarčević et al., 2020) .

Integration of anonymisation techniques in FeatureCloud

We provide two pre-processing apps for data anonymization: k-anonymity achieved via generalisation, and via micro-aggregation. These apps allow users to generate locally anonymized versions of their datasets and use this data instead of the original one for training the machine learning models in the federated pipeline. This serves as a protection mechanism for local data[11]. From Figure 3, one can see that the anonymisation apps should be applied at the pre-processing stage.

The **k-anonymity app** provides the capability for anonymizing sensitive data using k-anonymity and other approaches (e.g. l-diversity, t-closeness). The user can use generalisation by defining hierarchies and specifying the value of k and the set of quasi-identifiers. Additionally, the user can use privacy models such as t-closeness or l-diversity for protecting sensitive attributes. The app returns an anonymized dataset based on the configuration parameters.

The **microaggregation app** provides the capability for anonymizing datasets using different protection methods including suppression and microaggregation algorithms. Depending on the attribute type, the user can indicate the level of protection, the attribute type, the data type and the value of k. Optionally, it can include ontologies for modelling the domains of nominal data from a semantic perspective (D et al., 2020). The app returns an anonymized dataset based on the configuration parameters.

Synthetic data is artificial data that mimics the structure and properties of real-world data. Various models, such as GANs (Generative Adversarial Network), VAEs (Variational Autoencoders) and Bayesian Networks can be employed for the generation process (Figueira and Vaz, 2022) . While synthetic data can be used for goals such as data augmentation or imputation, the main goal of synthetic data in FeatureCloud is to protect the privacy of individuals within a sensitive dataset. Since there is no one-to-one correspondence between an entry in the original dataset and an entry in a fully synthetic dataset, re-identification is not considered a meaningful threat. Synthetic data

[9] Regulation (EU) 2016/679 (General Data Protection Regulation), recital 26
[10] In some cases, however, membership still can be inferred, e.g. if it is known that all potential members of an equivalence class are also contained in a dataset.
[11] And, as outlined in Deliverable D2.3 “Working PAML-Layer with low distortion”, can also be used in some jurisdictions in case consent was not obtained

serves as an effective alternative for training machine learning models in place of real data, offering valuable insights for researchers while reducing privacy risks[12]. More information about synthetic data generation can be found in Deliverable D2.3 “Working PAML-Layer with low distortion”.

Synthetic data can mitigate risks from identity and membership disclosure, as depicted in Table 1. Specifically, with fully synthetic data, the likelihood of identity disclosure is very low, since there is no direct mapping between the records in the original dataset and the synthetic dataset (Giomi et al., 2022). Similarly, using synthetic data instead of real data for training machine learning models can significantly reduce the effectiveness of membership inference attacks compared to models trained on real data, even in the case when synthetic data models overfit the real data (Khan and Buchegger, 2023).
It has to be noted, however, that there is still a lot of active research in the achieved privacy and residual risks in synthetic data, and no final consensus on this is reached within the research community. Further, synthetic data, unlike other mechanisms such as Differential Privacy or k-anonymity, does not exhibit a direct quantification of achieved privacy (such as the privacy budget ε or the achieved k). FeatureCloud consortium members are actively contributing to advance the state of the art in this area, by research publications and also participation into an IEEE standardisation group on privacy-preserving synthetic data[13].

Integration of Synthetic Data in FeatureCloud

In FeatureCloud, we provide a pre-processing **synthetic data app** that generates fully synthetic data from the local datasets. The synthetic data can then be used for training local models instead of real data. The app offers various model options for generating synthetic data, including Gaussian Copula, CTGAN, TVAE and Copula GAN. Additionally, it enables users to generate a given number of synthetic records, create fake data for a certain column (e.g. address information) and filter the columns used before training by selecting those for synthesis. The synthetic data app should be used at the pre-processing stage, like anonymisation apps (see Figure 3).

Cost of synthetic data: Depending on the model chosen for generating synthetic data, the fidelity of synthetic data with respect to real data may exhibit variations. This can have an impact on the utility when using synthetic data instead of real data. In particular, synthesised datasets that exhibit larger differences with real data also exhibit lower performance when employed for training machine learning models (Hittmeir et al., 2019). Therefore, for selecting the best model for synthetic data generation one should assess fidelity and utility using multiple metrics. For fidelity, one can use statistical tests to ensure that univariate distributions in real data are preserved, compare correlation scores in synthetic and real data and use metrics of distinguishability to compare the similarity on the entire distribution (Dankar et al., 2022). For utility, one can compare the performance of machine learning models training on real data and testing on real data against the performance when training on synthetic data and testing on real data. Additionally, one also might consider the size of the original data, to select the synthetic data method. For instance, deep learning models might incur higher computational costs, while in some cases, still providing the same utility as simpler models.
While these utility losses are inevitable, they can at least be estimated on validation data; FeatureCloud recommends this step before utilising synthetic data.

[12] Again, another benefit of using synthetic data might be in being able to leverage data that otherwise has no user consent, as discussed in Deliverable D2.3 “Working PAML-Layer with low distortion”
[13] https://standards.ieee.org/industry-connections/synthetic-data/

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

5.4.2 Unauthorised data redistribution

Local data in the federated process is prone to unauthorised re-distribution if trusted or untrusted parties gain access to the data. This is especially amplified by the value of sensitive, private data, e.g. in the medical domain with sensitive patient data. Therefore, both the unauthorised re-distributions should be deterred and the ownership rights of the original data should be respected.
As a defence strategy against unauthorised data redistribution, we employ *data fingerprinting*. This is a *passive* defence method that allows the identification of the legitimate owner and unauthorised redistributor after the redistribution happens. As a consequence, it might also deter unauthorised distribution in the first place.

