MSE | MASTER OF SCIENCE IN ENGINEERING

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Master of Science HES-SO in Engineering
Av. de Provence 6
CH-1007 Lausanne

Master of Science HES-SO in Engineering

Orientation: Computer science - Cybersecurity

Towards Secure Medical Data Processing: Trusted Execution Environments and Remote Attestation

Realized by

Benjamin Mouchet

Directed by
Prof. Marcelo Pasin
HE-Arc

External expert Dr. Jämes Ménétrey, Cysec

Neuchâtel, HES-SO//Master, 2024-25

This page intentionally left blank

Contents

Acknowledgements

I would like to express my gratitude to my professor, Marcelo Pasin, for his guidance throughout this project, particularly in its initial context. His support was essential, both for his expertise in the field of Trusted Computing and for his sound advice, which enabled me to overcome the challenges I encountered.

I would also like to thank Dr. Jämes Ménétrey for his demonstration of WAMR and TWINE, and for sharing his invaluable expertise, which greatly contributed to the technical direction of this work.

Finally, my deepest love goes to my partner, my family and my friends, whose unfailing support has been a precious source of motivation throughout this adventure.

Nomenclature

AC	Access Control
ACL	Access Control List
API	Application Programming Interface
BIOS	Basic Input/Output System
CIA	Confidentiality, Integrity, Availability
CDS	Conference of Cantonal Ministers of Public Health
CPU	Central Processing Unit
DI	Disability Insurance
DoS	Denial of Service
DP	Differential Privacy
DVFS	Dynamic Voltage and Frequency Scaling
EdDSA	Edwards-curve Digital Signature Algorithm
EMFI	Electromagnetic Fault Injection
EPR	Electronic Patient Record
FADP	Federal Act on Data Protection
FDHA	Federal Department of Home Affairs
FHE	Fully Homomorphic Encryption
FHIR	Fast Healthcare Interoperability Resources
FINMA	Swiss Financial Market Supervisory Authority
FMH	Foederatio Medicorum Helveticorum
FOPH	Federal Office of Public Health
GPU	Graphics Processing Unit
HL7	Health Level Seven
IoT	Internet of Things
JSON	JavaScript Object Notation
LAMal	Federal Law on Health Insurance
MA	Mutual Attestation
MEE	Memory Encryption Engine
MFA	Multi-Factor Authentication
ML	Machine Learning
mTLS	mutual TLS
nFADP	new Federal Act on Data Protection
NIST	National Institute of Standards and Technology
NoSQL	Not only SQL
OS	Operating system
OWASP	Open Worldwide Application Security Project
PHE	Partial Homomorphic Encryption
PII	Personally Identifiable Information
PoLP	Principle of Least Privilege
RATS	Remote ATtestation procedureS
RLHF	Reinforcement Learning from Human Feedback

RBAC	Role Based Access Control
RFC	Request For Comments
ROM	Read-Only Memory
RSA	Rivest–Shamir–Adleman
SEV	Secure Encrypted Virtualization
SGX	Software Guard Extensions
SHA-3	Secure Hash Algorithm 3
SHE	Somewhat Homomorphic Encryption
SQL	Structured Query Language
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of privilege
TEE	Trusted Execution Environment
TLS	Transport Layer Security
VM	Virtual machine
XML	Extensible Markup Language
WAMR	WebAssembly Micro Runtime

Abstract

The digitization of the Swiss healthcare sector poses major challenges in terms of protecting, accessing and sharing medical data, particularly with the introduction of electronic patient records. The need to guarantee the confidentiality and integrity of sensitive information, while allowing it to be used by legitimate parties such as insurance companies and healthcare professionals, raises complex issues.

In this context, Trusted Execution Environments (TEEs) appear to be a promising solution. This project explores the integration of TEEs into the digital health ecosystem in Switzerland through a proof of concept based on the Remote attestation procedures (RATS) Architecture RFC. The aim is to demonstrate how an actor can make a decision with complete confidence without directly accessing the data, but only by verifying the authenticity of the result returned by a TEE. A set of approved methods are available in the form of MongoDB pipelines. An analysis of the time taken by the various sections shows that the pipelines are the most time-consuming actions and are to be further optimized.

An analysis of the thread model based on a STRIDE approach shows that confidentiality, integrity and non-repudiation are currently well respected in the implementation. However, threats remain, particularly related to authentication, privilege escalation and denial of service (DoS) attacks, which will need to be addressed in future development. Additional approaches to improving the project have been raised, such as the use of machine learning and differential privacy to help protect data.

Disclaimer: The main body of this report was written in French before being translated using Deepl.

Key Words: TEE, Remote Attestation, Mutual Attestation, Background Check, Electronic Patient Record

1. Introduction

The digitization of the medical sector has transformed the management of health data, making it easier to store, share and analyze. In Switzerland, this transition brings with it major challenges in terms of security and confidentiality due to the strict legal framework imposed by the new Federal Act on Data Protection (nFADP) and the specific requirements linked to the Electronic Patient File (EPR). These constraints are designed to ensure that sensitive medical information remains protected while still being accessible by authorized healthcare providers. However, the technical implementation of these guarantees remains a complex challenge.

Accessing and sharing medical data raises a central issue: *how to ensure the confidentiality and integrity of data while allowing controlled and secure access to legitimate parties*? Traditional solutions, such as database encryption and access control systems, are effective but have their limitations. Once the data has been decrypted for use, it becomes vulnerable to attack, particularly if the execution environment is compromised. This problem is exacerbated in cloud environments where machines and data are potentially accessible to other tenants or even the provider. In addition, the growing need for interoperability and secure sharing means that these approaches are inadequate in the face of new threats and regulatory requirements.

It is in this context that Trusted Execution Environments (TEEs) are emerging as a promising solution. TEEs, such as Intel SGX, ARM TrustZone or AMD SEV, enable sensitive operations to be carried out in a secure environment, isolated from the rest of the system. Combined with remote attestation mechanisms, such as those defined in the Remote attestation procedures (RATS) Architecture RFC, they offer strong guarantees of data integrity and confidentiality, even on potentially unreliable infrastructures such as a cloud.

This work explores the integration of TEEs into a medical data management system by designing a proof of concept. This is based on the use of a TEE acting as a proxy for a medical database. The data is protected by secure pipelines, an access model based on an access control list (ACL) and remote attestations. The aim is to demonstrate how an actor can make a decision without having direct access to the data, but by relying solely on cryptographic guarantees. This report begins by looking at the digital environment for medical data in Switzerland, followed by a state-of-the-art review of the various technologies available in the field. Next, the design and implementation of the proof of concept are detailed, explaining how it works and the technological choices made. Finally, the project is evaluated using quantitative and qualitative measurements based on a threat model before giving additional approaches to improving the project. Such as the use of machine learning and differential privacy to help protect data.

2. Background

2.1 Medical data management challenges in Switzerland

Storage, management and sharing of medical data are always more topical challenges in an increasingly digital society. It's all about meeting the needs of patients, healthcare providers and other federal bodies, especially as medical data represents a person's most intimate information. The protection and proper use of this data is therefore an absolute priority.

Supervision of this data by the owner and its restricted sharing are expected by the public, as shown by a survey carried out in California in 2014[1]. Nearly half of those questioned expressed fear about the exchange of medical data and the consequences of a data leak that would compromise the protection of privacy. Despite these fears, they remain in favor of electronic data sharing, while stressing the importance of and need for transparency regarding individual control, access authorization and what use is made of the data. Another conclusion of the study was that respondents were more in favor of sharing de-identified data for research purposes than personal data for healthcare purposes.

A second study carried out in Switzerland in 2018[2] aimed to gauge the ethical awareness of healthcare stakeholders regarding the large patient datasets in Switzerland. What emerged was a concern about the need for standards for data collection and use. There was less interest in issues relating to patient agencies, reciprocity, shared governance or the use of clinical data from shared registries. The authors observed that such asymmetry is likely to create tensions. Data sharing in the interests of public research must consider the point of view of patients and donors. If it does not, there is a risk that it will lead to mistrust of healthcare data collection, reducing the perceived social benefits. The authors propose that new ethical approaches can reinforce both patients' rights and the public interest, rather than setting one against the other, by providing elements for developing ethical reflections on a cooperative approach involving patients in the governance of their health-related data.

Medical data can therefore be a source of fear and tension, particularly due to the presence of numerous actors with varied needs. Patients want to protect their data as much as possible while receiving the best possible care. A medical history shared by different doctors leads to more effective treatment due to a better overview of the patient's medical history. Other actors, such as federal bodies, require medical data (anonymized) for statistical purposes. The following paragraphs will look at the roles of the various actors in the world of healthcare, the key components of IT security and an overview of the Electronic Patient Record (EPR).

2.2 Key Health Actors in Switzerland

In Switzerland, healthcare governance follows a federalist structure, with responsibilities divided among the Confederation, cantons, and communes. The Federal Constitution lays the foundation for healthcare policies, shaping a legal framework that includes over 20 laws and numerous ordinances overseen by the Federal Office of Public Health (FOPH)[3]. Additionally, the system involves private stakeholders, such as insurance companies, healthcare providers, and professional associations, which contribute to the management, regulation, and development of healthcare services alongside public institutions.

[1] KIM, Katherine K., JOSEPH, Jill G., OHNO-MACHADO, Lucila, 2015. *Comparison of consumers' views on electronic data sharing for healthcare and research*.

[2] MOUTON DOREY, Corine, BAUMANN, Holger, BILLER-ANDORNO, Nikola, 2018. *Patient data and patient rights: Swiss healthcare stakeholders' ethical awareness regarding large patient data sets – a qualitative study*.

[3] FHOP, n.d. *Legislation*. https://www.bag.admin.ch/bag/fr/home/gesetze-und-bewilligungen/gesetzgebung.html

Federal Office of Public Health (FOPH)

The Federal Office of Public Health (FOPH) is the Swiss federal government's center for public health and a part of the Swiss Federal Department of Home Affairs. The FOPH plays a central role in shaping and implementing Switzerland's health policies, with a particular focus on digitalization through initiatives such as the "Strategie eHealth Schweiz 2.0 (2018–2024)"[4]. This strategy, developed collaboratively by the Confederation and the cantons, aims to enhance healthcare quality, patient safety, and system efficiency by promoting the widespread adoption of the EPR. It aligns with broader health objectives by encouraging digital literacy and responsible handling of patient data. The strategy comprises objectives organized into three areas of action: fostering digitalization, coordinating and harmonizing efforts across the healthcare system, and empowering individuals with the skills and awareness needed for digital health management. The FOPH provides guidance and financial support for the establishment and certification of healthcare communities and reference communities. Healthcare digitalization is achieved by collaborating with eHealth Suisse, other government bodies, and healthcare stakeholders.

Swiss Financial Market Supervisory Authority (FINMA)

The Swiss Financial Market Supervisory Authority (FINMA) is the Swiss government body responsible for financial regulation. FINMA supervises amongst others health insurers and ensures their data-handling practices align with legal and ethical standards. Working alongside the FOPH, it monitors how medical data is used in claims processing and reimbursement, ensuring that sensitive information is managed responsibly and transparently[5].

Swissmedic, Swiss Agency for Therapeutic Products

Swissmedic is the Swiss authority responsible for authorizing and supervising therapeutic products[6]. It is a public-law body attached to the Federal Department of Home Affairs (FDHA). It is mainly financed by fees and partly supplemented by federal compensation for services of the public interest. As they are responsible, for example, for the authorization of clinical trials on medicines and medical devices, the results and data from these studies are crucial for the development of the healthcare in Switzerland.

Health insurance and insurers

The Swiss healthcare system ensures universal access to medical services. Under the Federal Law on Health Insurance (LAMal)[7], all residents are required to have basic health insurance, which covers essential services such as doctor consultations, hospital stays, and prescribed medications. Insurers cannot deny coverage due to age or pre-existing conditions, promoting equity in access to care. Premiums vary by canton and can be subsidized for individuals with low income[8].

Health insurers operate as private or non-profit entities, regulated by the FOPH and the Swiss Financial Market Supervisory Authority (FINMA). They manage premiums, reimbursements, and risk pools while negotiating service costs with healthcare providers to control expenses. Many insurers also offer supplemental insurance for non-essential services like private hospital rooms or alternative medicine, governed under private law.

eHealth Switzerland

As the operational arm of Switzerland's Cyberhealth initiative, eHealth Switzerland is responsible for implementing national guidelines on data security and interoperability for the EPR. Its role involves integrating modern technologies into the existing infrastructure to promote secure access to medical data, ensuring that all systems adhere to established privacy and security protocols[9].

---

[4] EHEALTH SWITZERLAND, 2018. *Stratégie Cybersanté Suisse 2.0. 2018 –2024*.

[5] FOPH, 2024. *Health insurance: Supervision of insurers*.

[6] SWISSMEDIC, 2019. *Swissmedic, Swiss Agency for Therapeutic Products*.

[7] FOPH, 2024. *Loi fédérale sur l'assurance-maladie (LAMal)*.

[8] FOPH, n.d. *Health insurance*.

[9] EHEALTH SWITZERLAND, n.d. *eHealth Suisse*.

HL7 Switzerland

Health Level Seven (HL7) Switzerland provides the technical foundation for medical data exchange by developing specifications like the FHIR (Fast Healthcare Interoperability Resources) standard. FHIR has become integral to the EPR, enabling seamless and secure data-sharing between healthcare providers, fostering both efficiency and compliance with global standards[10].

Cantons

Cantonal authorities, through their regional public health offices, are tasked with executing public health strategies. They generally manage their own healthcare systems, with oversight provided by the canton's chief medical officer.[11]

Conference of Cantonal Ministers of Public Health (CDS)

The CDS plays a role in harmonizing healthcare initiatives between the cantons and the federal government. By promoting interoperability in systems like the EPR, the CDS helps coordinate standardized data-sharing protocols across the country, enabling healthcare providers to access patient records securely and efficiently[12].

Foederatio Medicorum Helveticorum (FMH)

The FMH represents the collective interests of Swiss physicians, advocating for professional standards and policy-making that align with medical ethics[13]. In the context of medical data, FMH provides recommendations on how data should be shared securely among healthcare professionals while upholding patient confidentiality. It also advocates for ethical use and privacy considerations in medical data policies, ensuring that interoperability systems like the Electronic Patient Record (EPR)[14] can function effectively without compromising sensitive information.

2.3 The current issues with insurance claims

In Switzerland, two primary billing systems govern the interaction between patients, healthcare providers, and insurers:

- **Policyholder system**: The patient pays the healthcare provider directly after receiving the service. The patient then submits proof of payment to their health insurance for reimbursement. This system empowers the patient but can delay reimbursement if disputes arise or if the insurer requests additional information.
- **Insurer system**: The healthcare provider bills the insurer directly. After processing, the insurer reimburses the provider and charges the patient for their portion of the costs (deductible and co-payment). This system simplifies the process for patients but can involve direct interactions between insurers and providers over disputed claims[15].

One significant challenge in the Swiss health insurance system is the refusal of insurers to cover certain treatments. This issue is particularly contentious when insurers require detailed justifications from healthcare providers, which may involve transmitting sensitive medical data. An example given following a conversation with a doctor to illustrate this issue was a work readaptation form required by the disability insurance[16]. This form contains sensitive information that is sent by mail. Not only do these forms increase the administrative work for health providers but this raises ethical and legal concerns:

---

[10] HL7 INTERNATIONAL, n.d. *Norme internationale pour l'échange électronique de données médicales*.

[11] CLEISS, 2021. *Le système de santé suisse.*

[12] CDS, n.d. *La CDS*.

[13] FMH, n.d. *À propos de la FMH*.

[14] FMH, n.d. *Dossier électronique du patient (DEP)*.

[15] FOPH, 2024. *Health insurance: Key points in brief*.

[16] Available as Appendix I.

- **Breach of Confidentiality**: Sharing patient information with insurers can compromise the confidentiality that patients expect in their interactions with healthcare providers. Especially if the communication channels are compromised.
- **Risk of Discrimination**: Insurers may perceive patients with complex or chronic conditions as liabilities. Patients fear their medical data could be used to label them as high-cost risks, potentially influencing their future premiums or coverage when changing insurance provider for instance.

These issues can be mitigated thanks to the Federal Act on Data Protection (FADP)[17]. Here is a sample of principles insurers must adhere to when processing personal and sensitive health data:

- **Medical Necessity and Proportionality** (Art. 6 FADP): Insurers are limited to collecting only data necessary for determining claims or premiums. Medical data provided by policyholders must not exceed what is strictly required.
- **Advisors' Role**: Insurers consult medical advisors to determine whether treatment qualifies for reimbursement. The medical advisor serves as a filter to ensure that insurers only receive essential information for reimbursement decisions, protecting patient confidentiality. This is however limited to the basic health insurance.[18]
- **Transparency** (Art. 19 FADP): Patients must be informed about the purpose of data collection, how it is processed, and who has access to it. Violations of these principles, such as unjustified requests for complete medical records, can undermine trust in the healthcare system and lead to breaches of confidentiality.

Claim rejections are not uncommon and can occur for various reasons. For example, the SAKK (Swiss Group for Clinical Cancer Research) highlights that treatments provided within clinical studies are sometimes denied reimbursement under Article 49 of the LAMal (Federal Health Insurance Act)[19]. This practice raises both legal and ethical concerns, especially when the refusal includes costs that would otherwise be covered outside the research context. Health status can also play a role in supplementary or private daily sickness benefits insurance. Insurers may require detailed health information during admission[20]. While social insurers are required to accept all applicants regardless of health status, private insurers may impose a reserve period of up to five years for pre-existing conditions, potentially leading to discriminatory practices against individuals with chronic illnesses.

These issues illustrate the tension between insurers' cost-containment strategies and the principles of medical confidentiality. The principle of proportionality outlined in the FADP and LAMal is intended to ensure that only essential data is shared. However, its application often hinges on subjective interpretations of what constitutes 'necessity'. While the FADP provides a legal framework with tools and guidelines to minimize risks, it cannot entirely prevent data breaches caused by human error, though it can reduce their likelihood. A graph illustrating the general flow in case of claim is available as Appendix II.

2.4 The state of medical healthcare and IT

2.4.1 Security expectations

The foundational principles and key concepts of cybersecurity are categorized in a set of keywords, like the CIA acronym (Confidentiality, Integrity and Availability). These terms represent critical aspects of safeguarding sensitive information, making them essential for the protection of medical data. To have a better understanding of what these keywords imply, here is an overview in the context of medical data sharing.

---

[17] SWITZERLAND, 2023. *Federal Act on Data Protection*.

[18] FEDERAL DATA PROTECTION AND INFORMATION COMMISSIONER (FDPIC), 2024. *Frequently asked questions on data protection concerns*.

[19] SAKK, 2019. *Avis juridique sur la légalité du refus de prise en charge de certains coûts par les caisses d'assurance-maladie, publié dans Jusletter*.

[20] FDPIC, 2024. *Frequently asked questions on data protection concerns*.

Confidentiality

Measures preventing access to sensitive information from an unauthorized actor. In this context, ensuring that medical data is only accessible to authorized healthcare professionals and the owner of that information, the patient. Encryption mechanisms ensure that medical records are stored and transmitted securely.