Mitigation against unauthorised data redistribution: Data Fingerprinting

Data fingerprinting is an information-hiding technique based on steganography, and enables tracing of unauthorised data usage. When applied to data, the fingerprint is embedded into the original data by slightly modifying its content. In case of suspicious re-distribution, the legitimate owner can perform a detection process that identifies them as a legitimate owner and identifies the malicious party initially received the data; while this might not always be the party who distributed the data without authorisation (e.g. if that party itself was subject to an attack), it provides a first trace. For more information about data fingerprinting see Deliverable D2.4 “Set of (novel) attack vectors and countermeasures”.

Integration of Data Fingerprinting in FeatureCloud

In FeatureCloud, the clients can use the **fingerprinting app** to apply fingerprinting to their data as part of pre-processing (see Figure 3). It is critical that fingerprinting is applied at the last stage of preprocessing, so the fingerprint is not damaged (and consequently rendered ineffective) by other pre-processing steps. The app is, firstly, used to embed the fingerprint in the tabular data. This is achieved by modifying the data values in a meaningful pattern using secret information known only to the client. The resulting fingerprint is robust against a range of data modifications at minute utility cost (Šarčević and Mayer, 2019). The applied fingerprint is by default designed to achieve the highest robustness since the utility cost is generally negligible. However, the user can optionally specify a weaker fingerprint in cases where the resulting utility does not meet their standard.
Secondly, the app is used to extract the previously embedded fingerprint from the data. The expected output is the identification of the rightful owner, and identification of the unauthorised re-distributor, in case of re-distribution.

Cost of Data Fingerprinting: Data modifications imply that the utility of the data is affected, however, due to the fact that the modifications are marginal, e.g. much less prominent than those of data anonymisation, these effects are minute.

5.4.3 Inference attacks from machine learning models

Local and global models shared in federated learning, either during and or after the training process, may leak sensitive information about training data, which might be uncovered through various inference attacks. Such confidentiality attacks on machine learning models include membership inference (Hu et al., 2022; Shokri et al., 2017), attribute inference (Zhao et al., 2021) (or the stronger form of that, model inversion (Hidano et al., 2017)), and property inference (Ganju et al., 2018). These attacks and risks stemming from them were thoroughly discussed in Deliverable D2.1 “Risk assessment methodology” and Deliverable D2.4 “Set of (novel) attack vectors and countermeasures”. In one of our works (Pustozerova and Mayer, 2020), we showed how these attacks can be employed in federated learning, therefore showing the need for additional mitigations.

In FeatureCloud, all communication traffic is encrypted (satisfying KPI 4), therefore, an outside attacker cannot gain access to local and global models by observing the communication processes in the federated learning. This is why we consider a threat model where the inference from local models can be performed only by a malicious or corrupted aggregator (see Figure 3). The inference from the global model can be performed by malicious or corrupted clients of federated learning or malicious users of a final distributed global model at the post-processing stage (see Figure 3).

To mitigate inference attacks from local and global models in federated learning, we provide techniques such as Secure Multi-party Computation and Differential Privacy (see Table 1).

Mitigations against inference attacks from machine learning models: SMPC and DP

Secure Multiparty Computation (SMPC) (Evans et al., 2018) is a cryptographic protocol designed to enable the computation of public functions over private data distributed across multiple parties. It ensures that no individual party can learn the sensitive data of others, making it suitable for collaborative learning scenarios where participants collectively compute functions without revealing their private inputs (see more in Deliverable D2.4 “Set of (novel) attack vectors and countermeasures”). SMPC safeguards input privacy, making it well-suited for federated learning settings.

In federated learning, SMPC can be used to compute the global model, e.g. as an average of local models. SMPC guarantees that no party except the local model owner has access to the local models. Therefore, SMPC mitigates privacy risks caused inference attacks from the local models, primarily eliminating the threat of an untrusted aggregator.

Integration of SMPC in FeatureCloud:

In FeatureCloud, clients can employ the SMPC protocols to jointly compute global models in federated learning (e.g. by local models averaging). SMPC ensures that inputs (local models) remain confidential during communication, so no party has access to local models; therefore the risk of inference from local models is reduced. The adapted SMPC implementation, available as an app template for the API, introduces an additional iteration before each aggregation step. While this may increase overall runtime by 2-fold in most cases, it significantly reduces the risk of interception (and decrypting) network communication, and further eliminates the need to place trust in the aggregator. In Figure 3, one can see that SMPC is used at the stage of the local models aggregation.

SMPC Cost. When using SMPC, one should acknowledge that it introduces efficiency challenges due to computational and communication overhead. SMPC also may encounter issues with client dropout, i.e. when some of the participating clients do not complete their local training and not send updates to the global model. FeatureCloud addresses this by utilising SMPC schemes where $n - t$ clients can drop out without hindering the final result, such as (t, n)-threshold secret sharing. In case the number of clients exceeds the $n - t$ dropouts, the aggregation step may be repeated, incurring potential computational overhead, but utilising previously computed local training results.

While SMPC protects the privacy of the local model, it discloses the final output, i.e. the global model. To address this potential vulnerability, FeatureCloud complements SMPC with techniques

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

like Central Differential Privacy to ensure output privacy (privacy of global models) in federated learning (see Table 1).

One of the benefits of SMPC is the lack of utility loss, unlike other techniques protecting the local models from inference attacks (e.g. Local Differential Privacy).

Differential Privacy (DP) achieves privacy by introducing a controlled amount of noise to the data or model (see more in Deliverable D2.4 "Set of (novel) attack vectors and countermeasures"). By introducing the noise, DP allows plausible deniability to mitigate privacy risks coming from data and model sharing, e.g. membership inference (Hu et al., 2022) (see Table 1). Important characteristics of DP are the *composition property* (a composition of several differentially-private algorithms is also differentially private), and the *privacy budget accounting* through parameter epsilon, which allows for control of the amount of noise added in DP, and therefore controls the privacy level. Differential Privacy can mitigate the impact of inference attacks on both local and global models shared in federated learning.

Dwork et al. (Dwork et al., 2006) proved that the privacy of the database can be preserved by adding noise according to the sensitivity of the function. Sensitivity denotes the maximum possible impact on the output of the function, caused by removing or adding any single instance to the database. In machine learning Differential Privacy can be applied at different stages, namely as input, gradient, objective or output perturbation. In Figure 3, we specify at which stage of the federated learning process different DP techniques should be applied.

DP via **input perturbation** (**Input DP**) (Fukuchi et al., 2017; Jarin and Eshete, 2022) can be achieved by adding noise to the data. By adding small, controlled noise to the data that becomes the input to a model training process, we make that model differential private. Application of Input DP happens on the local data of the clients in federated learning, at the pre-processing stage (see Figure 3: Input DP); it guarantees differential privacy of both the local and global models due to the sequential composition property of differential privacy (Dwork and Roth, 2014).

DP of a machine learning algorithm can be further achieved during model training via **objective perturbation** (Objective DP) or **DP-SGD.** Objective perturbation (Chaudhuri et al., 2011) suggests adding noise to the objective function during the training process. Differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016) achieves differential privacy by clipping the gradients, which are computed from the loss and used for updating model parameters in a training process of e.g. neural networks or logistic regression, and adding noise to them. DP-SGD remains one of the most popular ways for achieving privacy in machine learning, as it does not put any requirements on the loss function, nor requires knowing the sensitivity of complex machine learning algorithms. In federated learning, Objective DP and DP-SGD can be applied during local model training (see Figure 3).

After models are trained, one can achieve DP using **output perturbation** (Output DP)**.** To apply output perturbation, one has to know the sensitivity of a machine learning algorithm, which allows one to apply the Laplace Mechanism (Dwork, 2006) on the learned parameters of a model to achieve differential privacy. For example, in (Chaudhuri and Monteleoni, 2009), the authors show that the sensitivity of regularised logistic regression with the L2 regularisation parameter λ is at most $2/n\lambda$, where n is the number of samples in the database. In federated learning, Output DP can be applied on the locally trained models before aggregation (see Figure 3, "analysis stage") to achieve differential privacy for both local and global models. Alternatively, Output DP can be applied after aggregation on the global model, and then securing only this global model from leakage. Output DP can be also applied at a post-processing stage (see Figure 3), to mitigate risks of leakages through a global model that is distributed to outside parties.

In Federated Learning, where data is distributed, the stage in the learning process where Differential Privacy is applied can be further distinguished depending on the requirements of data owners and a threat model. In Table 1, we use Local, Central and Hybrid DP to address corresponding threats.

Central Differential Privacy (Central DP) (McMahan et al., 2018; Naseri et al., 2022; Yang et al., 2023) is applied when users trust the data aggregator, who consequently applies DP only on the global model to protect the privacy of users when the global model is shared for public usage. Therefore, central DP mitigates risks of inference attacks from a global model (see Table 1). Central DP can be achieved e.g. by applying output perturbation to the global model in FL, or by training differentially-private local models (e.g. via DP-SGD).

Local Differential Privacy (Local DP) (Kasiviswanathan et al., 2011) refers to the case when an aggregator can **not** be trusted and each party wants to protect their data (local models) before sending it to the aggregator (e.g. performing input perturbation or training of differentially-private local models). As local DP is stricter than central DP, it usually results in a more significant drop in the utility of the model. Due to the sequential composition property, local DP also satisfies central DP and mitigates inference attacks on both local and global models (see Table 1).

Distributed Differential Privacy (Distributed DP) (Agarwal et al., 2021; Kairouz et al., 2021) aims to achieve the utility of central DP, but without having to trust the central aggregator. Distributed DP can be achieved by using e.g. *secure aggregation protocols* like SMPC, which protects local models from inference attacks on the server side. At the same time, to protect global models from inference, either (i) output DP can be applied on global model after aggregation achieving central DP or (ii) clients can apply DP on local models with noise enough to achieve central DP (what by DP composition property makes a global model differentially private).

Hybrid Differential Privacy (Hybrid DP) (Avent et al., 2017; Beimel et al., 2019) considers scenarios when different clients have different privacy requirements or restrictions: while some of them may desire to have local DP guarantees, for others, central DP or no DP at all are viable options. In this case, the utility of the global model can be significantly improved.

Integration of Differential Privacy in FeatureCloud:

In FeatureCloud, we ensured that users can apply DP at any preferred stage to satisfy the necessary privacy requirements including central, local, distributed and hybrid DP.

Input DP can be applied as a pre-processing app in FeatureCloud (see Figure 3) to protect data from identity and membership disclosure as well as protecting local and global models from inference attacks, therefore, providing **central, local or hybrid DP.**

DP-SGD is available as an analysis app in the FeatureCloud App Store (**Logistic Regression with DP-SGD**) and can be applied to train differentially private logistic regression models locally by clients in federated learning (see Figure 3). To apply DP-SGD to other machine learning algorithms utilising SGD as an optimiser, app developers can use the existing app as a template. DP-SGD can be used to achieve **local, central and hybrid DP**.