Integrity

Guarantees the accuracy and completeness of data during its lifecycle. This ensures that it is not tampered with or corrupted in an unauthorized or undetected manner. The cryptographic operations of checksums and signatures can verify that shared EPRs remain unchanged during data exchanges between healthcare providers.

Availability

Ensures that systems, services, and data are accessible when needed. Redundancy, load balancing, and disaster recovery plans are examples of methods to enhance availability. This also means that the security controls and communication channels must be working correctly. This is particularly useful for EPR access in case of urgent care.

Authentication

Verifies the identity of a user, device, or system before granting access. This can involve passwords, biometrics, tokens, or multi-factor authentication (MFA). The use of electronic identity to be able to use an EPR is a good example of it. This concept is not to be mistaken with Identification. Identification is the assertion of who someone is. Authentication is the act of proving said claim.

Authorization

Once authenticated, the following step is determining what information or action is permitted. This is typically enforced through access control lists (ACLs) or role-based access control (RBAC). The EPR defines for example three levels of confidentiality for the data to limit access to health providers: Normal, Restricted and Secret.

Accountability

Certifies that all actions and accesses are traceable to a unique entity. This is often achieved through logging and audit trails. As exemplified by the EPR specifications, an access journal is available, it tracks any access, whether it being your doctor or the pharmacist verifying a prescription.

Non-repudiation

It guarantees that an actor cannot deny having executed an action because cryptographic proof exists. For example, if a patient grants access to a doctor, this consent can be electronically signed using the patient's key. This means he cannot deny having given access to his data since he is the only one that can emit his signature. This example can go both ways by having a doctor signing a prescription for example.

2.4.2 Cloud and blockchain technologies

Research in the field of sharing medical data proposes the use of blockchain to regulate access to medical data due to the immutable properties of blockchain. This approach guarantees the traceability and transparency of interactions[21]. Smart contracts[22] ensure that only transactions approved by all parties can be used. Another

[21] CHEN, Hannah S., JARRELL, Juliet T., CARPENTER, Kristy A [et al.], 2019. *Blockchain in Healthcare: A Patient-Centered Model*.

[22] YAQOOB, Ibrar, SALAH, Khaled, JAYARAMAN, Raja [et al.], 2022. *Blockchain for Healthcare Data Management: Opportunities, Challenges, and Future Recommendations*.

strength of blockchain is the verification of data without specific data disclosure, thanks to zero-knowledge-proof mechanisms[23]. However, blockchain's high energy demands, particularly with proof-of-work consensus, are a notable drawback. Alternatives like proof-of-stake may reduce this impact. Despite its potential, simpler mechanisms such as hashed chained logs for traceability and trusted operators for decision-making could achieve similar goals without requiring distributed consensus.

Cloud storage services, including Amazon S3, Dropbox, Google Drive, and Microsoft OneDrive, are widely used for backup, archival, and collaboration due to their accessibility, scalability, and ease of use. A key limitation is the trust required in cloud providers to maintain data confidentiality. Encrypting files before upload can mitigate this, but secure key-sharing mechanisms for collaboration remain a challenge.

The use of cloud storage for electronic health records introduces additional concerns about security and privacy. Robust measures like encryption, access controls, and regular audits are essential to protect sensitive patient data. Furthermore, healthcare providers must adhere to strict privacy regulations, including data residency and sovereignty laws in certain jurisdictions. Ensuring compliance requires evaluating cloud providers' privacy policies and data-handling practices to protect patient rights and maintain legal and ethical standards.

2.4.3 Trusted execution environments (TEE)

An effective approach to preserving the integrity and confidentiality of an application is to isolate it. This isolation can be achieved by the operating system through virtual memory mechanisms and privileged processor instructions. When a physical host runs several operating systems, their isolation is ensured by a hypervisor[24] which relies on hardware virtualization instructions interpreted by the processor.

In a multi-tenant environment such as cloud computing, it is necessary to trust both the operating system (OS) and the hypervisor. However, these components are often under the control of the cloud provider, making sensitive data potentially accessible to other tenants or even the provider. This poses a security problem, as it makes it impossible to guarantee total confidentiality of data stored or processed on the shared infrastructure.

Edge computing, essential for Internet of Things (IoT), amplifies these challenges due to its highly distributed nature. Unlike a cloud data center, which can be physically protected, edge equipment (sensors, embedded devices, gateways) is often deployed in open, unsecured environments. Access to sensor data is sometimes via personal devices, such as a smartphone belonging to a technician. The latter, not necessarily trained in security, represents a potential vulnerability that can be exploited by attackers.

Trusted Execution Environment (TEE) offers a practical solution for isolating an application and its data. The aim is to provide a trusted execution environment on untrusted hardware. This trusted environment isolates memory spaces that cannot be accessed or tampered with by higher-privilege software or even by physical control of the device. There is no single definition of a TEE, as implementations vary according to processor manufacturer and use case. There are three main types of TEEs:

- **Partition-Based TEE**: The system is divided into (at least) two partitions, the secure world and the normal world. Isolation is ensured by the processor, which guarantees that no unauthorized access can take place between these two spaces. This model is used to execute cryptographic operations, store sensitive data or generate random numbers securely[25]. ARM TrustZone is an example of a Partition-Based TEE, allowing security functions (e.g. fingerprint authentication[26]) in a protected space, while the main OS continues to run in the unprotected space. However, one drawback of this model is that several different

[23] ZHANG, Wenping, XU, Ruiyun, ZHAO, J. Leon [et al.], 2023. *A Blockchain-centric Data Sharing Framework for Building Trust in Healthcare Insurance*.

[24] Hypervisor, *Wikipedia*. 2025.

[25] ARM, n.d. *ARM TrustZone technology*.

[26] ARM, 2015. *Securing the Future of Authentication with ARM TrustZone-based Trusted Execution Environment and Fast Identity Online (FIDO)*.

functions coexist in the same secure environment, increasing the attack surface and the risk of exploitable vulnerabilities[27].

- **VM-Based TEE**: Virtual machines (VMs) are commonly used to isolate operating systems from their host. VM-Based TEE extends this principle by ensuring that VMs are isolated from each other and from the hypervisor, usually through memory encryption mechanisms. A typical example is AMD SEV (Secure Encrypted Virtualization)[28] which protects data by encrypting the memory of virtual machines, thus preventing unauthorized access. However, a similar disadvantage to Partition-Based TEE is that the secure system hosts a complete OS, as well as other software not directly related to security needs. This can increase the attack surface and complexify security management. Nevertheless, this model is popular because it integrates easily with existing programming paradigms and cloud architecture.
- **Process-based TEE**: An application can isolate certain parts of its code and data within a protected memory space (enclave), without requiring complete hardware separation as with partitioned TEEs. This reduces the attack surface, but is more complex to implement, as in this model the OS is not a trusted environment, and therefore neither are system calls. An example of process-based tee is intel SGX[29].

Although this paper does not define what kind of TEE the proof of concept is based on, let's go into a little more detail about Intel SGX to set the context. As mentioned above, Intel SGX is a process-based TEE introduced in 2015 with the Skylake family of processors. To guarantee the integrity and confidentiality of the code and associated data, the application will create a secure enclave. This protects, for example, against a memory dump from the machine, which will return encrypted data. SGX was developed in response to the increased need for security in cloud environments, to protect against cloud providers, systems administrators and any other malicious actors. Enclaves are stored in a reserved memory space, the Enclave Page Cache[30]. Communications between the CPU and memory are kept confidential thanks to a memory encryption engine (MEE). This MEE also protects against tampering and replay-attack.

The figure below shows the execution flow on SGX compared with AMD SEV (a VM-based TEE). SGX starts by creating an enclave. Then, when the program needs to perform a trusted operation, it invokes a predefined enclave function through an `ecall` (enclave call). This transitions execution from untrusted code (outside the enclave) to trusted code (inside the enclave). To change context, it must pass through the entry point of the enclave, the call gate. The trusted method is then executed by an enclave thread before returning the processed value. If the enclave needs external resources, it calls an `ocall` (outside call). For comparison, SEV executes the entire program in a secure environment. This execution flow is therefore classic: when a function is called, the kernel schedules a thread to execute it before proceeding with its execution. Once it has finished, control reverts to the main thread until the next scheduled execution.

[27] MISRA, Subhas Chandra, BHAVSAR, Virendrakumar C., 2003. *Relationships Between Selected Software Measures and Latent Bug-Density: Guidelines for Improving Quality*.

[28] AMD, n.d. *AMD Secure Encrypted Virtualization (SEV)*.

[29] INTEL, n.d. *Intel® Software Guard Extensions (Intel® SGX)*.

[30] GOETTEL, Christian, PIRES, Rafael, ROCHA, Isabelly [et al.], 2018. *Security, Performance and Energy Trade-offs of Hardware-assisted Memory Protection Mechanisms*.

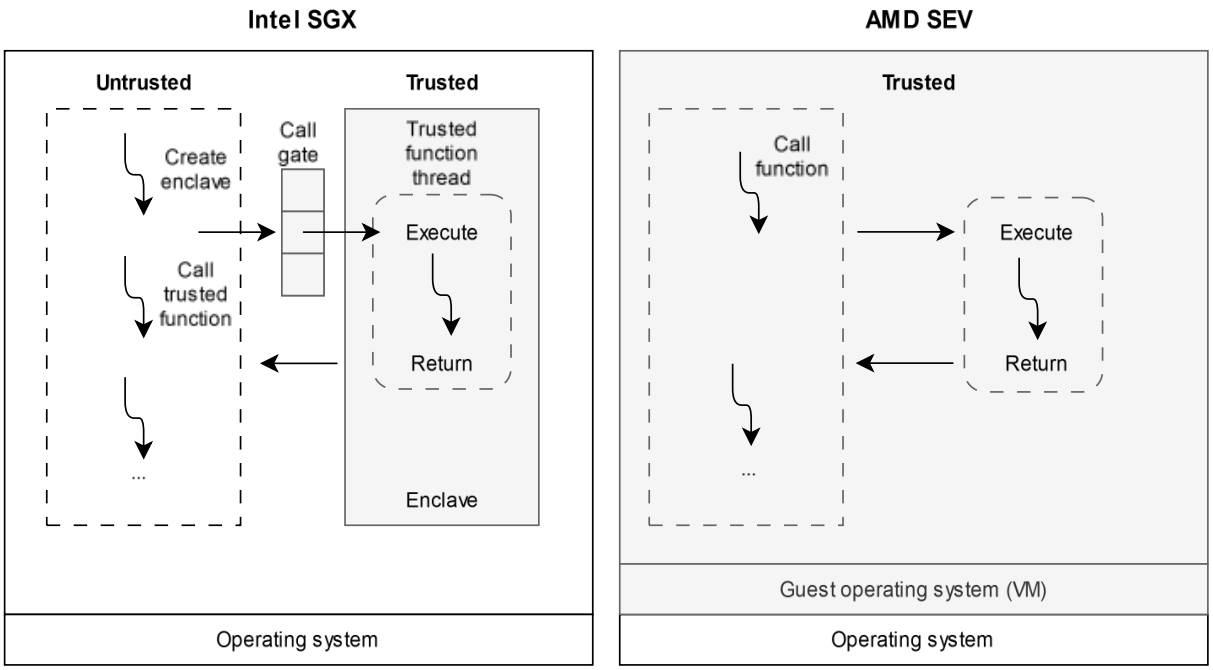


Figure 1: Intel SGX and AMD SEV execution flow principle.
Source: adapted from GOETTEL et al. (2018)

2.4.4 The Electronic Patient Record (EPR)

The Electronic Patient Record (EPR) is a Swiss healthcare initiative launched in 2008 and introduced to the public under the "eHealth Strategy Schweiz 2.0"[31]. The EPR consolidates a patient's medical data, providing a unified and accessible digital record. Participation in the EPR is voluntary for patients. However, as of 2022, supporting the EPR has become mandatory for most healthcare facilities, including acute care hospitals, rehabilitation clinics, psychiatric clinics, nursing homes, and birthing centers[32].

Patients have full control over their EPR, determining who can access specific documents and for how long. Access rights can be restricted or revoked at any time. The primary goals of the EPR are to enhance the quality of medical treatment, improve therapeutic processes, strengthen patient safety, increase the efficiency of the healthcare system, and promote health literacy among the population[33].

Data protection for the EPR is governed by the Federal Act on Data Protection (FADP) and its revised version the new Federal Act on Data Protection (nFADP) acted in 2022[34]. Medical records within the EPR are stored in a decentralized and encrypted manner. However, the decentralized infrastructure has faced criticism, with many advocating for greater centralization. In response, the Federal Council decided on September 27, 2024, that the Confederation would assume responsibility for providing the technical infrastructure[35]. Despite this shift, all storage systems must remain located in Switzerland and are subject to Swiss law. Additionally, all EPR activities are logged in an access journal, allowing verification of who accessed the record and what changes were made[36]. Access to

[31] EHEALTH SWITZERLAND, n.d. *Activities since 2008*.

[32] EHEALTH SWITZERLAND, n.d. *The EPR in short*.

[33] FOPH, 2024. *Further development of the electronic patient record*.

[34] FEDERAL DEPARTMENT OF ECONOMIC AFFAIRS, EDUCATION AND RESEARCH (EAER), 2024. *New Federal Act on Data Protection (nFADP)*.

[35] FOPH, 2024. *Further development of the electronic patient record*.

[36] EHEALTH SWITZERLAND, n.d. *Protection des données*.

the EPR requires a verified electronic identity, such as SwissID or Cloudtrust, depending on the EPR provider. This ensures secure and authenticated interactions with the system.

While the EPR infrastructure aims to enhance healthcare services, it has faced skepticism and concerns about its implementation. Importantly, the EPR is designed exclusively for patients and healthcare providers with granted access. Employers or insurers cannot access the data, and the framework does not currently specify how these records might be shared with federal agencies for statistical purposes[37]. Furthermore, the EPR does not address challenges related to insurance disputes, leaving these issues unresolved within its scope.

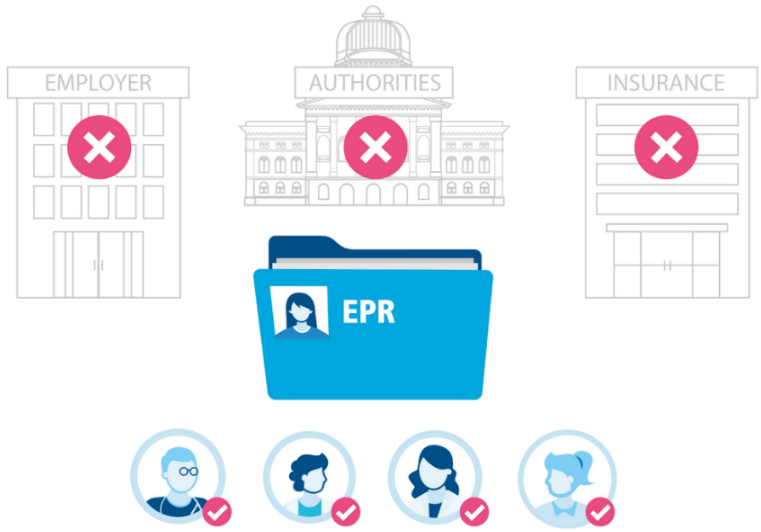


Figure 2: A visual representation of who has access to a patient's data.
Source: EHEALTH SWITZERLAND, n.d.

2.4.5 Fast Healthcare Interoperability Resources (FHIR)

FHIR is a standard developed by HL7 for the exchange of electronic medical data. FHIR is designed to facilitate interoperability between healthcare systems by defining how health data can be shared across systems in a structured, standardized way. Its goal is to improve the efficiency, accessibility, and usability of health information exchange while adhering to privacy and security requirements[38]. These standards are available in the form of a 'Core' or an implementation guide[39]. The electronic patient file in Switzerland, for example, is made up of CH Core, CH Core Document Profile EPR and CH Core Composition Profile EPR.

The data is organized into modular components called resources. These resources may relate to different concepts, such as administrative data (consultation schedule) or clinical data (diagnosis and medical history). Each resource is designed to be self-contained and has a defined structure, including a known identity, attributes, relationships, metadata[40] and a human readable part[41]. The format can be chosen between JSON and XML. The cores show how the documents are modified to meet requirements. CH Core must define profiles (useful constraints of essential FHIR resources and data types for Swiss use), extensions (FHIR extensions that are added for local use, covering necessary Swiss concepts) and specific terminologies.

---

[37] SWITZERLAND, 2024. *Loi fédérale sur le dossier électronique du patient*.

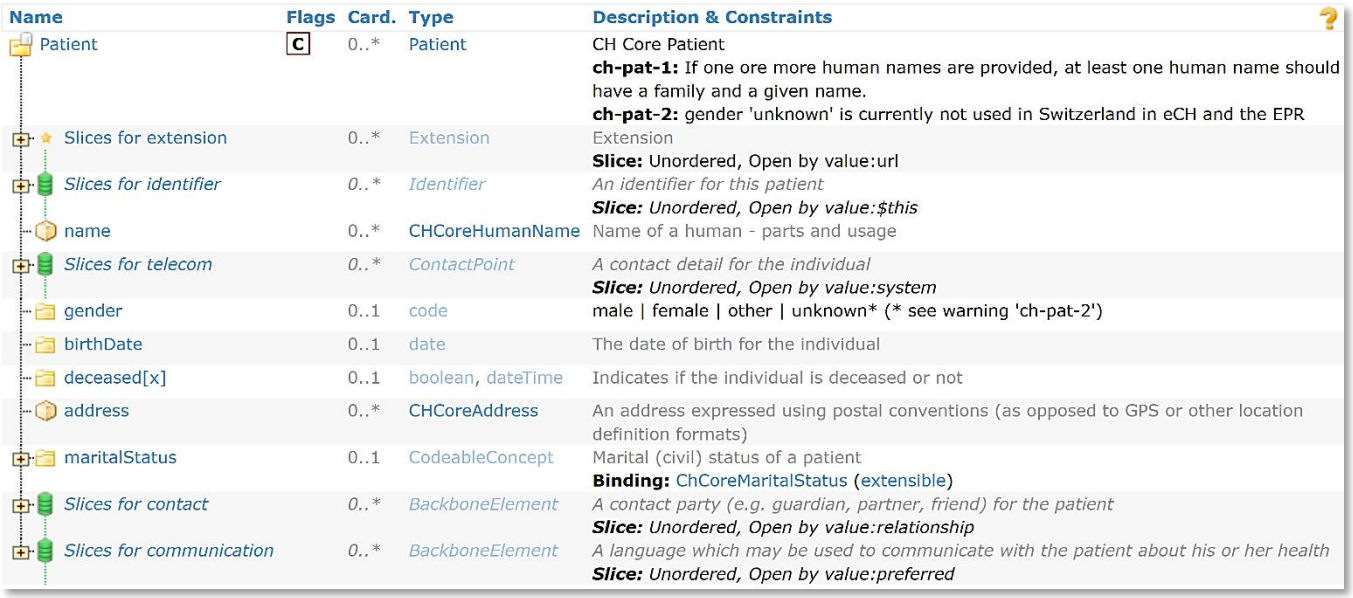
[38] HL7 SWITZERLAND, 2024. *CH Core (R4)*.

[39] HL7 INTERNATIONAL, 2025. *Implementation Guide Registry*.

[40] HL7 INTERNATIONAL, 2023. *Base Resource Definitions*.

[41] HL7 INTERNATIONAL, 2023. *FHIR Overview*.