Output DP is implemented as a general function available and can be applied to all feature cloud learning algorithms, as is the case for e.g. SMPC. The user needs to specify values for the sensitivity and the epsilon to achieve differential privacy through output perturbation. Application of output DP on locally trained models can be used to ensure **central, local or hybrid DP,** application of output DP on global models can ensure only **central DP**.

To achieve **distributed DP**, FeatureCloud users can combine SMPC with any of the DP approaches ensuring that epsilon is set to achieve central DP. In this setting, SMPC will ensure the security of local models against inference attacks, while central DP will mitigate the risks of inference from global models.

Differential Privacy Cost. By adding noise, DP inevitably causes a loss of utility of machine learning models. One always has to acknowledge the trade-off between privacy and utility when using DP. The choice of the privacy loss parameter epsilon is essential for a meaningful usage of differential privacy, as too high values of epsilon can result in practically non-private models and data (Dwork et al., 2019). There is no consensus in the scientific community or industry on how to choose epsilon resp. which values are considered to lead to sufficient privacy. DP, however, is already actively applied in different industries, which use quite a wide range for epsilon parameter[14]. In one of our works (Pustozerova et al., n.d.), we focused on analysing the utility loss when using differential privacy in federated learning. We compared output perturbation and DP-SGD approaches. We found that DP-SGD provides a better privacy-utility trade-off than output perturbation in most of the considered scenarios, in the settings when local DP has to be achieved.

5.4.4 Data Exfiltration Attacks

The acquisition of training data, a basis for building effective machine learning models, often involves the investment of significant resources, making the data valuable. Moreover, the confidential nature of non-public, sensitive data, such as medical data, contributes to its value. The application of (third-party) machine learning algorithms on such data can lead to issues concerning data privacy and confidentiality. Even though the training data is not shared in a federated learning setting, the sharing of models between the participants or with the aggregator makes the process susceptible to the use of a machine-learning model as a medium (carrier) to transfer information, such as parts of the training data.

Data exfiltration attacks aim to extract information about the training data from a protected site, in our scenario using a machine learning model. In these attacks, the attacker first hides the desired information within the model itself. Once they gain access to the trained model, the attacker can subsequently reconstruct the information. To perform this attack, the adversary adds malicious functionality to the code that is provided for training a machine learning model. The code uses steganographic techniques to encode information into the model. There are two ways for an attacker to use a machine learning model for hiding, depending on the assumed type of access to the trained model the attacker gains: a white-box scenario requires access to the model parameters, while a black-box scenario requires only access to a prediction API. In the white-box scenario, the malicious code abuses the **parameter capacit**y of the model at hand, and hides the information into (a) the parameter values by correlating the data to the parameter values (Correlated Value Encoding Attack), (b) the least significant bits of the parameters (LSB Encoding attack), or (c) the signs of the parameters (Sign-Encoding Attack). In the black-box scenario, the attacker abuses the **learning capacity** of the model, where the model is, in addition to the benign training data, trained on a malicious dataset. Once an API is queried using the same malicious dataset, the attacker can reconstruct the data from the outputs of the model.
These attacks are therefore able to exfiltrate the training data even in an isolated computing environment.

[14] https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy-future-work-open-challenges

In Figure 3, one can see that a data exfiltration attack has to be executed in two stages in federated learning: “Data exfiltration: embedding” and “Data exfiltration: extraction”. The embedding or encoding of the data into the model is executed during local model training by a malicious algorithm. The extraction of the encoded data happens when an attacker gets access to the model. In federated learning, we consider the attacker may get access to the distributed global model after the post-processing stage.

Data exfiltration mitigations

As data exfiltration attacks use machine learning models as information carriers, the defences against such attacks aim to remove the information hidden inside machine learning models. One of the major points to consider when designing the defences is their effect on the model's effectiveness. The goal is to preserve the effectiveness as much as possible while removing as much hidden information from the model as possible. As several different attack methods are possible, we also need consider a range of defences; these can also be applied in sequence.

LSB Sanitization

When an attack abuses the parameter capacity, directly modifying the parameters would modify and potentially remove the hidden information, so that an attacker is unable to extract it. The so-called parameter sanitization (or bit pruning) replaces bits of parameters, where information could potentially be hidden. We showed empirically that in many settings, a large number of bits can be sanitised before the model's performance deteriorates notably.

Sign Modification

Similarly to the LSB sanitization, this defence aims to remove the information hidden in the parameters of a machine learning model by modifying them. It serves to defend a model against a Sign-Encoding attack, in which an attacker uses the signs of the trainable model parameters for data hiding. The signs of the parameters are therefore modified by the defence. The sign represents the most-significant bit of the parameter. It is thus important to note that with an increasing number of modified parameters, the performance of the model on the original task declines. Therefore the trade-off between defence utility and model utility needs to be considered. In order to attempt to minimise the impact of the parameter modification on the model performance, it is first determined in which sequence the parameters will be modified by the defence.

Parameter Based Neuron Pruning

In case the learning capacity of the model is abused, pruning the nodes resp. connections in the network leads to the model 'forgetting' the learned information, rendering the attack ineffective. In the defence, the objective is to remove the learned information about the trigger set, on which the model was trained by the attacker. The defender chooses the percentage of the connections to be pruned from the network, and the values of the parameters of the trained model are the deciding factor for which respective connections are pruned. Once the attacker receives the output of the network, they are unable to reconstruct the hidden information.

Integration of data exfiltration defences in FeatureCloud:

As model parameters are shared in a federated learning setting, white-box attacks can be carried out in this scenario. Specifically, the Least Significant Bit Encoding attack, as it offers the highest capacity (when attacks on models of the same size are compared), is of relevance to attackers. We therefore implement a Parameter Sanitization defence against such attack as an application within the FeatureCloud framework - the **LSB Sanitization App**. This defence is intended for use on the

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

aggregated (global) model. As seen in Figure 3, by post-processing a trained model, we aim to remove the information hidden during the training process. This application takes a machine learning model as an input and requires the user (defender) to select the magnitude of sanitization by determining the amount of bits to be sanitised. The resulting sanitised model is then distributed and can subsequently be used for inference. The attacker would normally only get access to the model following the distribution, at which point the exfiltration capability is reduced or fully removed.

We further implement a defence against a Sign-Encoding attack - the **Sign Modification App.** This app provides a post-processing step to be applied on a trained model. In an effort to try to lessen the impact of the defence on the model utility, the sequence in which the signs of the parameters are modified is determined based on additional information. This sequence is based on the values of the parameters of the trained model. It means that the parameters with the absolute value closest to zero are modified first in a manner that does not allow the attacker to infer the nature of the modification, so that the attacker is not able to simply flip the signs of the parameters to still reconstruct the data. The user (defender) chooses the percentage of the parameters to be modified.

To fend off an attack in the black-box scenario, where an attacker abuses the learning capacity of a model, we implement the **Parameter Based Neuron Pruning App.** This defence also represents a model post-processing step. It aims to remove the learned representation of the trigger set, essentially attempting to make the model 'forget' the trigger set previously used by the attacker in order to exfiltrate the training data. The defence identifies the parameters with the lowest value. Therefore, similarly as above, the user (defender) the percentage of parameters that should be pruned from the network as an input to the app. The trade-off between model and defence utility needs to be considered in this scenario as well.

Cost of data exfiltration defences

As information is hidden directly in the machine learning models, the models themselves need to be altered in order to remove this information. This modification will therefore cause a change in model utility. The extent of this impact depends on the strength of the defence. It is important to consider that the effectiveness of an attacked model will be reduced compared to a benign model. Applying a defence on top of an attacked model then further reduces the effectiveness on the benign task.

In the black-box scenario, as a defence aims to remove the learned representation of the trigger set, the benign task can also be affected by the pruning. The magnitude of the impact on the benign task depends mainly on the amount of neurons pruned from the network.

The computational resources required to perform the pruning are not high compared to the training of the model, as only a forward pass of the benign data through the network to compute the activations and the subsequent removal of the nodes is required.

In the white-box scenario, the parameters of the models are directly modified, which leads to an altered behaviour of the model on the benign task. Specifically with the LSB Sanitization, the model effectiveness deteriorates with an increasing number of modified LSBs. In this scenario however, if the number of bits used for the defence is smaller or equal to the number of bits used for information hiding, the further impact on model utility caused by applying the defence will not be substantial. With the Sign Modification defence, the model utility deteriorates with an increasing number of parameters modified. We therefore aim to first modify the parameters that are likely to have the least negative impact on the model effectiveness on the original task. The user (defender) controls the number of parameters modified in the model, which directly relates to the trade-off

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

between model and defence effectiveness. Furthermore, this defence does not require considerable computational resources, as it only modifies the values of the parameters.

5.4.5 Model stealing attacks and unauthorised model re-distribution

Machine learning models, trained in a federated setting on non-public data, are owned by the participants of a federated learning process and can be considered their intellectual property (IP). However, these models can be illegally copied, leading to a violation of the owners' IP rights. Such model theft can happen in two scenarios. In the first case, a model is published, for instance, as a service on the Internet. A malicious client can then send unlabelled data to the model and use its predictions to create a dataset for training a so-called substitute model that aims to reproduce the functionality of the original model. In this case, the model becomes a target of a so-called **model stealing attack** (Tramèr et al., 2016). Within FeatureCloud, we conducted the first comprehensive survey on model stealing attacks and defences (Oliynyk et al., 2023), which served as a base for analysing potential stealing threats in the project. Another scenario assumes that a person with access to the model shares it with a third party without permission. We call this scenario **unauthorised model re-distribution**. In both cases, an illegal copy of the model is created, and the intellectual property rights are violated.

As shown in Figure 3, the threat of model copy appears in the final stage of the federated learning process, after a global model is distributed. In case the model becomes publicly available, i.e. its architecture and learned parameters become known, there are obviously no more risks of model theft. However, if the model is kept private or shared as a black box, model stealing attacks or unauthorised model re-distribution can lead to intellectual property rights violations.

Model IP protection by model watermarking

To analyse and define the most feasible countermeasures to the aforementioned threats, we conducted exhaustive research on the IP protection of machine learning models (Lederer et al., 2023). One of the most promising defence approaches that can mitigate both model stealing and unauthorised model re-distribution is proving the ownership of a model. In case a copy of the original model is published, one can use an ownership verification technique to prove that the model was stolen. The most common and well-studied ownership verification defence is **model watermarking** (Uchida et al., 2017). The main idea is to embed imperceptible information (watermark) into a model. If later the model is illegally copied, one can prove that by revealing the embedded watermark in the copied model. Model watermarking works in both white-box and black-box settings, meaning that it is enough to get access to only outputs of a stolen model to prove that a theft has happened.