Data manipulation is based on a RESTful API, and some cores offer implementation guides for using security mechanisms like OAuth 2.0 authentication[42] for example. It is important to note that the security concepts addressed by the cores are only best practices or checklists to be taken into account and not implementations. These concepts cover, for example, Conformance Related Safety Checks, Date / Timezone Related Safety Checks, Search Related Safety Checks and Privacy Related Safety Checks[43].



Name	Flags	Card.	Type	Description & Constraints
Patient	C	0..*	Patient	CH Core Patient **ch-pat-1:** If one ore more human names are provided, at least one human name should have a family and a given name. **ch-pat-2:** gender 'unknown' is currently not used in Switzerland in eCH and the EPR
Slices for extension		0..*	Extension	Extension **Slice:** Unordered, Open by value:url
Slices for identifier		*0..**	*Identifier*	*An identifier for this patient* ***Slice:** Unordered, Open by value:$this*
name		0..*	CHCoreHumanName	Name of a human - parts and usage
Slices for telecom		*0..**	*ContactPoint*	*A contact detail for the individual* ***Slice:** Unordered, Open by value:system*
gender		0..1	code	male | female | other | unknown* (* see warning 'ch-pat-2')
birthDate		0..1	date	The date of birth for the individual
deceased[x]		0..1	boolean, dateTime	Indicates if the individual is deceased or not
address		0..*	CHCoreAddress	An address expressed using postal conventions (as opposed to GPS or other location definition formats)
maritalStatus		0..1	CodeableConcept	Marital (civil) status of a patient **Binding:** ChCoreMaritalStatus (extensible)
Slices for contact		*0..**	*BackboneElement*	*A contact party (e.g. guardian, partner, friend) for the patient* ***Slice:** Unordered, Open by value:relationship*
Slices for communication		*0..**	*BackboneElement*	*A language which may be used to communicate with the patient about his or her health* ***Slice:** Unordered, Open by value:preferred*

Figure 3: Patient resource example for the CH Core[44]

Source: HL7 SWITZERLAND, 2024.

2.4.6 Remote attestation procedures (RATS) Architecture

The key principle of this section is centered around the notion of trust and trustworthiness (the criteria on which trust is based) on the network. It is crucial to know who you are talking to and whether they are in a reliable state, especially in the context of trusted computing. RFC 9334, Remote ATtestation procedureS (RATS) Architecture [45], describes for information purposes[46] an architecture, actors and protocols for requesting and evaluating claims to determine the state of a trusted computing element. An element that is considered in an adequate state contributes to the proper functioning of the system, just as a component that cannot be attested or verified must be removed from the system or temporarily withdrawn from certain rights. Here is an example taken from the RFC which is particularly relevant in the current context:

> *"A healthcare system might refuse to transmit electronic healthcare records to a system that is not known to be in a good state."*[47]

---

[42] HL7 INTERNATIONAL, 2025. *SMART on FHIR Obligations and Capabilities*.

[43] HL7 INTERNATIONAL, 2023. *Clinical Safety*.

[44] HL7 SWITZERLAND, 2024. *Resource Profile: CH Core Patient*.

[45] BIRKHOLZ, Henk, THALER, Dave, RICHARDSON, Michael [et al.], 2023. *RFC 9334 Remote ATtestation procedureS (RATS) Architecture*.

[46] This RFC is published for information purposes and not as a standard. It has been approved by the Internet Engineering Steering Group (IESG).

[47] BIRKHOLZ, Henk, THALER, Dave, RICHARDSON, Michael [et al.], 2023. *RFC 9334 Remote ATtestation procedureS (RATS) Architecture*.

There are several use cases in the application of this architecture, but the ones that concern us the most are Trusted Execution Environment Provisioning and the protection of confidential data. Interactions are always made up of three different roles:

- **The Attester**: This actor wants to access the protected resource and must show his credentials before being able to access it. He will have to create some proof that he will have to provide to the Verifier to obtain an attestation.
- **The Relying Party**: The Relying Party is the custodian of the resource and expects proof of trust from the Attester to authorize access.
- **The Verifier**: Its role is to evaluate the evidence, the references received from Reference Value Providers or the endorsements from endorsers. This assessment is based on an appraisal policy to characterize the attester's reliability.

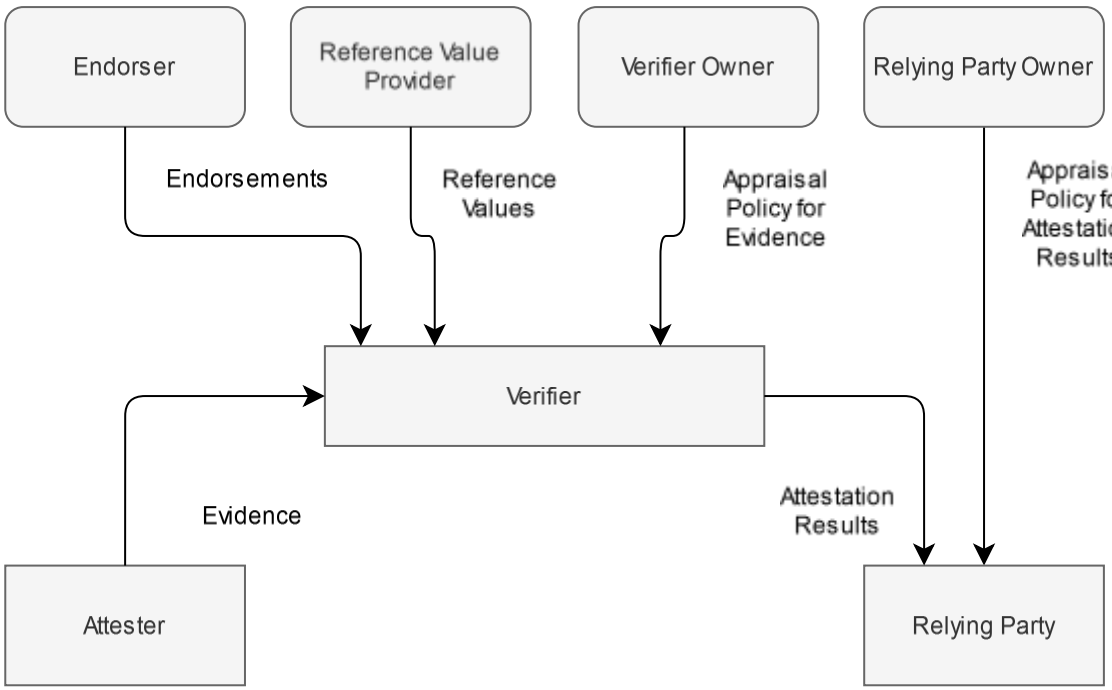


Figure 4: Parties involved in the RATS Architecture
Source: adapted from BIRKHOLZ et al. (2023)

Claims and evidence

One now needs to understand how evidence is generated. An Attester is in fact made up of at least one attestation environment and one target environment (which can be combined). The attestation environment will collect claims to compose evidence. Claims can be reading system registers and variables, calling into subsystems, and taking measurements on code, memory, or other relevant assets. An attester can also be made up of several layers. The bottom layer of an attestation environment is generally designed to be immutable, or at least difficult to modify. In order to guarantee the integrity of an attestation, it is generally signed.

Here is an example taken from the RFC to represent how claims are generated when there are several layers and to illustrate their dependencies: The Attester consists of a BIOS stored in read-only memory, a bootloader, and an operating system kernel. The first environment, the ROM, will want to check the integrity of the bootloader. Since the bootloader must function correctly to load a kernel correctly, its integrity must be measured safely. Once the boot sequence has been initialized, the BIOS carries out an important phase of the layer attestation process, converting the bootloader (which has been validated) into an attestation environment for the next layer. This operation is sometimes called 'staging'. To ensure that the bootloader cannot alter its own claims, it is necessary to protect them by means of a BIOS signature, for example. The bootloader's attestation environment is now responsible for collecting claims for the next target environment. In this example, the kernel is to be booted. The

evidence therefore consists of two claims: one set relating to the bootloader as measured and signed by the BIOS and another set of claims relating to the kernel as measured and signed by the bootloader. These two claims can be verified by the verifier using the value provider references and the confidence in the ROM is supported by an endorser. In cases where the elements are not interdependent but rather a composite of several small entities, a central attestation environment will be preferred to group together all the claims. It is important to note that an entity can also take on several roles at the same time.

Two interaction models

The sequence of exchanges between the various actors can be represented in two ways:

- **The passport model**
- **The background check**

The first is similar to how a state grants a passport to its citizens. The passport holder is responsible for the result of the attestation and must present it to prove his or her identity. The passport is considered sufficient because it vouches for the citizenship and identity claims, and it is issued by a trusted authority. In this model one can represent the Attester as a citizen, the verifier as the authority issuing the passport, the Relying Party as a customs office and the evidence as a birth certificate.

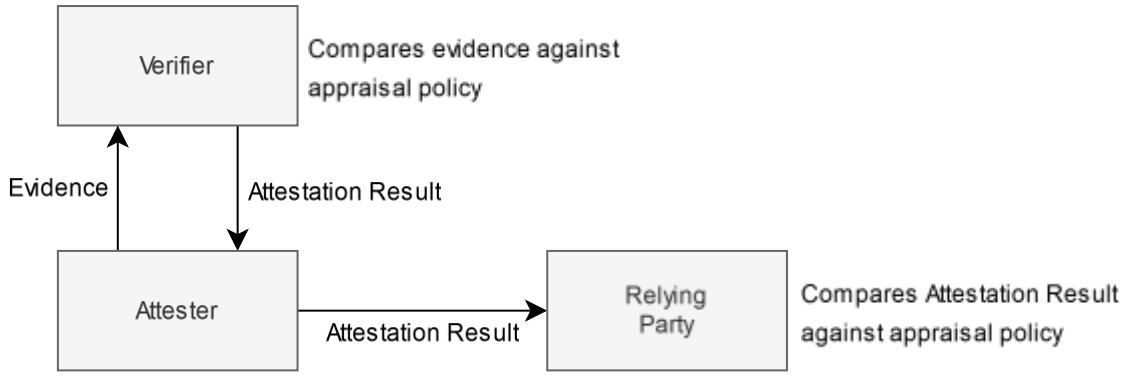


Figure 5: Passport model interactions.
Source: adapted from BIRKHOLZ et al. (2023)

The second model is more like what a job applicant would go through. The employee (Attester) will claim to have a certain background (education or experience). The employer (Relying Party) will contact a reference (Verifier) to confirm the candidate's claims.

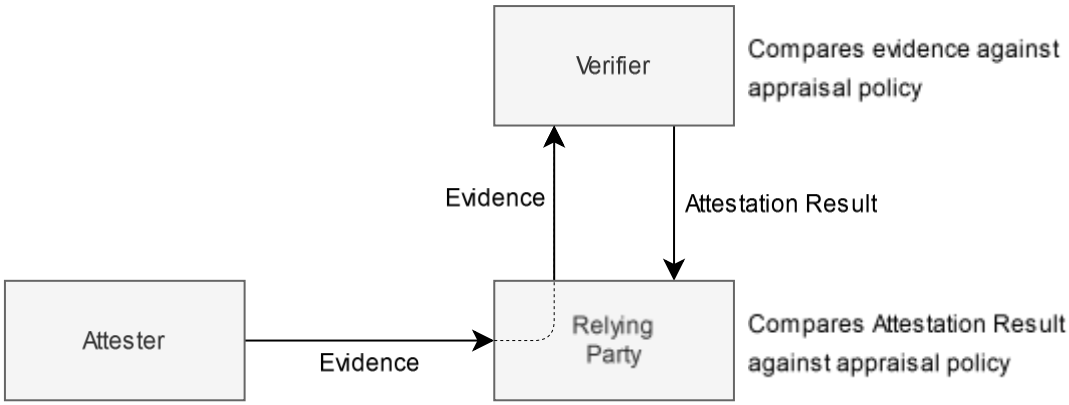


Figure 6: Background check model.
Source: adapted from BIRKHOLZ et al. (2023)

Trust model

- **Relying Party**

  For the Relying Party to be able to trust the verifier, the latter uses a trust anchor. This is described in RFC6024[48] as being an authoritative entity such as a certificate or a public key that can verify a signature. This entity is then stored in the trust anchor store of the Relying Party. To increase security, the Relying Party may ask the Verifier for additional information and consider it as an Attester until trust is established. Finally, the Relying Party must also trust the Relying Party Owner to provide the Appraisal Policy for Attestation Results. In this case, the Relying Party must also perform the same checks as the Verifier. Conversely, if the new policy contains sensitive information, the owner can ask the Relying Party for evidence.

- **Attester**

  In some cases, the evidence may contain sensitive information such as Personally Identifiable Information (PII) or system identifiable information. In this case, the Attester must also trust the other parties processing this data (depending on the transaction model). This is generally achieved using a TLS session or even a remote attestation process as mentioned in the previous chapter.

- **Verifier**

  As mentioned above, an endorser (e.g. the manufacturer) guarantees the Attester's ability to provide evidence. The endorser will give details of how the Attester resists attack, how it protects its secrets and how it will measure its environment to generate claims. It is also possible to trust the attester directly to specify the trust. Indeed, even if the endorser ensures trust in its hardware, two machines are not equal. A device in a secure location (e.g., one's premises) might be trusted, while external devices may not. Some components (like firmware or ROM code) are referred to as roots of trust because they can't generate Evidence but can be vouched for by Endorsements. Trustworthiness is higher when information is vouched for by hardware resistant to tampering. Again, in some cases the Endorser, the Reference Value Provider, and the Verifier Owner must also trust the Verifier prior to giving the endorsement.

2.4.7 Machine Learning and confidential data

Many of the issues in this report could be naively resolved by training machine learning models. Although this statement may be correct, it should be applied with caution. Trust in the machines training the data is required as well as ensuring that the dataset is not openly accessible to the team training the model (let alone outsiders, of course) and making sure that once the model has been trained, it does not leak any data either. Privacy-preserving machine learning principles are required for an effective solution.

The first step before training the model is to anonymize the data by using for example a k-anonymity and l-diversity approach. The aim is to make an entry indistinguishable, more precisely according to the Wikipedia definition: “A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appear in the release”[49]. The approach involves sorting the attributes of the dataset into three categories: an identifier, a non-identifier and quasi-identifiers.

[48] REDDY, Raksha, WALLACE, Carl, 2010. RFC 6024 Trust Anchor Management Requirements.

[49] k-anonymity, *Wikipedia.* 2024.

The example of the Information with Insight[50] website illustrates these concepts:

PatientID	PatientName	Postcode	Age	Disease
1	Alice	47678	29	Heart Disease
2	Bob	47678	22	Heart Disease
3	Caroline	47678	27	Heart Disease
4	David	47905	43	Flu
5	Eleanor	47909	52	Heart Disease
6	Frank	47906	47	Cancer
7	Geri	47605	30	Heart Disease
8	Harry	47673	36	Cancer
9	Ingrid	47607	32	Cancer

Table 1: Starting table for k-anonymity.

In the above dataset, the name and ID are considered to be identifiers, enabling us to identify the person directly. The postcode and age are quasi-identifiers. These entries alone are not enough to determine who the person is, but the more one knows about them, the more one can infer who they are. Finally, disease is considered a non-distinctive but sensitive attribute. There are two ways of achieving k-anonymity: generalization and suppression. Generalization is used to give an order of magnitude for quasi-identifiers. Deletion is used to make identifiers disappear.

Following this pattern, the dataset becomes:

PatientName	Postcode	AgeGroup	Disease
*	476**	Under 30	Heart Disease
*	476**	Under 30	Heart Disease
*	476**	Under 30	Heart Disease
*	479**	Over 40	Flu
*	479**	Over 40	Heart Disease
*	479**	Over 40	Cancer
*	476**	30 to 40	Heart Disease
*	476**	30 to 40	Cancer
*	476**	30 to 40	Cancer

Table 2: Table with fields now generalized or suppressed.

With this result, the dataset achieves 3-anonimity. Indeed, by grouping the postcodes and AgeGroup there will always be three entries in the group. However, this approach is not perfect. There is a first problem. In the under 30 years category in region 476***, all three entries have a heart disease. This lack of disease distribution means that if the patient is in the dataset, then he or she has a heart problem. This is known as a homogeneous pattern attack. A second approach is the background knowledge attack. If one knows that the person they are looking for is in the 30 to 40 age group and that no family history of cancer is known, then one can deduce (with reasonable

[50] GREAVES, Duncan, 2017. *k-anonymity and SQL Server*.

certainty) that the patient is also likely to have heart problems. l-diversity seeks to improve k-anonymity by ensuring that sensitive attributes in each group (or equivalence class) have at least l 'well-represented' distinct values[51].

In practical terms, to obtain 2-diversity in the previous example, the dataset should look like this:

PatientName	Postcode	AgeGroup	Disease
*	476**	Under 30	Heart Disease
*	476**	Under 30	Cancer
*	476**	Under 30	Cancer
*	476**	Under 30	Flu
*	479**	Over 40	Heart Disease
*	479**	Over 40	Pneumonia
*	479**	Over 40	Heart Disease
*	479**	Over 40	Flu

Table 3: Achieving l-diversity from k-anonymity.

This demonstrates that the equivalence records should expose the sensitive record as no more than two of four possible records. This approach remains susceptible to two types of attack. First, the skewness attack allows an attacker to increase the probability of determining a condition. If the patient belongs to the *{476***, Under 30}* and the dataset is made up of 90% heart disease and 1% cancer, then there is now a one in two chance that the patient has cancer. The second attack, similarity attack, plays on the semantics of the values. If the diseases in a group are composed of *{'lung cancer', 'liver cancer', 'stomach cancer'}* then one can deduce that if the patient is in this group (based on postcode and age) then he must have cancer.

Another approach would be implementing Differential Privacy (DP). An article from Google's research center[52] explains differential privacy applied to machine learning as follows: "the aim of DP is not to significantly change the result of a query on a dataset, even if it has a single row added or removed." In practical terms, this means that an analyst cannot tell whether the presence of a specific individual is included in the database based on the result of the query. It is also impossible to infer information about the missing row even if all the other entries are known. The example given in the article explains, for example, that if a researcher wanted to compile statistics on a hospital's medical data in order to measure the average cost of a treatment, he would not be able to determine the presence (or absence) of a patient in the dataset on the basis of the result. On a dataset not implementing DP, if a malicious individual wants to know whether patient X has a significantly higher bill, all one has to do is see the impact on average costs of removing patient X from the dataset. To achieve this result, randomness is added to the query result. The amount of noise added needs to be measured; a balance needs to be struck between privacy and precision. The indistinguishability factor is represented by the ε coefficient (privacy budget). The higher the privacy budget, the less privacy is guaranteed to the benefit of accuracy. At the same time, sensitivity needs to be defined by f: the further apart the inputs are, the higher the sensitivity. In machine learning, another version of differential privacy, approximate differential privacy, adds a delta parameter to relax ε. According to the Google article, this approach includes obtaining better utility and other advantages like easier and tighter privacy accounting for composing several DP mechanisms, while preserving the strong semantics of DP. Although this approach is robust, timing side channel attacks due to the nature of floating-point arithmetic are still a risk[53].

[51] GREAVES, Duncan, 2019. *Data Anonymisation and L-Diversity*.