Integration of Model Watermarking in FeatureCloud

In FeatureCloud, we provide an app that embeds a watermark into neural networks: **NN Watermarking**. Watermarking is done in a black-box manner by fine-tuning a neural network on a so-called trigger set, which comes out of the distribution of the original training data. This trigger set together with predefined (usually not meaningful) labels is also used for watermark verification. If a model achieves a certain accuracy threshold while being evaluated on the trigger set, this model is considered a copy of the original model. As shown in Figure 3, watermarking should be applied to the final model before it will be distributed, and after any other post-processing steps, to avoid the risk of damaging the watermark.

Cost of Model Watermarking

In order to apply model watermarking, one needs to define the following:

1. Trigger set size, labels, and distribution
   The trigger set size should be significant enough to prove that the predicted labels were

embedded on purpose. At the same time, having too much data in the trigger set may lead to utility loss of the model, as it will be forced to memorise more information from the trigger set. Hence, we recommend defining trigger set size based on the size of the original training data. Another aspect is to set labels for the trigger images. As the trigger set contains out-of-distribution samples, labels from the original training set are not meaningful. One has to decide which subset of labels have to be used and which trigger images receive which label. Finally, as stated above, the trigger set is assumed to come from a different distribution compared to the original training data of the model. However, in the FeatureCloud app, we support different dataset options, so that the source of the trigger set can also be varied.

2. Model utility loss and watermark accuracy
   As the model is fine-tuned on the trigger set, it might "forget" knowledge obtained previously from the federated learning. This effect can be mitigated by reducing the size of the trigger set or setting lower requirements for the accuracy of the watermark (i.e. the model's performance on the trigger set). Hence, one needs to define to which extent the utility of the model can decrease and what is the required accuracy threshold for watermark recognition.

5.4.6 Threats and Defences Timeline

Figure 3 shows the timeline of data analysis processes provided on the FeatureCloud platform in the FeatureCloud AI App Store[15]. In our threat analysis, we focus on the three stages of processes and application types: Pre-processing, Analysis and Post-processing. We skip the "Evaluation" apps, as they are executed locally by clients, and also the outputs are stored locally.
We provide an overview to demonstrate at which state a particular attack can be executed and at which stage one should use mitigations and defence mechanisms to secure from privacy attacks. The order of application of the defences is crucial to ensure a meaningful execution and privacy protection.

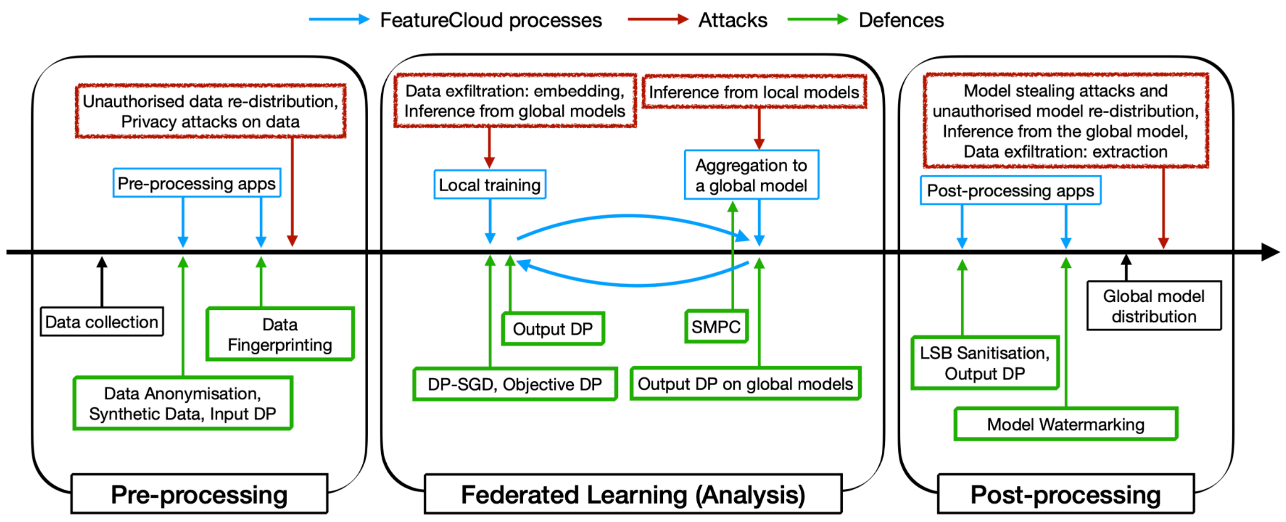
FeatureCloud processes
Attacks
Defences
Unauthorised data re-distribution, Privacy attacks on data
Pre-processing apps
Data collection
Data Fingerprinting
Data Anonymisation, Synthetic Data, Input DP
Pre-processing
Data exfiltration: embedding, Inference from global models
Inference from local models
Local training
Aggregation to a global model
Output DP
SMPC
DP-SGD, Objective DP
Output DP on global models
Federated Learning (Analysis)
Model stealing attacks and unauthorised model re-distribution, Inference from the global model, Data exfiltration: extraction
Post-processing apps
Global model distribution
LSB Sanitisation, Output DP
Model Watermarking
Post-processing

Figure 3. Threats and Defences Timeline.

[15] FeatureCloud App Store https://featurecloud.ai/app-store?view=store&sub=&q=&r=0&uc=1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Pre-processing applications for defending against attacks include anonymisation, synthetic data generation, differential privacy through input perturbation (Input DP), as well as data fingerprinting. Data fingerprinting has to be performed **after** preprocessing apps (such as scaling), as otherwise, there is a risk of damaging the embedded data fingerprint by these other preprocessing apps. Mitigations and defences applied at this stage aim to secure sensitive local data from privacy attacks and unauthorised data redistribution.

Analysis applications in the FeatureCloud platform include different federated learning algorithms. These algorithms consist of local model training and aggregation of local models into a global model. At this stage, we consider privacy threats like data exfiltration attacks and inference attacks on local and global models. Data exfiltration happens in two steps and stages: (1) when an attacker uses a malicious training algorithm to encode training data into e.g. model parameters during local model training; and (2) when an attacker extracts the data encoded into the model from the global model. To defend against data exfiltration, FeatureCloud apps at the post-processing stage provide techniques like LSB Sanitisation, to mitigate the risks of exfiltrated data extraction after global model distribution.

For privacy threats originating from the sharing of local and global models during federated learning, we use mitigations based on differential privacy and SMPC. DP-SGD and objective DP are applied during local model training. In FeatureCloud, they can be used as specific apps executing local training with an additional DP layer. DP through output perturbation and SMPC are available in FeatureCloud as an app template for the API. Output DP is applied on the trained local model and before the aggregation step. SMPC allows secure aggregation of local models. Output DP on a global model can be applied after aggregation, or after the whole federated learning process on the final global model.

Post-processing applications include defences against data exfiltration attacks, as mentioned in the previous paragraph. As the last step of post-processing before sharing the final (resulting) global model with other parties, one can employ model watermarking to mitigate the risks of model stealing attacks and model re-distribution.

6 Conclusion

In this deliverable, we provided a comprehensive overview of the security of different components of the FeatureCloud platform. We discussed the security of architecture and which practices we used during platform development to satisfy state of the art security standards. We analysed the KPIs set in the Deliverable D2.2 “KPIs and metrics for local execution platforms" and demonstrated that we were able to meet the set targets for each of the KPIs. We also discussed security and privacy aspects of the consent management component. At last, we provided an extensive overview of potential privacy threats and guidelines on how to use the mitigations provided within FeatureCloud platform to address these threats.

This analysis was accompanied also by a stress and security test of the platform, to check for vulnerabilities in the architecture and implementation. Detailed results are provided in Deliverable D7.7 Report on implementation of assessment, requirement criteria, and “stress testing”.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

7 References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L., 2016. Deep Learning with Differential Privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS ’16. Association for Computing Machinery, New York, NY, USA, pp. 308–318. https://doi.org/10.1145/2976749.2978318

Agarwal, N., Kairouz, P., Liu, Z., 2021. The Skellam Mechanism for Differentially Private Federated Learning, in: Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 5052–5064.

Avent, B., Korolova, A., Zeber, D., Hovden, T., Livshits, B., 2017. {BLENDER}: Enabling Local Search with a Hybrid Differential Privacy Model. Presented at the 26th USENIX Security Symposium (USENIX Security 17), pp. 747–764.

Beimel, A., Korolova, A., Nissim, K., Sheffet, O., Stemmer, U., 2019. The power of synergy in differential privacy: Combining a small curator with local randomizers [WWW Document]. URL https://www.semanticscholar.org/paper/The-power-of-synergy-in-differential-privacy%3A-a-Beimel-Korolova/6f1b85f3a4064505d2d982acb3d206b7691732ca (accessed 8.11.23).

Chaudhuri, K., Monteleoni, C., 2009. Privacy-preserving logistic regression, in: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems 21. Curran Associates, Inc., pp. 289–296.

Chaudhuri, K., Monteleoni, C., Sarwate, A.D., 2011. Differentially Private Empirical Risk Minimization. J. Mach. Learn. Res. JMLR 12, 1069–1109.

D, S., S, M., J, D.-F., J, S.-C., M, B., 2020. μ -ANT: semantic microaggregation-based anonymization tool. Bioinforma. Oxf. Engl. 36. https://doi.org/10.1093/bioinformatics/btz792

Dankar, F.K., Ibrahim, M.K., Ismail, L., 2022. A Multi-Dimensional Evaluation of Synthetic Data Generators. IEEE Access 10, 11147–11158. https://doi.org/10.1109/ACCESS.2022.3144765

Dwork, C., 2006. Differential Privacy, in: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.), Automata, Languages and Programming, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 1–12. https://doi.org/10.1007/11787006_1

Dwork, C., Kohli, N., Mulligan, D., 2019. Differential Privacy in Practice: Expose your Epsilons! J. Priv. Confidentiality 9. https://doi.org/10.29012/jpc.689

Dwork, C., Roth, A., 2014. The Algorithmic Foundations of Differential Privacy. Found. Trends® Theor. Comput. Sci. 9, 211–407. https://doi.org/10.1561/0400000042

Evans, D., Kolesnikov, V., Rosulek, M., 2018. A Pragmatic Introduction to Secure Multi-Party Computation. Found. Trends® Priv. Secur. 2, 70–246. https://doi.org/10.1561/3300000019

Figueira, A., Vaz, B., 2022. Survey on Synthetic Data Generation, Evaluation Methods and GANs. Mathematics 10, 2733. https://doi.org/10.3390/math10152733

Fukuchi, K., Tran, Q.K., Sakuma, J., 2017. Differentially Private Empirical Risk Minimization with Input Perturbation, in: Yamamoto, A., Kida, T., Uno, T., Kuboyama, T. (Eds.), Discovery Science, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 82–90. https://doi.org/10.1007/978-3-319-67786-6_6

Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N., 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS ’18. Association for Computing Machinery, New York, NY, USA, pp. 619–633. https://doi.org/10.1145/3243734.3243834

Giomi, M., Boenisch, F., Wehmeyer, C., Tasnádi, B., 2022. A Unified Framework for Quantifying Privacy Risk in Synthetic Data.