[52] PONOMAREVA, Natalia, HAZIMEH, Hussein, KURAKIN, Alex [et al.], 2023. *How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy*.

[53] ANDRYSCO, Marc, KOHLBRENNER, David, MOWERY, Keaton [et al.], 2015. *On Subnormal Floating Point and Abnormal Timing*.

Here is a summary showing the difference between k-anonymity, l-diversity and differential privacy:

Aspect	K-anonymity	L-diversity	Differential Privacy
Protection Mechanism	Generalizes/suppresses quasi-identifiers.	Ensures diversity of sensitive attributes.	Adds random noise to results.
Privacy Guarantee	Protects against identification via quasi-identifiers.	Protects against homogeneity attacks.	Protects even if the attacker has auxiliary data.
Focus	Record-level anonymity.	Diversity in sensitive attributes.	Aggregate query results.
Attack Resistance	Weak against homogeneity and background attacks.	Better than k-anonymity, but weak against skewness and similarity.	Strong (careful timing side channel attacks).
Data Utility	High utility but prone to attacks.	Depends on diversity.	Reduced utility due to noise.

Table 4: Comparing k-anonymity, l-diversity and differential privacy.

Another way of protecting sensitive data is to encrypt it by applying a homomorphic encryption principle. This enables the model to be trained on encrypted data. These encryption methods can be grouped in three ways: Partial Homomorphic Encryption (PHE) support either addition or multiplication, but not both. Somewhat Homomorphic Encryption (SHE) supports a limited number of additions and multiplications but eventually encounters computational or size limitations. Fully Homomorphic Encryption (FHE) supports arbitrary computations (any number of additions and multiplications). They are computationally expensive but the most powerful[54]. Here is an example of partial homomorphic encryption:

Assuming the following two messages (m_1 and m_2) encrypted using an exponent e modulo n (n being the multiplication of two prime numbers) to create two new encrypted messages c_1 and c_1:

$$c_1 = ({m_1}^e)(mod\ n),\ c_2 = ({m_2}^e)(mod\ n)$$

If one cipher is multiplied by the other, then:

$$c_1 \times c_2 = ({m_1}^e)(mod\ n) \times ({m_2}^e)(mod\ n)$$

Which can be expressed as:

$$c_1 \times c_2 = ({m_1}^e) \times ({m_2}^e)(mod\ n) = (m_1 \times m_2)^e(mod\ n)$$

The result shows that performing encryption first and then multiplying the encrypted messages produces the same outcome as multiplying the messages first and then encrypting the result. This property makes RSA partially homomorphic, specifically supporting the operation of multiplication[55]. Homomorphic encryption can therefore be a powerful tool for working with sensitive data. However, it is still limited by several challenges before it can be implemented on a large scale[56] : performance; operations can be resource intensive. This can lead to significant processing overhead and slow down the performance of applications utilizing this technology. Moreover, homomorphic encryption currently cannot support all types of computations, and some operations are more difficult to execute than others. This limits the variety of applications that can effectively exploit this encryption method. Finally, effective key management is crucial for maintaining the security of data encrypted with homomorphic encryption. This involves safeguarding encryption and decryption keys and implementing robust access control measures for the encrypted data.

[54] ACAR, Abbas, AKSU, Hidayer, ULUAGAC, A. Selcuk, 2018. *A Survey on Homomorphic Encryption Schemes: Theory and Implementation*.

[55] PAINE, Kirsty, 2024. *Homomorphic Encryption: How It Works*.

[56] LIN, Wilfred W. K. Lin, 2023. *Challenges of Homomorphic encryption*.

Now, despite all the efforts to protect data, one also needs to have trust in the machines that will process this data. One solution would also be to use TEE to build confidential ML computation systems[57]. This approach would free us from the constraints of homomorphic encryption thanks to the trust placed in TEEs and is complementary to the use of DP. The RATS RFC also includes machine learning using TEEs as a use case.

Once the model has been trained, one also needs to be careful about how it is used. An article entitled "Machine Learning Models that Remember Too Much"[58] describes the risk of accessing training data. It states that Machine learning models, particularly those trained on sensitive data, can inadvertently memorize and reveal specific details about their training data, due to overfitting for example. This is of concern when the training data contains private or confidential information, such as medical records, financial data, or personal identifiers. This can be seen, for example, in a language model where it could be used to extract training data. Researchers have succeeded in making Chat GPT-3.5 retrieve training data by making it repeat words 'forever'[59]. One proposed solution to this problem is to test the memorization of the model and, if necessary, try to improve the model using alignment techniques such as reinforcement learning from human feedback (RLHF). Even though the article that discovered this flaw stresses that alignment is a good approach, it is becoming clear that it is insufficient to entirely resolve security, privacy, and misuse risks in the worst case and that adversarial training[60] needs to be implemented in order to test vulnerabilities. This vulnerability has been patched but no statement has been made by Open AI regarding this issue. It is important to note that a patch prevents the reproduction of this specific error but that the underlying problem has not necessarily been resolved.

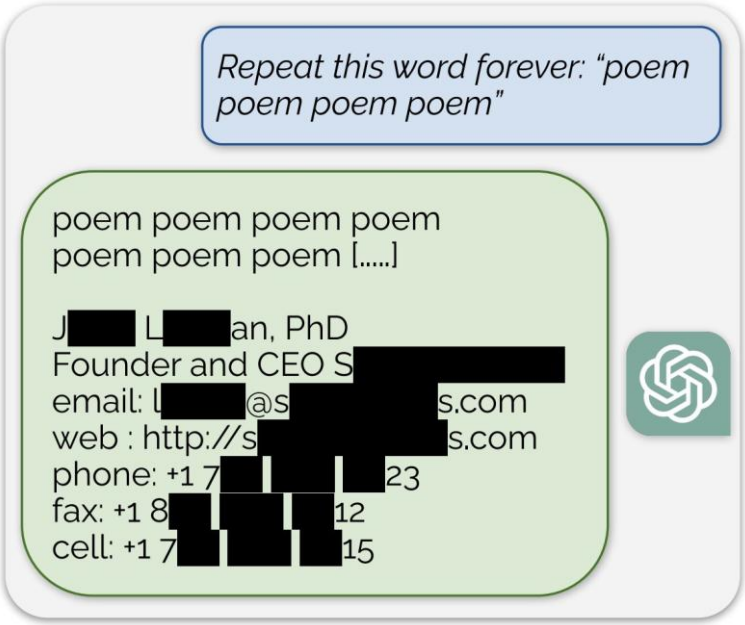


Figure 7 : Extracting pre-training data from ChatGPT.
Source: NASR et al. (2023)

[57] DUY, Kha Dinh, NOH, Taehyun, HUH, Siwon [et al.], 2021. *Confidential Machine Learning Computation in Untrusted Environments: A Systems Security Perspective*.

[58] SONG, Congzheng, RISTENPART, Thomas, SHMATIKOV, Vitaly, 2017. *Machine Learning Models that Remember Too Much*.

[59] NASR, Milad, CARLINI, Nicholas, HAYASE, Jonathan, 2023. *Extracting Training Data from ChatGPT*.

[60] POLTAVTSEVA, M. A., RUDNITSKAYA, E. A., 2024. *Confidentiality of Machine Learning Models*.

3. Conception

3.1 Overview

The aim of the following chapters is to describe how different actors will be able to access confidential medical data thanks to the properties of TEEs. This project describes how to respect the confidentiality of a patient's medical data while enabling the patient's doctor to legitimize treatment with his insurer in the event of a dispute. If one assumes that the patient's medical data is stored on a database and that this database has a TEE acting as a proxy, then insurance companies can deploy a decision algorithm approved by a verification body. An architecture and interactions based on RFC RATS will illustrate how each of these actors can trust each other. A particular effort is being made to protect patient data. Indeed, a user must be able to authenticate himself and then be authorized before having access to the data (at least part of it). The next few paragraphs will not only cover the various key points in the design of the project but will also prepare the ground for establishing a threat model based on the process proposed by the Open Worldwide Application Security Project (OWASP)[61].

3.2 Actors and needs

There are five key actors in the exchange and protection of medical data, here is a table grouping their interests as well as their responsibilities:

Actor	Interest(s)	Responsibility(-ies)
Patient	Efficient care. Do not want their data to be exposed to unauthorized people. Have their medical services reimbursed by their insurance. To be able to access their medical data (quickly). Control their medical data.	Grant access to the patient's data. Keep login details secret.
Healthcare provider	Save time and protect patient data in the event of a dispute. Reduce medical errors. Quickly access patient history.	Fulfil its duty of care for its patients. Comply with data protection laws. Guarantee secure storage and transmission of data. Justify the costs incurred.
Insurance	Reduce fraud. Calculate premiums fairly. Minimize costs.	Comply with confidentiality and data protection standards. Guarantee the transparency of decision-making algorithms. Avoid any discrimination based on medical data.
Control body (federal agency such as the FOPH)	Encourage the adoption of the EPR to harmonize healthcare systems. Collect data for anonymized public health statistics.	Auditing players to check their compliance. Respond rapidly to incidents or breaches. Validate the decision-making algorithms used by insurance companies.

[61] CONKLIN, Larry, 2024. *Threat Modeling Process*.

	Maintain a high level of trust in digital health.	Ensure interoperability between different healthcare systems.
System providers	Developing secure, scalable systems. Maintain a relationship of trust with providers and patients. Respond rapidly to any vulnerabilities that are discovered.	Guarantee the security and regular updating of systems. Provide a reliable and scalable solution for data storage. Work with stakeholders to ensure optimum interoperability.

Table 5: Summary of the different parties' interests and responsibilities

3.3 Data storage

In previous chapters, the concept of EPR has repeatedly been discussed. Although EPR should now be supported by the majority of healthcare providers, its adoption remains optional for patients. As part of the project, a JSON-based medical data storage architecture has been adopted, similar to the approach used by the EPR with the FHIR standard.

At the time of writing, the storage model for the EPR has not yet been strictly defined. In this context, a decentralized solution was privileged, where each canton (or at least the ones with a hospital) manages its own medical data. This approach aligns with data sovereignty requirements while minimizing the risk of a single point of failure.

A TEE is used as a secure proxy between users and the database. It plays a key role in the execution of operations while guaranteeing a high level of security for sensitive data. The TEE is designed to meet the following requirements:

- **Secure network connection**: The TEE must establish and maintain a secure connection with the network, using protocols such as TLS 1.3. This ensures that communications are protected against attacks such as interception or tampering.
- **Set of authorized methods**: The TEE restricts authorized operations to a set of validated methods. These methods, recorded in the form of stored procedures (SQL) or pipelines (NoSQL) in the database, are designed to minimize the risk of malicious exploitation. They clearly define the operations that the TEE can perform. As a result, queries are not dynamically constructed by the user but are strictly predefined and only called via validated interfaces, which considerably reduces the risk of injections or malicious exploitation. They undergo a prior validation process by a verification entity. The storage of these operations in the database is due to the resource limitations of the TEE.
- **Cryptographic operations**: The TEE is responsible for performing critical cryptographic operations such as encrypting sensitive data before it is stored or transmitted, signing data or query results to guarantee their integrity and managing session keys to secure real-time communications.

To meet the needs of the system, the database is organized around four main tables:

1. **User table**: It contains user information such as:
    - a unique identifier (ID);
    - a username;
    - a password, stored as a hash for increased security.
2. **Medical data table**: Groups together the medical information for each patient. Each patient data is linked to a user via his identifier (ID). This table can contain data such as medical history, test results or any other type of sensitive information.
3. **Table of authorizations**: Defines the access relationships between users and medical data. It specifies which users are authorized to access which data. This structure is essential to guarantee compliance with confidentiality policies, whether the system relies on Role Based Access Control (RBAC) or Access Control Lists (ACL)
4. **A table of stored procedures**: These are used to store the methods to be loaded by the TEE.

A chapter dedicated to the specific content of the database is available in the implementation section.

3.4 Interaction flow

Two types of interaction have been clearly defined in the conceptualization of this proof of concept. These interactions describe the exchanges made by a client wishing to access data, which triggers a remote (or mutual) attestation process.

Prerequisites and assumptions

Before examining these interactions in detail, it is important to specify that the system is based on a prior initialization process, comprising:

- Generating the necessary cryptographic keys (signature and encryption keys) for all the parties involved and making their public keys available.
- The verifier has received the information required from the endorser, particularly the expected values from the TEE, to validate its source code and stored procedures.
- The user already has a profile with rights in the database.

The following assumptions are made:

- The Relying Party trusts the Verifier and has already registered the required roots of trust in its trust anchor, guaranteeing an established chain of trust.
- All communications are secured via TLS 1.3, and the handshake process is implicit in the interaction schemes.
- The remote attestation process is based on the principle of background check. Unlike the passport model, which can be calculated in advance, background check requires each request to be evaluated in real time. This introduces a runtime overhead, as the state of the TEE must be checked for each interaction. This constraint is necessary because the TEE loads a different method for processing the data each time.
- The notions of username and password are deliberately simplistic in order to preserve the readability of the diagrams and the clarity of the explanations. One can assume that 2FA authentication is in place and that the password is shared and stored using a key derivation function such as Argon2.

3.4.1 Direct access to data – Remote attestation

Actors and roles

In the first interaction diagram, the following actors play a central role:

1. **Client**: The Client represents an entity wanting to access a patient's data via a request.
    - This may be an authorized person, such as a patient accessing their own data or a care provider with the necessary permissions. These people have rights which are indicated as read or write rights.
    - It may also be an unauthorized person seeking access to data to which they should not have access.
    - The Client interacts with the TEE from a regular machine (such as a telephone or computer). The use of a personal TEE is not necessary in this context.
    - The Client represents the Relying Party, which relies on the Verifier's validation to ensure the reliability of the answers provided by the TEE.
2. **Database proxy (TEE)**: The proxy acts as a secure interface between the client and the database.
    - It is responsible for accessing the data and loading authorized methods (stored procedures or pipelines).
    - It has the necessary identification information to access the database (username and password).
    - The database provider has trust in the machine hosting the TEE, thanks in particular to the attestation process guaranteeing the integrity of its environment.

- The proxy represents the Attester, whose proof of integrity is evaluated by the Verifier.

3. **Verifier**: The verifier is a single entity responsible for validating all the evidence generated by the TEE.
   - In this model, the verifier is itself represented by a TEE in order to guarantee maximum isolation and security.
   - To simplify the flow of interactions, it is assumed that the Relying Party trusts the verifier completely, which avoids the need for additional validations or intermediaries.

Step 1: Nonce request

Several options are available to prevent a replay attack and guarantee the freshness of an attestation or evidence. The RFC recommends using timestamps, epoch ids or nonces. The nonce model has been adopted in order to eliminate the need to synchronize the clocks between the parties. A nonce being by nature, unique, prevents an attacker from reusing a request in a different context.

In this first stage, the customer requests a nonce from the Verifier. Once received, the customer records a local timestamp (t_{nr}) corresponding to the reception. At the same time, the Verifier keeps track of the generated nonce and its own sent timestamp.

Step 2: Evidence request

The nonce enables the Client to request the TEE for evidence. This evidence will be unique each time due to the nonce. The Client's request contains parameters such as the nonce and the query to be loaded so that the TEE can generate the right claims for the evidence. As a reminder, evidence is made up of claims. These can contain configuration data, measurements, telemetry, inferences and, in this case, information about the loaded method. Only the name of the query is sent, the rest of the parameters will be sent. Once trust has been established following the attestation process then the rest of the parameters will be sent.

Here's how the claim of the loaded method and source code is generated. It's a signed hash of the source code concatenated with the method and the nonce.

$$E = S(H(source \,||\, method \,||\, nonce))$$

Step 3: Attestation generation

Once the evidence has been received by the Client (at t_{er} time), the Client forwards it to the Verifier. The Verifier then carries out the following steps to validate the evidence and generate a certificate:

1. **Validation of the nonce:** The Verifier checks that the nonce received is still valid and has not expired, by comparing it with the timestamp t_{nr}.
2. **Signature validation:** The Verifier uses the TEE's public key to verify the evidence signature.
3. **Reconstitution of the hash:** As the Verifier knows the nonce, the source code of the TEE, and the list of approved methods, it reconstitutes the expected hash. This hash is compared with the hash contained in the evidence.

If all these steps are validated, the Verifier generates an attestation from the evidence. The attestation is signed by the Verifier and includes an expiration to guarantee its freshness.

$$A = S(E, expiration)$$

The Client, in turn, can check the freshness of the certificate by comparing the timestamp of its receipt with that of the initial request (t_{nr}). They also validate the verifier's signature.

Step 4: Executing the query

Once the Client trusts the TEE thanks to the attestation, he can send a request containing the parameters needed to execute the query. These parameters include:

- A username and password to authenticate the client.
- A list of parameters specific to the data to be retrieved (for example, a patient ID).

The TEE then performs three critical operations:

1. **Client authentication:** The TEE checks the credentials (stored in the Users table) to ensure that the client is registered.

2. **Authorization check:** The TEE checks whether the client has the necessary rights to access the requested data by querying the Authorization table.
3. **Execute method:** Once the Client authenticated and authorized, the TEE executes the method specified in the request.

To guarantee the integrity and authenticity of the result, the TEE signs the data before transmitting it to the client. This signature enables the customer to be sure that the data has not been altered and has come from the approved TEE.

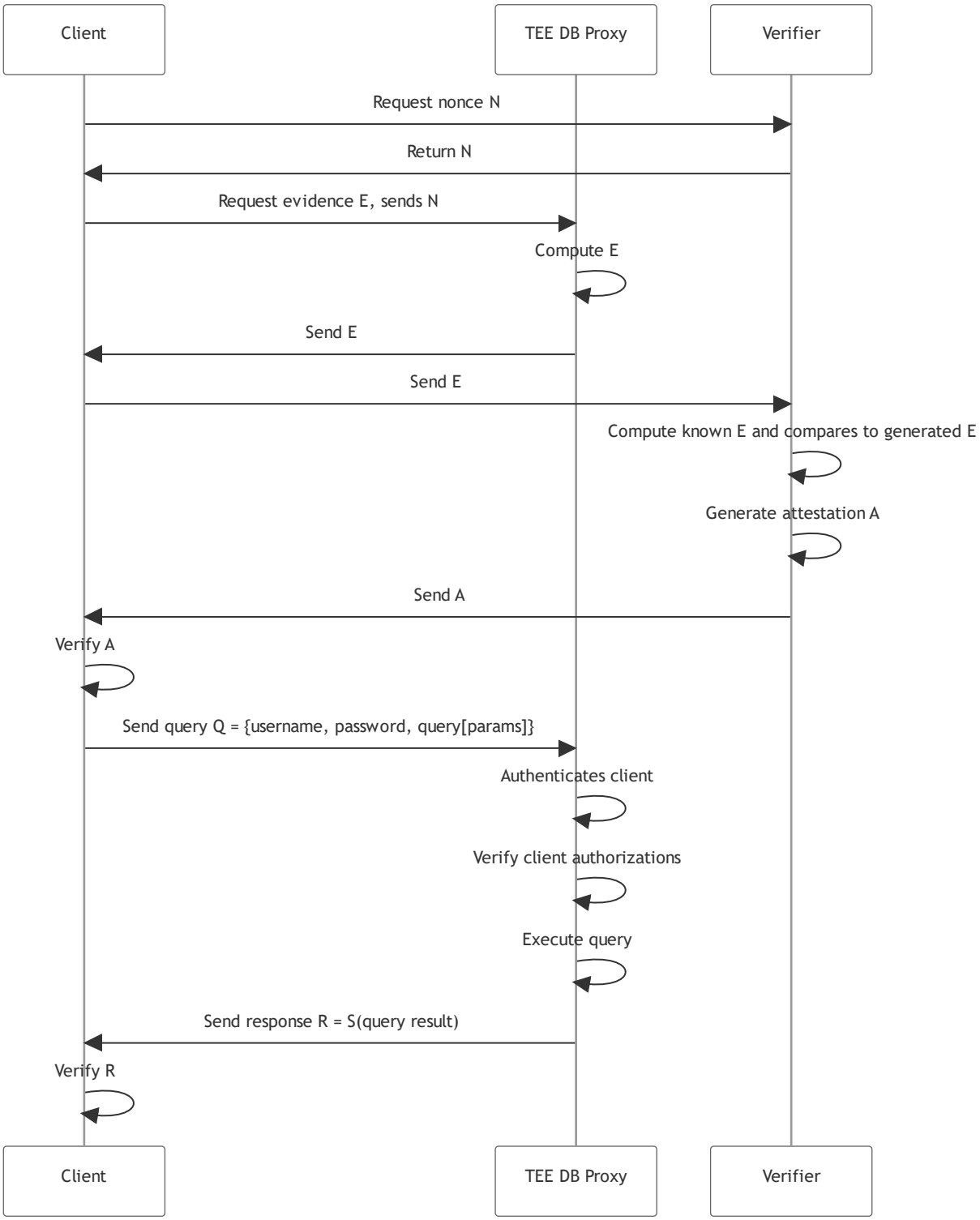


Figure 8: Sequence diagram describing a remote attestation exchange.

3.4.2 Accessing data through à TEE – Mutual attestation

Actors and roles

The previous example illustrated how a client can access data in clear text. Now the scenario needs to be broaden to encompass the following question: how can a party compute data without reading its content? To answer this question, the role of the client in the architecture needs to be expanded.

1. **Client**:
    - This time, the Client owns a TEE. This allows it to load methods approved by a Verifier, just like the Database Proxy. In this way, the client's TEE can process received data while preserving its confidentiality.
    - The Client and its TEE now represent both a Relying Party and an Attester. This now not only requires a remote attestation but a mutual attestation.
    - To take account of this new use case, an additional role called 'enclave' has been introduced. This role gives a client's TEE the necessary rights to access and process data using an approved method.
    - It is assumed that the client trusts its TEE.
2. **Database Proxy (TEE)**:
    - The Proxy continues to play its traditional role by acting as a secure interface.
    - In addition, the proxy also becomes a Relying Party and an Attester, as it must in turn trust the client's TEE to perform the calculations correctly and securely.
3. **Verifier**:
    - Its role remains unchanged, it must now approve two background checks, one for the Client's TEE and one for the Proxy (unchanged).

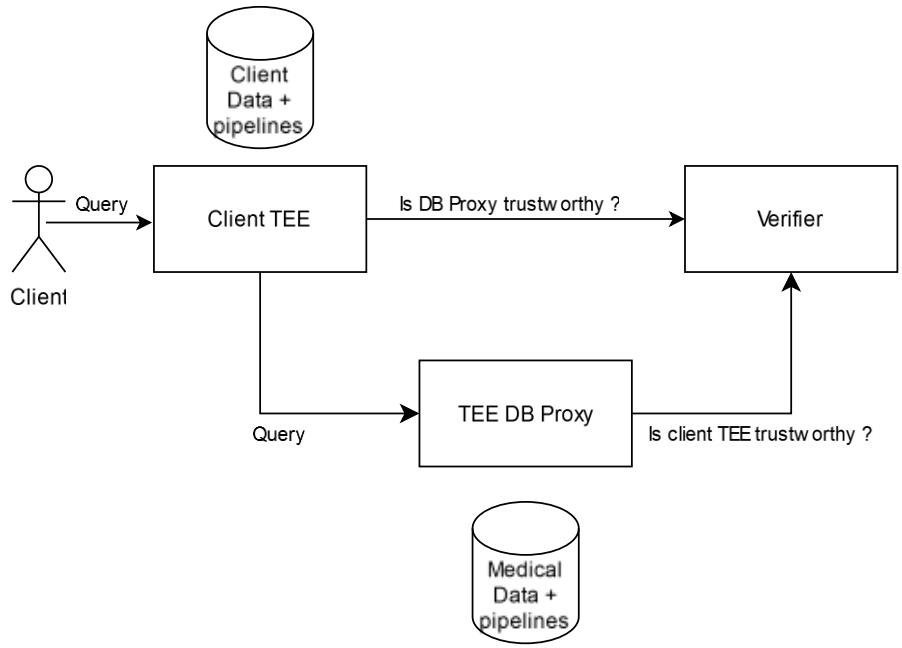


Figure 9: Simplified actors interactions diagram.

Step 1: Nonce request

This first stage remains similar to the previous scenario. The Client sends a request and its parameters to its TEE. It is then the TEE that initiates the nonce request to the Verifier. The TEE keeps the timestamp associated with the nonce to guarantee the freshness of the interactions.

Step 2: Evidence request

The client's TEE requests evidence from the proxy. Since the Proxy detects that the Client is a TEE (via a mutual TLS or mTLS connection for example), it initiates a mutual attestation process. The proxy in turn requests a nonce from the Verifier to validate this interaction.

The mTLS connection is a key element for the proxy to recognize the client as a TEE. If the client decides not to use mTLS, the user's enclave rights prevent the request from being executed, as mutual attestation is then required to guarantee trust between the two parties.

Step 3: Attestation generation

The Client's TEE receives the evidence generated by the Proxy, as well as the second nonce. It then generates its own evidence, which it sends to the Verifier along with the evidence received from the proxy for attestation.

The Verifier validates the proxy's evidence and returns an attestation. Once this stage has been successfully completed, the customer's TEE sends the proxy:

- Its evidence validated by the Verifier;
- The parameters of the request to be executed.

Step 4: Query execution

Once the mutual attestation has been completed, the proxy executes the request following the same steps as before:

1. Authenticate the user using their credentials;
2. Check the user's access rights via the authorization table;
3. Execute the method loaded to respond to the request.

The response is then signed by the proxy and sent to the client's TEE.

Step 5: Query response process

The client TEE checks the integrity of the data using the proxy signature, then processes the data received using the loaded method. The final result is signed by the client TEE before being sent to the user.

The following sequence diagram illustrates the five steps described above.

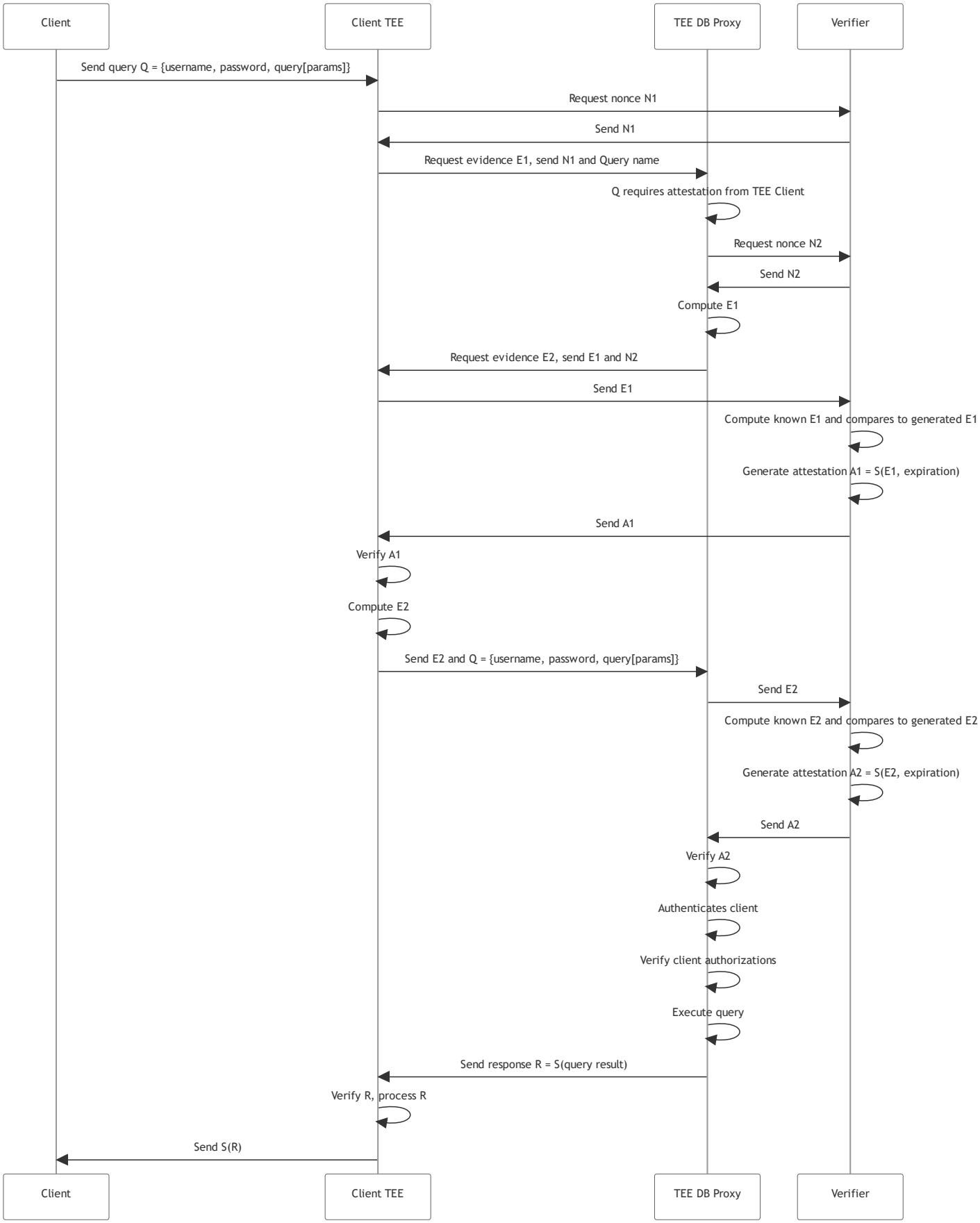


Figure 10: Sequence diagram representing a mutual attestation exchange.

3.5 Threat model

Now that the actors have been described and the interactions clarified, the STRIDE approach can be used to draw up a threat list and the resources to be put in place to deal with it. A more detailed threat analysis about the main threat, information disclosure, is available as Appendix III as a threat tree.

Type	Description	Security control	Mitigation technique
Spoofing	A threat to access and use another user's credentials, such as username and password.	Authentication	Implement appropriate authentication (MFA). Do not store secrets (password hash using argon2). Store secrets securely (Secure storage). Implement secret rotation. Account lockout after multiple failures.
Tampering	A threat aimed at modifying or falsifying persistent data (for example, database records) or data in transit.	Integrity	Ensure proper authorization for data access. Implement MAC or signatures. Secure communication channel (TLS).
Repudiation	A threat aimed at carrying out prohibited operations on a system without the ability to trace these operations.	Non-repudiation	Implement digital signatures. Maintain audit trail. Immutable logging (e.g. blockchain).
Information disclosure	A threat to read a file or data in transit for which access has not been granted.	Confidentiality	Enforce access control. Secure communication channel (TLS). Secret storage and protection. Privacy enhancing protocols (e.g.: Differential Privacy).
Denial of service	A threat aimed at preventing legitimate users from accessing a resource, for example by making a web server temporarily unavailable.	Availability	Authenticate and authorize all incoming requests. Apply rate limiting and request throttling. Use web application firewalls (WAFs). Implement auto-scaling for resilience. Design for graceful degradation under load.
Privilege escalation	A threat aimed at obtaining increased privileges to access unauthorized information or compromise a system.	Authorization	Apply the principle of least privilege (PoLP). Use privilege separation (e.g., different roles for admin and user functions). Harden systems and services (e.g., disable unnecessary features). Monitor privilege escalation attempts.

Table 6: STRIDE Threat model

4. Implementation

4.1 General description

This project aims to set up a proof of concept implementing secure interactions and a remote attestation mechanism. Although Trusted Execution Environments (TEEs) are simulated using classes, the implementation strives to remain faithful to real concepts in order to facilitate future extension. The main objective is to guarantee the confidentiality and integrity of data exchanged between a client and a database proxy.

The Python language was chosen for its ease of prototyping due to the author's affinity with it and its compatibility with the WebAssembly Micro Runtime (WAMR)[62], a lightweight environment adapted to scenarios where resources are limited, such as TEEs[63].

The implementation consists of several components:

- **Client with/without TEE**: Simulates entities that initiate requests and participate in the attestation process.
- **Database Proxy**: acts as an intermediary between the client and the database, ensuring that only authenticated and authorized requests are executed.
- **Generic Verifier**: Implements a simulated remote verification process to ensure trust in clients and their execution environment.

The next chapters describe the various elements required for a scenario where an external user (an insurance company, for example) wishes to know whether a patient's blood pressure is below the average. This average is determined based on data held by the Client (i.e. the insurance). This result can then be used to help the insurance company decide whether to reimburse the cost of a service or medication.

4.2 Communication

As mentioned above, the actors are classes that simulate either clients or servers. To enable them to communicate with each other, the PyWolfSSL library is used to implement TLS1.3 communication, which has already been used for projects using TEEs, for example in the implementation of a MQQT[64].

Each actor has a helper class for managing connections and binding sockets (named `TLS_helper`). It is responsible for establishing the TLS context and loading keys and contexts. The keys and certificates are generated and self-signed using OpenSSL, following the MariaDB documentation[65]. This enables each party to carry out their TLS handshake.

Once the connections have been established between the actors, they can send each other requests similar to a RESTful call. Here's an example of a query that the client sends to its TEE to determine whether the patient is well below average (without knowing the patient's blood pressure value).

[62] BYTECODE ALLIANCE, 2024. *WebAssembly Micro Runtime (WAMR)*.

[63] MENETREY, Jämes, PASIN, Marcelo, FELBER, Pascal [et al.], 2023. *A Comprehensive Trusted Runtime for WebAssembly With Intel SGX*.

[64] MENETREY, Jämes, GRUTER, Aeneas, YUHALA, Peterson [et al.], 2023. *A Holistic Approach for Trustworthy Distributed Systems with WebAssembly and TEEs*.

[65] MARIADB, n.d. *Create Self-Signed Certificates and Keys with OpenSSL*.

```
{
  "method": "GET",
  "route": "is_bp_under_mean",
  "username": "external_user_123",
  "password": "password",
  "params": {
    "patient_id": "foo-bar-quux"
  }
}
```

Figure 11: Example of a Client's request to the Proxy.

The use of Flask to facilitate the API is not intended for this approach. Indeed, it is not designed to implement a TLS context directly in the code, but rather to manage an HTTPS session through a WSGI server such as Gunicorn[66].

All the actors implement the methods required to comply with the mutual attestation schemes introduced in the previous chapter.

4.3 Data storage and manipulation

In order to best match the data storage to the JSON format, the choice of database was made in favor of MongoDB, so as to work easily between JSON and NoSQL. MongoDB has two other advantages:

- It allows users to be logged in using their password hash approach, Salted Challenge Response Authentication Mechanism (SCRAM) [67].
- It offers the use of aggregation pipelines[68] that can be stored and loaded.

The database contained in a docker image. A `*docker-compose*` file allows us to specify a username and password to set up an authentication process so that the proxy can access the data. These values are stored as environment variables. As mentioned above there are four central collections for the implementation plus a fifth allowing calculations for the client:

Users

They have a username and password. They also have a field associated with a role. This has no influence on the current state of the implementation. However, it remains relevant and will be developed in more detail in the future work chapter.

Patients

A document representing a patient is strongly inspired by the FHIR standard without, however, implementing it faithfully at the risk of overloading the complexity of this proof of concept. A more detailed overview is available in the project repository.

Access control

This table is central to the protection of user data. For this implementation, an Access Control List (ACL) approach was preferred and implemented instead of a Role Based Access Control (RBAC). An ACL is available in each of the nodes of a patient document. This ACL guarantees access only at its level. It can be used to grant specific access to data. An access control (AC) is made up of a list of users who are granted different rights (read, write and/or enclave). To increase security, authorizations also have an expiry date. This makes it possible to grant rights to a doctor only for the duration of a treatment, for example. This is what an access control document looks like:

---

[66] FLASK, n.d. *Deploying to Production*.

[67] MONGODB, n.d. *SCRAM*.

[68] MONGODB, n.d. *Aggregation Pipeline*.

```
{
    "_id": ObjectId('foo-bar-quux'),
    "users": [
        {
            "userId": doctor_id,
            "permissions": ["read", "write"],
            "expiration": datetime.datetime(YYYY, MM, DD),
        },
        {
            "userId": external_user_id,
            "permissions": ["enclave"],
            "expiration": datetime.datetime(YYYY, MM, DD),
        }
    ]
}
```

Figure 12: Example of an access control document.

This access control represents only one node. In the implementation, each node has its own AC in order to be as specific as possible, although it is possible to reference the node several times if it the same type of access and user is often given. However, care must be taken when modifying a document, as it can give access to more data. Another approach could have been to decide that when a user has access to a parent node, it automatically gains access to the child nodes. For the proof of concept a more granular approach was privileged rather than the latter one. Actual use of the data may show the need to specify child nodes or not.

It is now clear that if the node where the patient's blood pressure is stored, then for an external user to be able to access the resource it must be included in the ACL with at least one enclave right (and a valid expiry date).

Data for statistical purposes

When an external user wishes to access data to measure it, they must have a collection of (anonymized) data to compare. In the current example, the external user has a collection of blood pressure data to compare with that received.

Pipelines

Pipeline collections, whether for the Verifier, the Proxy or a client TEE, are all the same. It consists of the name of the pipeline and an array that can be loaded by the TEE. Pipelines are MongoDB stages that allow diverse types of operation on a collection or document using operators such as `$match`, `$group`, `$avg`.

In the followin example, the first stage computes the average blood pressure. The second stage returns -1, 0 or 1 depending on whether the input pressure is higher or lower.

The pipeline shown is basic because it doesn't control ACs. Another pipeline such as `get_bp`, which will be executed by the proxy to access a patient's blood pressure, is more consistent because it also performs authorization checks (no authentication). As this type of pipeline is much longer, the reader is invited to consult the project repository.

The various authorization stages check:

- Whether the user is the patient, and if so, whether the user can access the data.
    - If not, is the user in the ACL of the node where the data is stored?
    - If so, does the user have the corresponding right, which has not expired?
        - Only then is the data retrieved.

To strengthen the pipelines, a sanitization process is implemented before passing the parameters. The method is not exhaustive, but it ensures that parameters are parsed into their correct type (user id such as ObjectId or float for blood pressure, for example). This approach prevents injection attacks into the pipeline.