Hidano, S., Murakami, T., Katsumata, S., Kiyomoto, S., Hanaoka, G., 2017. Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-Sensitive Attributes, in: 2017 15th Annual Conference on Privacy, Security and Trust (PST). Presented at the 2017 15th Annual Conference on Privacy, Security and Trust (PST), pp. 115–11509.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

https://doi.org/10.1109/PST.2017.00023

Hittmeir, M., Ekelhart, A., Mayer, R., 2019. On the Utility of Synthetic Data: An Empirical Evaluation on Machine Learning Tasks, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19. Association for Computing Machinery, New York, NY, USA, pp. 1–6. https://doi.org/10.1145/3339252.3339281

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P., Zhang, X., 2022. Membership Inference Attacks on Machine Learning: A Survey. ACM Comput. Surv. 54. https://doi.org/10.1145/3523273

Jarin, I., Eshete, B., 2022. DP-UTIL: Comprehensive Utility Analysis of Differential Privacy in Machine Learning, in: Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, CODASPY '22. Association for Computing Machinery, New York, NY, USA, pp. 41–52. https://doi.org/10.1145/3508398.3511513

Kairouz, P., Liu, Z., Steinke, T., 2021. The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation. Presented at the International Conference on Machine Learning.

Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A., 2011. What Can We Learn Privately? SIAM J. Comput. 40, 793–826. https://doi.org/10.1137/090756090

Khan, M.S.N., Buchegger, S., 2023. The Impact of Synthetic Data on Membership Inference Attacks, in: Arief, B., Monreale, A., Sirivianos, M., Li, S. (Eds.), Security and Privacy in Social Networks and Big Data, Lecture Notes in Computer Science. Springer Nature, Singapore, pp. 93–108. https://doi.org/10.1007/978-981-99-5177-2_6

Lederer, I., Mayer, R., Rauber, A., 2023. Identifying Appropriate Intellectual Property Protection Mechanisms for Machine Learning Models: A Systematization of Watermarking, Fingerprinting, Model Access, and Attacks. IEEE Trans. Neural Netw. Learn. Syst. 1–19. https://doi.org/10.1109/TNNLS.2023.3270135

McMahan, H.B., Ramage, D., Talwar, K., Zhang, L., 2018. Learning Differentially Private Recurrent Language Models. Presented at the International Conference on Learning Representations.

Naseri, M., Hayes, J., De Cristofaro, E., 2022. Local and Central Differential Privacy for Robustness and Privacy in Federated Learning. Proc. 2022 Netw. Distrib. Syst. Secur. Symp. https://doi.org/10.14722/ndss.2022.23054

Oliynyk, D., Mayer, R., Rauber, A., 2023. I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. ACM Comput. Surv. 55, 1–41. https://doi.org/10.1145/3595292

Pustozerova, A., Baumbach, J., Mayer, R., n.d. Analysing Utility Loss in Federated Learning with Differential Privacy, in: International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE. Presented at the International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE. https://doi.org/0.1109/TrustCom60117.2023.00167

Pustozerova, A., Mayer, R., 2020. Information leaks in federated learning, in: Workshop on Decentralized IoT Systems and Security (DISS) 2020. https://dx.doi.org/10.14722/diss.2020.23004

Samarati, P., Sweeney, L., 1998. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression.

Šarčević, T., Mayer, R., 2019. An Evaluation on Robustness and Utility of Fingerprinting Schemes, in: Machine Learning and Knowledge Extraction. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-29726-8_14

Šarčević, T., Molnar, D., Mayer, R., 2020. An Analysis of Different Notions of Effectiveness in k-Anonymity, in: Domingo-Ferrer, J., Muralidhar, K. (Eds.), Privacy in Statistical Databases, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 121–135. https://doi.org/10.1007/978-3-030-57521-2_9

Shokri, R., Stronati, M., Song, C., Shmatikov, V., 2017. Membership Inference Attacks Against Machine Learning Models, in: 2017 IEEE Symposium on Security and Privacy (SP). Presented at the 2017 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, USA, pp. 3–18. https://doi.org/10.1109/SP.2017.41

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T., 2016. Stealing Machine Learning Models via Prediction APIs, in: 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, Austin, TX, pp. 601–618.

Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S., 2017. Embedding Watermarks into Deep Neural Networks, in: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. Presented at the ICMR '17: International Conference on Multimedia Retrieval, ACM, Bucharest Romania, pp. 269–277. https://doi.org/10.1145/3078971.3078974

Yang, Y., Hui, B., Yuan, H., Gong, N., Cao, Y., 2023. {PrivateFL}: Accurate, Differentially Private Federated Learning via Personalized Data Transformation. Presented at the 32nd USENIX Security Symposium (USENIX Security 23), pp. 1595–1612.

Zhao, B.Z.H., Agrawal, A., Coburn, C., Asghar, H.J., Bhaskar, R., Kaafar, M.A., Webb, D., Dickinson, P., 2021. On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models. 2021 IEEE Eur. Symp. Secur. Priv. EuroSP 232–251. https://doi.org/10.1109/EuroSP51992.2021.00025

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.