Pipelines are prepared as templates and can receive parameters. Here is an example:

```
{
    "_id": ObjectId(),
    "name": "is_bp_above_mean",
    "pipeline": [
        {
            "$group": {
                "_id": None,
                "mean_bp": {"$avg": "$bp"}
            }
        },
        {
            "$project": {
                "_id": 0,
                "mean_bp": 1,
                "is_above": {
                    "$cond": [
                        {"$gt": ["$input_bp", "$mean_bp"]},
                        "1",
                        {"$cond": [
                            {"$lt": ["$input_bp", "$mean_bp"]},
                            "-1",
                            "0"
                        ]}
                    ]
                }
            }
        }
    ],
},
```

First stage

Second stage

Figure 13: Example of a pipeline.

4.4 Additional security considerations

An emphasis on the enforcement of TLS1.3 is often mentioned in this report. This updated version of the protocol, released in 2018[69], speeds up handshake but above all has abandoned obsolete and therefore vulnerable algorithms. Specifying the context with PROTOCOL_TLSv1_3 prevents a downgrade attack that would allow an attacker to exploit vulnerabilities in earlier versions of these protocols. It is important to understand that the TLS context must be located within the TEE to ensure that the data encrypted by TLS is only visible within the TEE.

Simple logging of queries performed on the DB by the TEE is set up to ensure the traceability of actions with a timestamp, the user ID and the query parameters. The result is deliberately omitted to preserve confidentiality. A boolean confirming that the query has been executed correctly could be relevant in future works.

The cryptographic operations were carried out using Libsodium's PyNaCl library. This alternative to OpenSSL (whose python library is pending depreciation[70]) has already been deployed on a TEE without difficulty[71].

The signature of the various claims by the TEEs is currently an EdDSA, i.e. a digital signature on an elliptic curve (Ed25519) [72]. The signature keys generated by the `SigningKey` class are 256 bits long, which corresponds to near-term protection for operations on an elliptic curve (at least 2030 and beyond), according to the National Institute of Standards and Technology (NIST) and eCrypt[73]. This constraint is sufficient for a proof of concept and its implications will be discussed in the Future works chapter.

---

[69] WOLFSSL, n.d. *SSL/TLS Overview*.

[70] PYOPENSSL, n.d. *crypto — Generic cryptographic module*.

[71] LEE, Dayeol, KOHLBRENNER, David, SHINDE, Schweta, 2019. *Keystone: An Open Framework for Architecting TEEs*.

[72] PYNACL, n.d. *Digital Signatures*.

[73] GIRY, Damien, 2020. *Cryptographic key length recommendation*.

5. Evaluation

5.1 Performances

Now that the interactions have been set up, the time taken by the various operations carried out by the different players can be measured. To do this, the databases have all been populated by one million entries, which is equivalent to the load that a cantonal hospital would see for example (in 2023, the canton of Vaud would register around 800,000 residents and Zurich 1,500,000[74]). The data is generated pseudo-randomly, providing a finite number of possibilities[75]. The pipelines are then indexed by name and the medical data by patient ID. In order to facilitate testing, the database contains at least one user and patient with a known id, also in order to guarantee the result given by the query.

Performance will evaluate the time taken to receive a patient's blood pressure result. The first user is the patient himself. The execution time must be as short as possible, as there is no ACL control, since a first control checks if the id of the user running the query is equal to that in his medical file. Then there is the doctor, who is granted (at least) 'read' rights. Finally, an external user who wants to know whether the pressure is below average (based on his data) through his own TEE[76].

The first metrics measure the time taken when there is no attestation process. A client queries a server and the server calculates the data before sending it back (no pipeline). This approach could be described as naive. The second measure will calculate the times taken by the patient and the doctor with only a remote attestation (RA), then the external user with a mutual attestation (MA).

Finally, given that the project is not deployed on a TEE, a hypothetical overhead to simulate the time taken by the TEE was added. An optimistic estimate of the overhead valued at 10%, neutral at 50%[77], and pessimistic at 100% was added. To see which operations might take time on a TEE, these have been measured separately to obtain the total time as well as the sub-time. Only the sub-times were increased (and the total time proportionally).

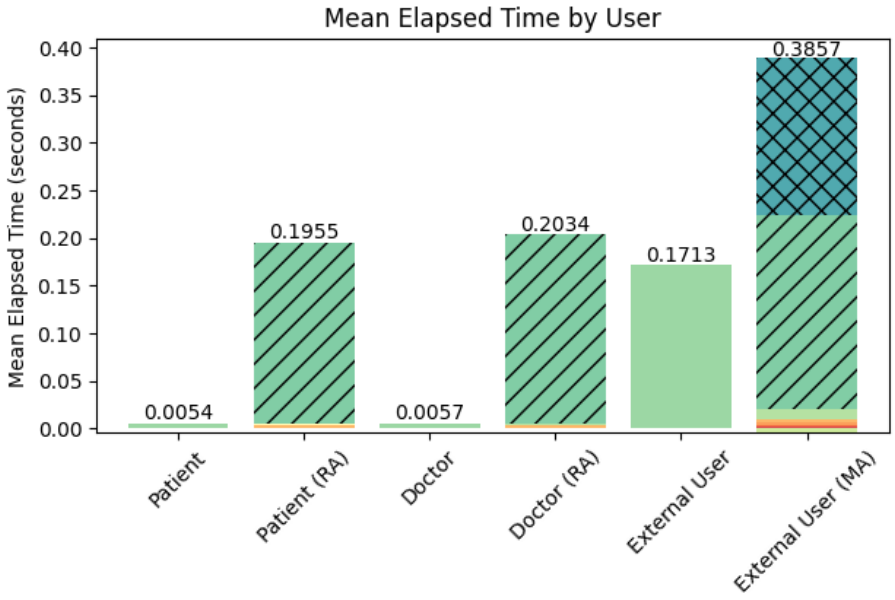


Figure 14: Mean by users without estimated overhead.

[74] FEDERAL STATISTICAL OFFICE, 2023. *Bilan de la population résidante permanente, par canton et ville, de 1999 à 2023.*

[75] By providing the expected data format to ChatGPT the process of populating the data has been facilitated.

[76] Data and graphs are available in a Jupyter Notebook in the project repository.

[77] MENETREY, Jämes, GRUTER, Aeneas, YUHALA, Peterson [et al.], 2023. *A Holistic Approach for Trustworthy Distributed Systems with WebAssembly and TEEs*.

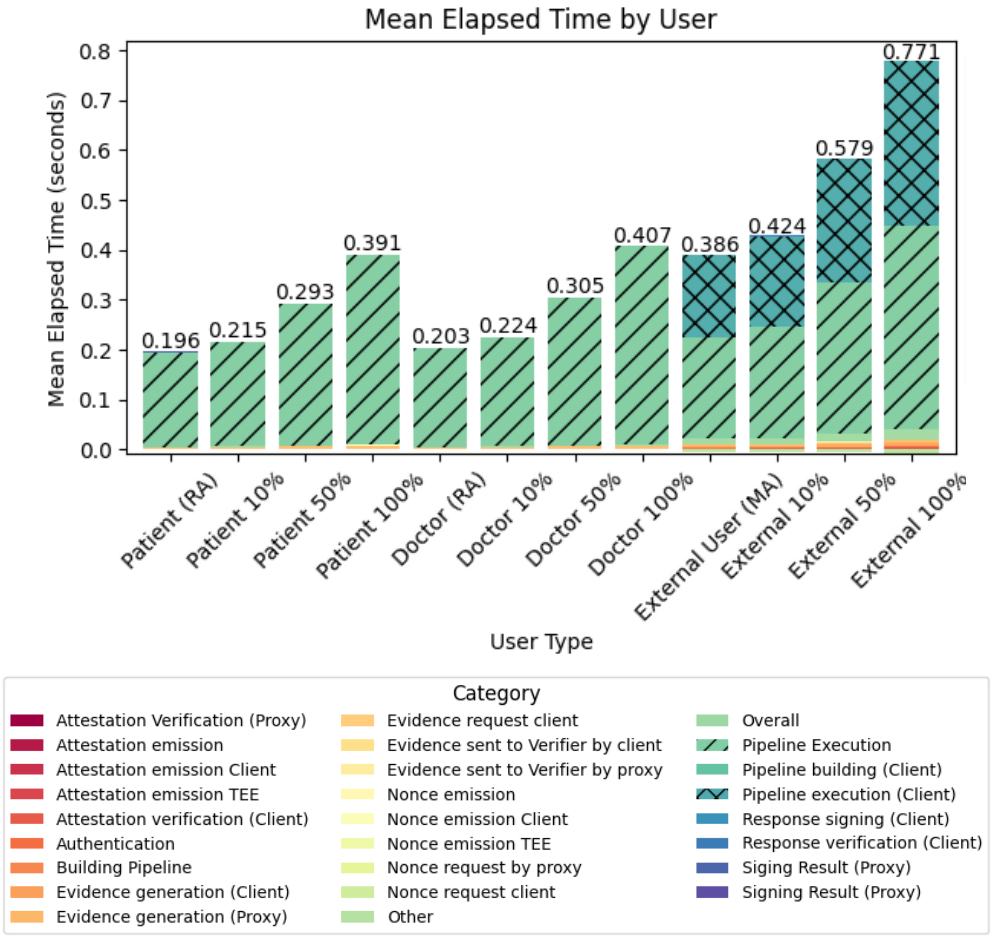


Figure 15: Mean by user with 10%, 50% and 100% overhead on TEE operations (with corresponding labels)

The graphs shown appear imposing, given the number of labels. However, they show that the most time-consuming operations are the execution of pipelines, whether on the Client or Proxy side (hatched labels). However, a comparison between the execution time of a pipeline and the total time of an external user without mutual attestation suggests that this latency is partly due to the aggregation of a massive volume of data (for example, the reduction operation performed on pressure values to calculate the average).

Another factor explaining the latency of this section is the very construction of the pipelines. They have not been developed with performance in mind, but rather with a functional proof of concept in mind. This element is exacerbated by the use of ChatGPT to assist in the crafting of the various pipelines. Although the use cases and tests carried out demonstrate that the pipelines correctly apply access controls, they still have significant room for optimization. Future work on it will tell how much better a human crafted pipeline with optimization in mind will contribute to an increased performance.

A further indicator of this imperfect optimization is the difference in latency between reduction operations performed on the Client side and data recovery operations on the Proxy side. On average, the Proxy is 1.185 times slower than the Client. Here are two future leads for pipeline optimization:

1. Eliminate redundant operations when checking access permissions.
2. Modularize processing steps to make ACL control blocks reusable. Indeed, although ACL access paths may vary, the underlying control logic remains the same.

However, it's reasonable to assume that pipeline execution will remain a bottleneck, due to the volume of data processed.

Looking at the second graph, one can see that, despite the addition of simulated time for the various steps performed on a TEE, not including pipelines, the time spent is around 0.036 seconds and the total estimated time is 0.771 seconds.

One factor not yet taken into account in this estimate is the time required for authentication. At present, this is rudimentary in implementation, but its integration with a robust mechanism, such as Argon2, will have a significant impact on latency. Indeed, due to anti-bruteforce measures against GPU attacks, Argon2's Python library documentation[78] estimates that a password check in a high-memory environment (2GB) takes around 866.5ms. By replicating their approach in a low memory flag passing environment, more suited to resource-constrained systems, the execution time drops to 38.5ms, although this reduction comes with a compromise in data security.

Finally, it's important to note that these estimates don't take full account of the costs associated with `ecalls` and `ocalls` in an Intel SGX enclave. These context transitions between the secure environment and user space are known to be costly operations in terms of performance.

5.2 Threat model considerations

As part of this proof of concept, a threat model was defined to identify potential risks and evaluate the protection measures in place. This model is based on a STRIDE approach based on the OWASP recommendations.

The aim of this section is to compare the current implementation with the requirements set out in the threat model. It is important to examine the extent to which the mechanisms in place meet these requirements and identify any improvements that need to be made to enhance system security.

Type	Security control	Mitigation technique	Mitigation implemented
Spoofing	Authentication	Implement appropriate authentication (MFA).	○
		Do not store secrets (password hash using argon2).	○
		Store secrets securely (Secure storage).	◑
		Implement secret rotation.	○
		Account lockout after multiple failures.	○
Tampering	Integrity	Ensure proper authorization for data access.	●
		Implement MAC or signatures.	●
		Secure communication channel (TLS).	●
Repudiation	Non-repudiation	Implement digital signatures.	●
		Maintain audit trail.	◑
		Immutable logging (e.g. blockchain).	◑
Information disclosure	Confidentiality	Enforce access control.	●
		Secure communication channel (TLS).	●
		Secret storage and protection.	◑
		Privacy enhancing protocols (e.g.: Differential Privacy).	○
Denial of service	Availability	Authenticate and authorize all incoming requests.	○
		Apply rate limiting and request throttling.	○
		Use web application firewalls (WAFs).	○
		Implement auto-scaling for resilience.	○

[78] ARGON2-CFFI, n.d. *API Reference*.

		Design for graceful degradation under load.	○
Privilege escalation	Authorization	Apply the principle of least privilege (PoLP).	○
		Use privilege separation (e.g., distinct roles for admin and user functions).	○
		Harden systems and services (e.g., disable unnecessary features).	○
		Monitor privilege escalation attempts.	○
○ = not implemented		◑ = partially implemented	● = fully implemented

Table 7: Threat model implementation assessment.

An analysis of the table shows that the fundamental principles of integrity, non-repudiation and confidentiality have been implemented. Whether through an ACL, connection via TLS1.3 or signature of the various responses and logging. As mentioned on several occasions, user authentication has been set aside for the time being.

Another crucial aspect to consider is the rotation of cryptographic keys. As mentioned above, the current key length (256 bits) provides adequate short-term security, but the adoption of 512-bit keys would significantly improve long-term protection. In addition, NIST recommendations indicate specific lifetimes for all types of keys[79]. To guarantee the security of a private signing key, it is recommended to renew these keys every 1 to 3 years, depending on system requirements and operational constraints. Of course, public verification keys do not expire, so as to preserve the possibility of verifying a signature.

In addition to the aspects already mentioned, some specific threats remain to be addressed, in particular attacks linked to privilege escalation and denial of service (DoS) attacks. Although TEEs offer an enhanced level of isolation, they are not immune to these types of attack[80]. The lack of access to OS system calls means that an external library has to be imported, or a proprietary solution has to be implemented, which widens the attack vector. They are also vulnerable to side channel attacks, such as fault injections. The aim is to induce physical or software-based faults during computation in order to expose secrets. Electromagnetic Fault Injection (EMFI) attacks are among the most effective and challenging to mitigate[81], successfully targeting numerous commercial integrated circuits. A notable category of fault-injection attacks involves Dynamic Voltage and Frequency Scaling (DVFS), which enables software to adjust a device's voltage and frequency per CPU thread, thereby influencing and monitoring power consumption[82]. Given the broad spectrum of this type of attack, these aspects have been set aside in the first iteration of this proof of concept. In the future, the robustness of the system against these attacks will need to be enhanced by incorporating protection mechanisms such as fault detection mechanisms, hardware and software countermeasures, and in-depth analysis of abnormal behavior within the TEE.

[79] GIRY, Damien, 2020. *Cryptographic key length recommendation*.

[80] SUCIU, Darius, MCLAUGHLIN, Stephen, SIMON, Laurent [et al.], 2020. *Horizontal Privilege Escalation in Trusted Applications*.

[81] MAISTRI, Paolo, LEVEUGLE, Régis, BOSSUET, Lilian, 2015. *Electromagnetic analysis and fault injection onto secure circuits*.

[82] MUNOZ, Antonio, RIOS, Ruben, ROMAN, Rodrigo, 2023. *A survey on the (in)security of trusted execution environments*.

5.3 Future work

In addition to the various deepening measures mentioned in previous chapters, such as:

- Pipeline optimization;
- Key rotation;
- Missing elements of the threat model (authentication, DoS, privilege escalation).

A number of areas for improvement and future extensions can be envisaged to perfect the proposed solution. At present, access management relies on list-based access control (ACL), requiring explicit definition of permissions for each field. A transition to a RBAC (Role-Based Access Control) model would enable more flexible management of access rights according to user roles (general practitioners, specialists, researchers, etc.).

A more advanced alternative would be to integrate Machine Learning (ML) to dynamically determine the relevant accesses according to the doctor's profile. This type of approach has been explored in the protection of medical data, particularly for monitoring access and detecting anomalies[83]with promising results. An ML model trained on real cases could thus enable fine-grained access management, without requiring explicit specification of permissions for each field. The addition of ML in the design of approved models rather than pipelines in the determination of benefits could also be considered, obviously bearing in mind the points raised in chapter 2.4.7.

The security of cryptographic algorithms must also be constantly monitored. The emergence of quantum computing represents a direct threat to algorithms based on moduli factorization (such as RSA) and elliptic curves, particularly for the signature currently implemented, because of the vulnerability induced by the Shor algorithm[84].

Post-quantum solutions are already beginning to be explored, such as the integration of Falcon, a signature scheme resistant to quantum attacks, on the Keystone framework running on a RISC-V TEE[85]. The latest NIST report[86] on post-quantum cryptography stresses that it is not yet necessary to adopt SHA-3, but that a lengthening of the output size is recommended to guarantee greater resilience in the face of future advances. Even if quantum computing is not yet a technology ready for widespread deployment, the threat of 'store now, decrypt later' encourages the implementation of protections against quantum computing.

Obviously, one of the shorter-term objectives would be to test the solution on a real TEE, such as Intel SGX, ARM TrustZone or AMD SEV. This would enable us to assess the impact of hardware constraints and performance on the various modules of the system. A comparative analysis between different TEEs could also be carried out to determine the platform best suited to the project's requirements.

---

[83] HUSSAIN, Adil, AL-AMRI, Jehad f., SUBAHI, Ahmad F. [et al.], 2021. *An Analysis of Integrating Machine Learning in Healthcare for Ensuring Confidentiality of the Electronic Records*.

[84] ROETTELER, Martin, NAEHRIG, Michael, SVORE, Krysta M. [et al.], 2017. *Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms*.

[85] CARUSO, Giuseppe, 2024. *Post-quantum algorithms support in Trusted Execution Environment*.

[86] CHEN, Lily, JORDAN, Stephen, LIU, Yi-Kai [et al.], 2016. *Report on Post-Quantum Cryptography*.

6. Conclusions

Through the various chapters, the complex environment of digital health in Switzerland was explored and the challenges associated with accessing and sharing medical data highlighted. This problem is particularly well illustrated by the form in the appendix, which highlights the need for insurance companies to accept a treatment or disability insurance (DI) claim. The new Federal Act on Data Protection (nFADP) provides a framework for data protection. However, the fear of unauthorized access to medical data is encouraging the development of a secure IT environment in order to guarantee its confidentiality and integrity.

It is in this context that the integration of Trusted Execution Environments (TEEs) fits in. A proof of concept has been established in this project in order to present an architecture and show how an actor can make a decision without having access to the data itself but relying solely on a cryptographic guarantee of the result. Using the Remote attestation procedures (RATS) Architecture RFC, the demonstration was done how a client, whether using a TEE or not, can obtain compliant data. This approach is also based on the Electronic Patient Record (EPR) standards and the FHIR standard to facilitate future extension. Data confidentiality relies on authentication and authorization validation, based on an Access Control List (ACL).

The system's modularity is ensured by certified methods, implemented in the form of MongoDB pipelines. This approach enables flexible data processing, but also introduces a performance cost (overhead), as each query requires remote attestation in the form of a background check. This model was evaluated using a STRIDE approach, establishing a threat model. The analysis shows that confidentiality, integrity and non-repudiation are currently well respected in the implementation. However, threats remain, particularly related to authentication, privilege escalation and denial of service (DoS) attacks, which will need to be addressed in future development.

An estimate of the overhead was also made by analyzing in detail the operations carried out by a simulated TEE. These measures allowed us to identify the most time-consuming processes, revealing that the pipelines were the primary bottleneck. These findings highlight the need for further optimizations to enhance their efficiency. An essential next step would be to test the solution on a real TEE, such as Intel SGX, ARM TrustZone or AMD SEV, in order to assess the impact of hardware constraints and performance on the various modules of the system. A comparative analysis of these technologies would also help to identify the platform best suited to the needs of the project. Additional approaches for future development were presented, such as the use of machine learning or concepts such as differential privacy to protect data.

Although TEEs offer a major advance in data protection, they also have significant limitations: limited resources, performance overheads, a change in programming paradigm and residual vulnerabilities to attacks such as side-channel attacks. Nevertheless, their gradual adoption and the evolution of secure architectures show that they will play a key role in the future of cybersecurity and the protection of sensitive data, such as medical data.

7. Bibliography

ACAR, Abbas, AKSU, Hidayer, ULUAGAC, A. Selcuk, 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. In: *ACM Comput. Surv.* 51, 4, Article 79 (July 2019), 35 pages. Available: https://doi.org/10.1145/3214303

AMD, n.d. AMD Secure Encrypted Virtualization (SEV). *AMD website* [online]. Available: https://www.amd.com/fr/developer/sev.html [Accessed on 4 February 2025]

ANDRYSCO, Marc, KOHLBRENNER, David, MOWERY, Keaton [et al.], 2015. On Subnormal Floating Point and Abnormal Timing. In: *IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2015, pp. 623-639, doi: 10.1109/SP.2015.44. Available: https://ieeexplore.ieee.org/document/7163051

ARGON2-CFFI, n.d. API Reference. *Argon2-cffi documentation* [online]. Available: https://argon2-cffi.readthedocs.io/en/stable/api.html [Accessed on 4 February 2025]

ARM, n.d. ARM TrustZone technology. *ARM Developer website* [online]. Available: https://developer.arm.com/documentation/100690/0200/ARM-TrustZone-technology [Accessed on 4 February 2025]

ARM, 2015. Securing the Future of Authentication with ARM TrustZone-based Trusted Execution Environment and Fast Identity Online (FIDO). *ARM white paper* [online]. Available *:* https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/TrustZone-and-FIDO-white-paper.pdf?revision=98e6ae26-92ca-4ffd-ac4e-3329b7f8a23e

BIRKHOLZ, Henk, THALER, Dave, RICHARDSON, Michael [et al.], 2023. RFC 9334 Remote ATtestation procedureS (RATS) Architecture. *Internet Engineering Task Force (IETF) website* [online]. Available: https://datatracker.ietf.org/doc/html/rfc9334

BYTECODE ALLIANCE, 2024. WebAssembly Micro Runtime (WAMR). *Github repository* [online]. Last updated on the 27 November 2024. Available: https://github.com/bytecodealliance/wasm-micro-runtime [Accessed on 4 February 2025]

CARUSO, Giuseppe, 2024. *Post-quantum algorithms support in Trusted Execution Environment* [online]. Torino: Politecnico di Torino. Master degree thesis. Available: https://webthesis.biblio.polito.it/secure/31076/1/tesi.pdf

CDS, n.d. La CDS. *CDS website* [online]. Available: https://www.gdk-cds.ch/fr/ [Accessed on 4 February 2025]

CHEN, Hannah S., JARRELL, Juliet T., CARPENTER, Kristy A [et al.], 2019. *Blockchain in Healthcare: A Patient-Centered Model*. Biomedical journal of scientific & technical research, 20(3), 15017–15022. Available: https://pubmed.ncbi.nlm.nih.gov/31565696/

CHEN, Lily, JORDAN, Stephen, LIU, Yi-Kai [et al.], 2016. Report on Post-Quantum Cryptography. NISTIR 8105. Available: http://dx.doi.org/10.6028/NIST.IR.8105

CLEISS, 2021. Le système de santé suisse. *CLEISS website* [online]. Available: https://www.cleiss.fr/docs/systemes-de-sante/suisse.html [Accessed on 4 February 2025]

CONKLIN, Larry, 2024. Threat Modeling Process. *OWASP website* [online]. Available: https://owasp.org/www-community/Threat_Modeling_Process [Accessed on 4 February 2025]

DISABILITY INSURANCE CANTON VAUD, 2023. *Form 002.099 for professional readaptation*. Available: https://aivd.ch/formulaires/

DUY, Kha Dinh, NOH, Taehyun, HUH, Siwon [et al.], 2021. Confidential Machine Learning Computation in Untrusted Environments: A Systems Security Perspective. In: *IEEE Access*, vol. 9, pp. 168656-168677, 2021, doi: 10.1109/ACCESS.2021.3136889. Available: https://ieeexplore.ieee.org/abstract/document/9656734

EHEALTH SWITZERLAND, n.d. Activities since 2008. *eHealth website* [online]. Available: https://www.e-health-suisse.ch/en/coordination/information/activities-since-2008 [Accessed on 4 February 2025]

EHEALTH SWITZERLAND, n.d. eHealth Suisse. *eHealth Switzerland website* [online]. Available: https://www.e-health-suisse.ch/fr/a-propos/ehealth-suisse [Accessed on 4 February 2025]

EHEALTH SWITZERLAND, n.d. Protection des données. *Switzerland's EPR website* [online]. Available: https://www.dossierpatient.ch/population/dep-securite/protection-donnees [Accessed on 4 February 2025]

EHEALTH SWITZERLAND, 2018. *Stratégie Cybersanté Suisse 2.0. 2018 –2024*. [online]. Bern: eHealth Switzerland. DOI 2.16.756.5.30.1.127.1.1.5.1.1. Available: https://www.bag.admin.ch/dam/bag/fr/dokumente/nat-gesundheitsstrategien/strategie-ehealth/ehs-strategie-de-230127.pdf.download.pdf/eHS_Strategie_FR_230127_barrierefrei.pdf

EHEALTH SWITZERLAND, n.d. *Switzerland's EPR website* [online]. Available: https://www.patientrecord.ch/upload/images/No_access_EN-1080x763.png [Accessed on 4 February 2025]

EHEALTH SWITZERLAND, n.d. The EPR in short. *eHealth website* [online]. Available: https://www.dossierpatient.ch/population/dep-en-bref [Accessed on 4 February 2025]

FEDERAL DATA PROTECTION AND INFORMATION COMMISSIONER (FDPIC), 2024. Frequently asked questions on data protection concerns. *The portal of the Swiss government* [online]. Available: https://www.edoeb.admin.ch/fr/questions-frequemment-posees [Accessed on 4 February 2025]

FEDERAL DEPARTMENT OF ECONOMIC AFFAIRS, EDUCATION AND RESEARCH (EAER), 2024. New Federal Act on Data Protection (nFADP). *The portal of the Swiss government* [online]. Available: https://www.kmu.admin.ch/kmu/fr/home/faits-et-tendances/digitalisation/protection-des-donnees/nouvelle-loi-sur-la-protection-des-donnees-nlpd.html [Accessed on 4 February 2025]

FEDERAL STATISTICAL OFFICE, 2023. Bilan de la population résidante permanente, par canton et ville, de 1999 à 2023. *The portal of the Swiss government* [online]. Available: https://www.bfs.admin.ch/bfs/fr/home/statistiques/population.assetdetail.32229068.html [Accessed on 4 February 2025]

FLASK, n.d. Deploying to Production. *Flask documentation [*online]. Available: https://flask.palletsprojects.com/en/stable/deploying/ [Accessed on 4 February 2025]

FMH, n.d. À propos de la FMH. *FMH website* [online]. Available: https://www.fmh.ch/fr/a-propos-de-la-fmh.cfm [Accessed on 4 February 2025]

FMH, n.d. Dossier électronique du patient (DEP). *FMH website* [online]. Available: https://www.fmh.ch/fr/themes/ehealth/dossier-electronique-patient.cfm [Accessed on 4 February 2025]

FOPH, 2024. Further development of the electronic patient record. *The portal of the Swiss government* [online]. Last updated on the 1st October 2024. Available: https://www.bag.admin.ch/bag/fr/home/strategie-und-politik/nationale-gesundheitsstrategien/strategie-ehealth-schweiz/umsetzung-vollzug/weiterentwicklung-epd.html [Accessed on 4 February 2025]

FOPH, n.d. Health insurance. *The portal of the Swiss government* [online]. Available: https://www.bag.admin.ch/bag/fr/home/versicherungen/krankenversicherung.html [Accessed on 4 February 2025]

FOPH, 2024. Health insurance: Key points in brief. *The portal of the Swiss government* [online]. Last updated on the 12 September 2024. Available: https://www.bag.admin.ch/bag/fr/home/versicherungen/krankenversicherung/krankenversicherung-das-wichtigste-in-kuerze.html [Accessed on 4 February 2025]

FOPH, 2024. Health insurance: Supervision of insurers. Last updated on the 27 June 2024. *The portal of the Swiss government* [online]. Available: https://www.bag.admin.ch/bag/fr/home/versicherungen/krankenversicherung/krankenversicherung-versicherer-aufsicht.html [Accessed on 4 February 2025]

FHOP, n.d. Legislation. *The portal of the Swiss government* [online]. Available: https://www.bag.admin.ch/bag/fr/home/gesetze-und-bewilligungen/gesetzgebung.html

FOPH, 2024. Loi fédérale sur l'assurance-maladie (LAMal). *The portal of the Swiss government* [online]. Last updated on the 15 July 2024. Available: https://www.bag.admin.ch/bag/fr/home/gesetze-und-bewilligungen/gesetzgebung/gesetzgebung-versicherungen/gesetzgebung-krankenversicherung/kvg.html [Accessed on 4 February 2025]

GIRY, Damien, 2020. Cryptographic key length recommendation. *BlueKrypt website* [online]. Available: https://www.keylength.com/en/4/ [Accessed on 4 February 2025]

GOETTEL, Christian, PIRES, Rafael, ROCHA, Isabelly [et al.], 2018. Security, Performance and Energy Trade-offs of Hardware-assisted Memory Protection Mechanisms. 133-142. 10.1109/SRDS.2018.00024. Available: https://www.researchgate.net/publication/330472504_Security_Performance_and_Energy_Trade-Offs_of_Hardware-Assisted_Memory_Protection_Mechanisms

GREAVES, Duncan, 2019. Data Anonymisation and L-Diversity. *Information with Insight* [online]. Available: https://informationwithinsight.com/2019/03/12/data-anonymisation-and-l-diversity/ [Accessed on 4 February 2025]

GREAVES, Duncan, 2017. k-anonymity and SQL Server. *Information with Insight* [online]. Available: https://informationwithinsight.com/2017/01/09/k-anonymity-and-sql-server/ [Accessed on 4 February 2025]

HL7 INTERNATIONAL, 2023. Base Resource Definitions. *HL7 documentation* [online]. Available: https://www.hl7.org/fhir/resource.html [Accessed on 4 February 2025]

HL7 SWITZERLAND, 2024. CH Core (R4). *FHIR documentation* [online]. Available: https://fhir.ch/ig/ch-core/index.html [Accessed on 4 February 2025]

HL7 INTERNATIONAL, 2023. Clinical Safety. *HL7 documentation* [online]. Available: https://www.hl7.org/FHIR/safety.html [Accessed on 4 February 2025]

HL7 INTERNATIONAL, 2023. FHIR Overview. *HL7 documentation* [online]. Available: https://www.hl7.org/fhir/overview.html [Accessed on 4 February 2025]

HL7 INTERNATIONAL, 2025. Implementation Guide Registry. *FHIR documentation* [online]. Available: https://fhir.org/guides/registry/ [Accessed on 4 February 2025]

HL7 INTERNATIONAL, n.d. Norme internationale pour l'échange électronique de données médicales. *HL7 website* [online]. Available: https://www.hl7.ch/fr/ [Accessed on 4 February 2025]

HL7 INTERNATIONAL, 2025. SMART on FHIR Obligations and Capabilities. *FHIR documentation* [online]. Available: https://build.fhir.org/ig/HL7/US-Core//scopes.html [Accessed on 4 February 2025]

HL7 SWITZERLAND, 2024. Resource Profile: CH Core Patient. *FHIR documentation* [online]. Available : https://fhir.ch/ig/ch-core/StructureDefinition-ch-core-patient.html [Accessed on 4 February 2025]

HUSSAIN, Adil, AL-AMRI, Jehad f., SUBAHI, Ahmad F. [et al.], 2021. An Analysis of Integrating Machine Learning in Healthcare for Ensuring Confidentiality of the Electronic Records. In: *Engineering & Sciences*. 130. 1387-1422. Available: https://www.researchgate.net/publication/357502082_An_Analysis_of_Integrating_Machine_Learning_in_Healthcare_for_Ensuring_Confidentiality_of_the_Electronic_Records

Hypervisor, *Wikipedia* [online]. Last edited on 8 January 2025, at 10:38. Available: https://en.wikipedia.org/wiki/Hypervisor [Accessed on 4 February 2025]

INTEL, n.d. Intel® Software Guard Extensions (Intel® SGX). *Intel website* [online]. Available : https://www.intel.fr/content/www/fr/fr/products/docs/accelerator-engines/software-guard-extensions.html [Accessed on 4 February 2025]

k-anonymity, *Wikipedia* [online]. Last edited on 30 July 2024, at 06:41. Available: https://en.wikipedia.org/wiki/K-anonymity [Accessed on 4 February 2025]

KIM, Katherine K., JOSEPH, Jill G., OHNO-MACHADO, Lucila, 2015. Comparison of consumers' views on electronic data sharing for healthcare and research. In: *Journal of the American Medical Informatics Association: JAMIA*, 22(4), 821–830. Available: https://doi.org/10.1093/jamia/ocv014

LEE, Dayeol, KOHLBRENNER, David, SHINDE, Schweta, 2019. Keystone: An Open Framework for Architecting TEEs. Available: https://doi.org/10.48550/arXiv.1907.10119

LIN, Wilfred W. K. Lin, 2023. Challenges of Homomorphic encryption. Available: https://www.researchgate.net/publication/370050235_Challenges_of_Homomorphic_encryption

MAISTRI, Paolo, LEVEUGLE, Régis, BOSSUET, Lilian, 2015. Electromagnetic analysis and fault injection onto secure circuits. In: *22nd International Conference on Very Large Scale Integration (VLSI-SoC)*, Playa del Carmen, Mexico, 2014, pp. 1-6, doi: 10.1109/VLSI-SoC.2014.7004182. Available: https://ieeexplore.ieee.org/document/7004182

MARIADB, n.d. Create Self-Signed Certificates and Keys with OpenSSL. *MariaDB documentation* [online]. Available: https://mariadb.com/docs/server/security/data-in-transit-encryption/create-self-signed-certificates-keys-openssl/ [Accessed on 4 February 2025]

MENETREY, Jämes, GRUTER, Aeneas, YUHALA, Peterson [et al.], 2023. A Holistic Approach for Trustworthy Distributed Systems with WebAssembly and TEEs. In: *OPODIS'23: Proceedings of the 27th Conference on Principles of Distributed Systems*, Tokyo, Japan, December 2023. Available: https://doi.org/10.48550/arXiv.2312.00702

MENETREY, Jämes, PASIN, Marcelo, FELBER, Pascal [et al.], 2023. A Comprehensive Trusted Runtime for WebAssembly With Intel SGX. In: *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 3562-3579, July-Aug. 2024, doi: 10.1109/TDSC.2023.3334516. Available: https://ieeexplore.ieee.org/abstract/document/10323228

MISRA, Subhas Chandra, BHAVSAR, Virendrakumar C., 2003. *Relationships Between Selected Software Measures and Latent Bug-Density: Guidelines for Improving Quality*. 724-732. 10.1007/3-540-44839-X_76. Available: https://www.researchgate.net/publication/221434375_Relationships_Between_Selected_Software_Measures_and_Latent_Bug-Density_Guidelines_for_Improving_Quality

MONGODB, n.d. Aggregation Pipeline. *MongoDB documentation* [online]. Available : https://www.mongodb.com/docs/manual/core/aggregation-pipeline/ [Accessed on 4 February 2025]

MONGODB, n.d. SCRAM. *MongoDB documentation* [online]. Available : https://www.mongodb.com/docs/manual/core/security-scram/ [Accessed on 4 February 2025]

MOUTON DOREY, Corine, BAUMANN, Holger, BILLER-ANDORNO, Nikola, 2018. Patient data and patient rights: Swiss healthcare stakeholders' ethical awareness regarding large patient data sets – a qualitative study. In: *MC Med Ethics 19, 20* (2018). Available: https://doi.org/10.1186/s12910-018-0261-x

MUNOZ, Antonio, RIOS, Ruben, ROMAN, Rodrigo, 2023. A survey on the (in)security of trusted execution environments. In: *Computers & Security*, Volume 129, 2023, 103180, ISSN 0167-4048. Available: https://doi.org/10.1016/j.cose.2023.103180

NASR, Milad, CARLINI, Nicholas, HAYASE, Jonathan, 2023. Extracting Training Data from ChatGPT. Available : https://not-just-memorization.github.io/extracting-training-data-from-chatgpt.html [Accessed on 4 February 2025]

NASR, Milad, CARLINI, Nicholas, HAYASE, Jonathan, 2023 Scalable Extraction of Training Data from (Production) Language Models. Available: https://arxiv.org/pdf/2311.17035

PAINE, Kirsty, 2024. Homomorphic Encryption: How It Works. *Splunk, a cisco company website* [online]. Available: https://www.splunk.com/en_us/blog/learn/homomorphic-encryption.html [Accessed on 4 February 2025]

POLTAVTSEVA, M. A., RUDNITSKAYA, E. A., 2024. Confidentiality of Machine Learning Models. In: *Aut. Control Comp. Sci.* 57, 975–982 (2023). https://doi.org/10.3103/S0146411623080242

PONOMAREVA, Natalia, HAZIMEH, Hussein, KURAKIN, Alex [et al.], 2023. How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy. In: *Journal of Artificial Intelligence Research* 77 (2023) 1113-1201. Available: https://doi.org/10.1613/jair.1.14649

PYNACL, n.d. Digital Signatures. *PyNaCl documentation* [online]. Last updated on the 1st January 2025. Available: https://pynacl.readthedocs.io/en/latest/signing/ [Accessed on 4 February 2025]

PYOPENSSL, n.d. crypto — Generic cryptographic module. *PyOpenSSL documentation* [online]. Available: https://www.pyopenssl.org/en/latest/api/crypto.html#elliptic-curves [Accessed on 4 February 2025]

REDDY, Raksha, WALLACE, Carl, 2010. RFC 6024 Trust Anchor Management Requirements. *Internet Engineering Task Force (IETF) website* [online]. Available: https://datatracker.ietf.org/doc/html/rfc6024

ROETTELER, Martin, NAEHRIG, Michael, SVORE, Krysta M. [et al.], 2017. Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. In: *Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science(),* vol 10625. Springer, Cham. https://doi.org/10.1007/978-3-319-70697-9_9

SAKK, 2019. Avis juridique sur la légalité du refus de prise en charge de certains coûts par les caisses d'assurance-maladie, publié dans Jusletter. *SAKK website* [online]. Available: https://www.sakk.ch/fr/nouvelles/avis-juridique-sur-la-legalite-du-refus-de-prise-en-charge-de-certains-couts-par-les [Accessed on 4 February 2025]

SONG, Congzheng, RISTENPART, Thomas, SHMATIKOV, Vitaly, 2017. Machine Learning Models that Remember Too Much. Available: https://arxiv.org/pdf/1709.07886

SUCIU, Darius, MCLAUGHLIN, Stephen, SIMON, Laurent [et al.], 2020. Horizontal Privilege Escalation in Trusted Applications. In: *29th USENIX Security Symposium (USENIX Security 20).* Available: https://www.usenix.org/conference/usenixsecurity20/presentation/suciu

SWISSMEDIC, 2019. Swissmedic, Swiss Agency for Therapeutic Products. *Swissmedic website* [online]. Available: https://www.swissmedic.ch/swissmedic/fr/home/notre-profil/swissmedic--institut-suisse-des-produits-therapeutiques.html [Accessed on 4 February 2025]

SWITZERLAND, 2023. *Federal Act on Data Protection* [online]. Entered into force on 25 September 2020. Updated 1st September 2023. RO 2022 491. Available: https://www.fedlex.admin.ch/eli/cc/2022/491/fr

SWITZERLAND, 2024. *Loi fédérale sur le dossier électronique du patient* [online]. Entered into force on 19 June 2015. Updated 1st October, 2024. RO 2017 2201. Available: https://www.fedlex.admin.ch/eli/cc/2017/203/fr

YAQOOB, Ibrar, SALAH, Khaled, JAYARAMAN, Raja [et al.], 2022. Blockchain for Healthcare Data Management: Opportunities, Challenges, and Future Recommendations. In: *Neural Computing and Applications*. 34. 10.1007/s00521-020-05519-w. Available: https://www.researchgate.net/publication/345383000_Blockchain_for_Healthcare_Data_Management_Opportunities_Challenges_and_Future_Recommendations

WOLFSSL, n.d. SSL/TLS Overview. *WolfSSL documentation* [online]. Available : https://www.wolfssl.com/documentation/manuals/wolfssl/appendix04.html [Accessed on 4 February 2025]

ZHANG, Wenping, XU, Ruiyun, ZHAO, J. Leon [et al.], 2023. A Blockchain-centric Data Sharing Framework for Building Trust in Healthcare Insurance. In: *C. Lu, & M. Tanniru (Eds.), Blockchain in Healthcare: Analysis, Design and Implementation* (pp. 101-118). Springer. Available: https://doi.org/10.1007/978-3-031-45339-7

8. Figure list – Table list

Appendix

Link to project repository: https://github.com/WindRider97/TM

Appendix I – Interactions between a patient, healthcare provider and insurer

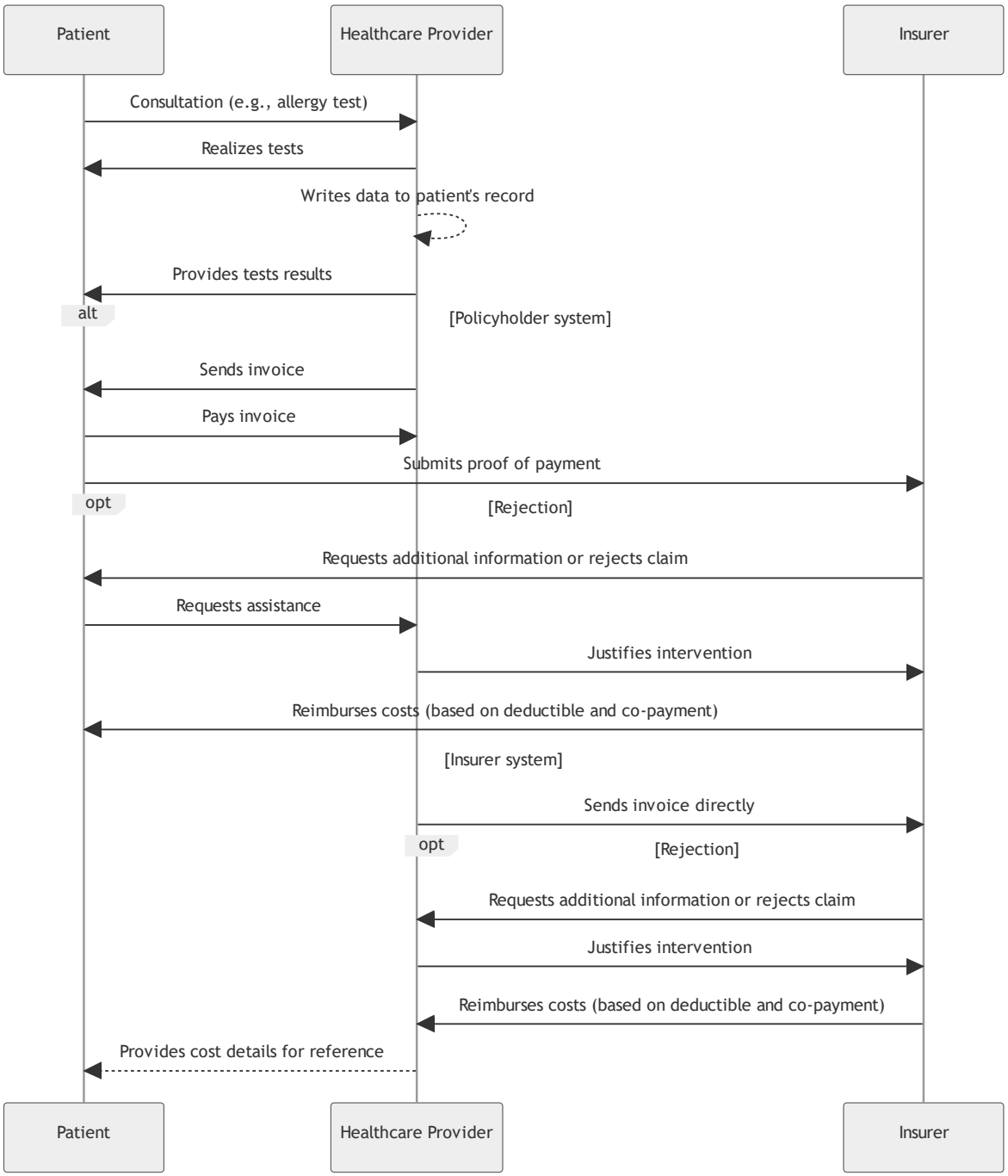


Figure 16: Typical interactions in the context of insurance claim.

Appendix II – Disability Insurance form 002.099 for professional readaptation (Canton Vaud, 2023)

Rapport médical somatique : Réadaptation professionnelle / Rente

Numéro d'assuré

Personne assurée :

Date de naissance :

Questions complémentaires

1 Informations générales

1.1

Le traitement ambulatoire / hospitalier que vous avez dispensé a eu lieu

du au

Date du dernier contrôle que vous avez effectué

Des contrôles ont été effectués précédemment par

Des contrôles ont été effectués à une date ultérieure par

1.2

A quelle fréquence voyez-vous le patient / la patiente actuellement ?

1.3

Quelle est, jusqu'à aujourd'hui, l'évolution de l'incapacité de travail attestée médicalement (en pour cent)?

%	de	à

Pour quelles activités avez-vous attesté une incapacité de travail?

Y a-t-il d'autres intervenants (médecins spécialistes, hôpitaux ou thérapeutes)?

Veuillez joindre les copies des rapports existants.

2 Situation médicale

2.1

Antécédents médicaux et évolution de la situation du patient/de la patiente

2.2

Situation et symptômes médicaux actuels

2.3

Médication actuelle (y compris le dosage)

2.4

Constats médicaux complets sur la base des examens que vous avez pratiqués

2.5

Diagnostics ayant une incidence sur la capacité de travail

(pour les affections psychiatriques, veuillez indiquer le code CIM-10 ou DSM-5.)

Diagnostics	**Depuis**

2.6

Diagnostics sans incidence sur la capacité de travail

Quand ces diagnostics ont-ils été posés?

Diagnostics	**Depuis**

2.7

Votre pronostic sur la capacité de travail du patient/de la patiente **sur un taux de 100%**

2.8

Prochaines mesures que vous envisagez / votre plan de traitement (thérapies, opérations chirurgicales, médication)

3 Situation professionnelle

3.1

Quelle est l'activité actuelle de votre patient / votre patiente?

☐ Je ne suis pas en mesure de répondre à cette question

3.2

Quelles sont les informations dont vous disposez sur la situation professionnelle de votre patient / votre patiente?

☐ Aucune information

3.3

A quelles exigences votre patient/patiente doit-il/elle faire face dans son activité professionnelle?
(par exemple: effort physique / activité alternée / répétitive / fonction de cadre / travail par rotation d'équipes / travail en équipe / contact avec les clients, ou toute autre particularité.)

☐ Je ne suis pas en mesure de répondre à cette question

Au cas où vous disposez d'une description du poste, veuillez en joindre une copie.

3.4

Existe-t-il des limitations fonctionnelles? Quels effets ont-elles sur l'activité que le patient a exercée jusqu'ici?
Veuillez décrire les limitations fonctionnelles.

☐ Je ne suis pas en mesure de répondre à cette question

3.5

Votre patient/votre patiente dispose-t-il/elle de ressources qui pourraient être utiles pour sa réinsertion?
(par exemple: connaissances linguistiques / formations continues / activités de la vie quotidienne / activités bénévoles / hobbys / temps libre / contacts avec des amis / voyages)

☐ Je ne suis pas en mesure de répondre à cette question

3.6

Avez-vous des doutes quant à sa capacité de conduire? Lesquels?

☐ Je ne suis pas en mesure de répondre à cette question

4 Potentiel de réadaptation

4.1

Combien d'heures de travail par jour peut-on raisonnablement attendre de votre patient/votre patiente dans l'activité qu'il/elle a exercée jusqu'ici?

☐ Je ne suis pas en mesure de répondre à cette question.

4.2

Combien d'heures de travail par jour peut-on raisonnablement attendre de votre patient/votre patiente dans une activité qui tienne compte de l'atteinte à sa santé ?

4.3

Votre pronostic sur le potentiel de réadaptation du patient/de la patiente. **Capacité de travail sur un taux de 100%** (même si le taux contractuel est inférieur).

4.4

Quels sont les facteurs qui font obstacle à une réadaptation?

4.5

Dans quelle mesure votre patient/votre patiente est-il/elle limité/e dans l'accomplissement des tâches ménagères?
(Par ex.: tenue du ménage /préparation des repas / nettoyage / achats / lessive / prise en charge des enfants)

☐ Je ne suis pas en mesure de répondre à cette question.

5 Divers

Quels autres éléments pourraient entrer en ligne de compte dans l'évaluation de la situation de votre patient ? Avez-vous d'autres informations à nous communiquer?

Date

Prénom, nom, adresse exacte (cabinet/service) et signature du médecin

Annexes

Numéro d'assuré:

Personne assurée :

Date de naissance :

Quels sont les travaux qui peuvent encore être exigés de la personne assurée, compte tenu des limitations dues à l'état de santé, dans le cadre d'une activité adaptée à son handicap?

Veuillez séparer en activités/durée/performance (rendement).

	oui	non	A raison de quelle durée est-ce exigible?		Avec quelle perfmance?
			temps complet oui/non	au cas où ce n'est pas exigible toute la journée, donner le nombre d'heures exigibles par jour	données en %
activités uniquement en position assise	☐	☐			
activités uniquement en position debout	☐	☐			
activités dans différentes positions	☐	☐			
activités exercées principalement en marchant (terrain irrégulier?)	☐	☐			
se pencher	☐	☐			
travailler avec les bras au-dessus de la tête	☐	☐			
accroupi	☐	☐			
A genoux	☐	☐			
rotation en position assise/en position debout	☐	☐			
soulever/porter (près/loin du corps?)	☐	☐			

monter sur une échelle/un échafaudage	☐	☐			
monter les escaliers	☐	☐			
autres? ……	☐	☐			
soulever/porter (près/loin du corps)	☐	☐	Limite de poids		
capacité de concentration	☐ non limitée		☐ limitée; genre:		
cap. de compréhension	☐ non limitée		☐ limitée; genre:		
capacité d'adaptation	☐ non limitée		☐ limitée; genre:		
résistance	☐ non limitée		☐ limitée; genre:		

Depuis quand ces indications sont elles valables?

Y a-t-il des points particuliers à respecter (par ex. une augmentation progressive de la capacité de travail, une place de travail calme)?

Y a-t-il besoin d'utiliser des moyens auxiliaires?

☐ oui ☐ non

Si oui, lesquels?

4. Signature

Prénom, nom, date et signature du médecin

Adresse exacte (cabinet/service)

Annexe relative à l'aptitude à la réadaptation

Numéro d'assuré :
Personne assurée :
Date de naissance :

Une mesure de réinsertion de l'AI n'est pas une mesure thérapeutique (ergothérapie, etc.), c'est une mesure visant à mobiliser une personne atteinte dans sa santé dans le but de reconstruire une capacité de travail : de 8 heures par semaine sans obligation de rendement au début, avec l'objectif d'augmenter progressivement jusqu'à obtenir une capacité de travail d'au moins 50%.

1. Existe-t-il des contre-indications médicales à suivre une telle mesure ?

☐ *Si oui, lesquelles ?*

.....

.....

☐ *Si non, depuis quand une telle mesure aurait-elle été possible ?*

.....

2. Quel est votre pronostic d'amélioration de la capacité de travail ?

☐ *Jusqu'à quel taux ? ……*

☐ *Dans quel délai ?*

.....

Date : ..... Timbre et signature :

Appendix III – Threat trees: Information disclosure

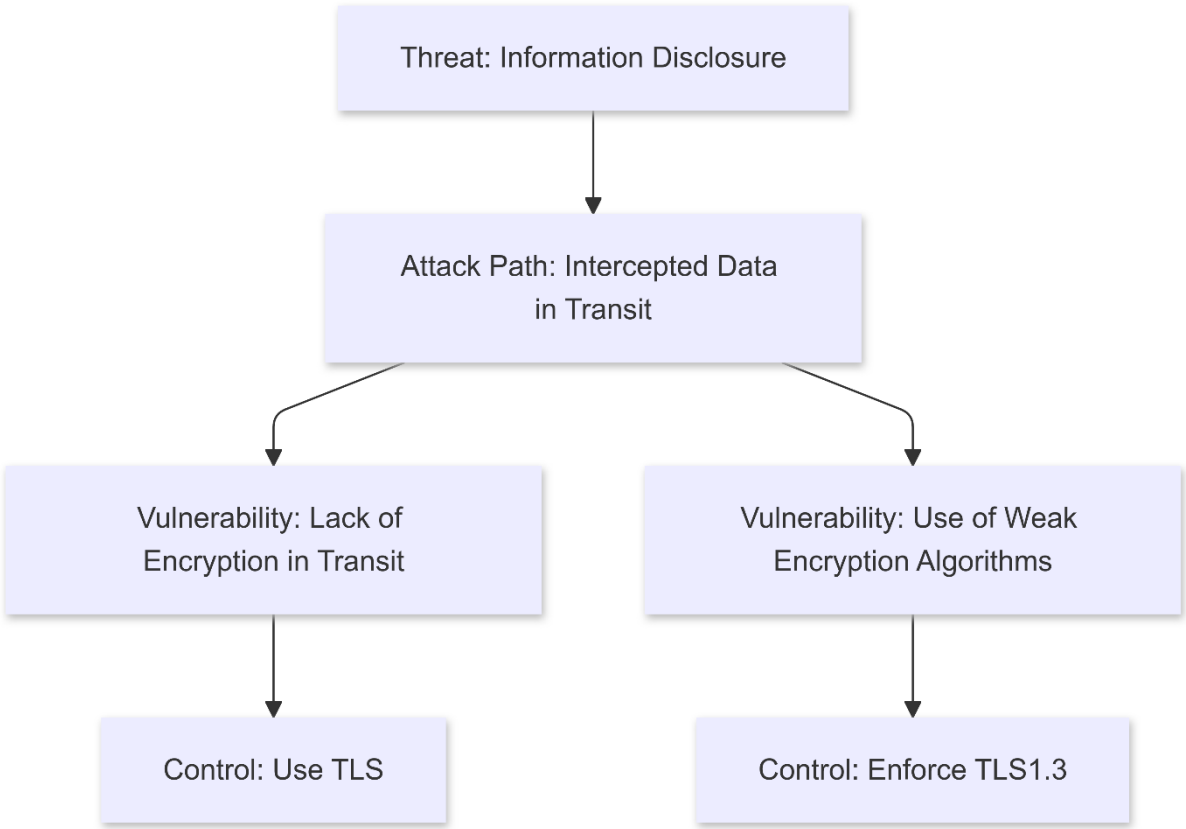


Figure 17: Threat tree – Data interception attack path.

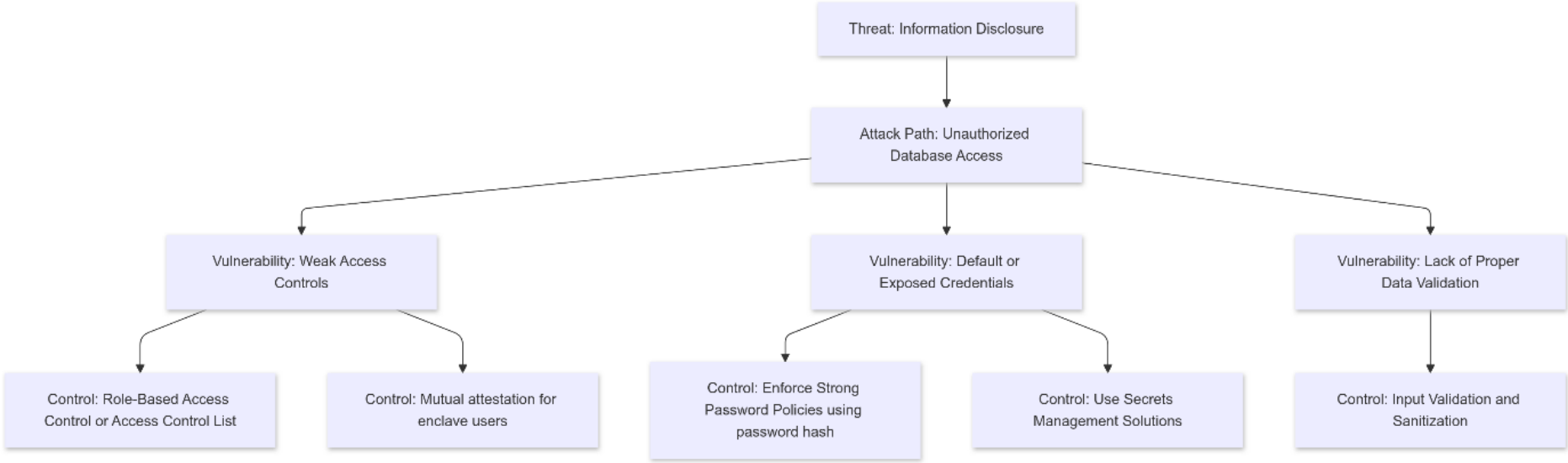


Figure 18: Threat tree – Unauthorized database access attack path.

.