Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare

Project Acronym: **ASCLEPIOS**
Project Contract Number: **826093**

Programme**: Health, demographic change and wellbeing**
Call: **Trusted digital solutions and Cybersecurity in Health and Care to protect privacy/data/infrastructures**
Call Identifier: **H2020-SC1-FA-DTS-2018-2020**

Focus Area: **Boosting the effectiveness of the Security Union**
Topic**: Toolkit for assessing and reducing cyber risks in hospitals and care centres**
Topic Identifier: **H2020-SC1-U-TDS-02-2018**

Funding Scheme: **Research and Innovation Action**

Start date of project: 01/12/2018 Duration: 36 months

Deliverable:

ASCLEPIOS Reference Architecture, Security and E-health Use Cases, and Acceptance Criteria

Due date of deliverable: 31/08/2019 Actual submission date: 29/08/2019

WP1: Reference Architecture and Use Cases

Dissemination Level: Public

Version: 2.6

1 Table of Contents

2 List of Figures and Tables

Figures

Tables

3 Status, Change History and Glossary

Status:	Name:	Date:	Signature:
Draft:	Antonis Michalas	15.07.2019	Antonis Michalas
Reviewed:	Nicolae Paladi	17.07.2019	Nicolae Paladi
Approved:	Tamas Kiss	29.08.2019	Tamas Kiss

Table 1: Status Change History

Version	Date	Pages	Author	Modification
v0.1	04.01.2019	6	A. Bakas	Begin working on section Cryptographic Primitives
v0.2	15.01.2019	7	A. Michalas	Correct format of the document and typos
v0.3	02.02.2019	18	A. Zalitko	Begin working on describing the API
v0.4	14.02.2019	20	A. Bakas	Finished working on Cryptographic Primitives
v0.5	21.02.2019	20	A. Michalas	Correct format of the document and typos
v0.6	30.02.2019	25	A. Bakas	Begin working on Overview of ASCLEPIOS architecture
v0.7	15.03.2019	30	A. Zalitko	Begin working on the classification of the requirements
v0.8	22.03.2019	31	A. Michalas	Correct format of the document and typos
v0.81	02.04.2019	35	K.Y. Yigzaw	Security and E-health Use Cases. ToC and methodology draft
v0.9	10.04.2019	37	A. Bakas	Begin working on the system model
v1.0	21.04.2019	45	A. Bakas	Begin working on ASCLEPIOS concepts
v1.1	22.04.2019	46	A. Zalitko	Begin working ASCLEPIOS Functions
v1.11	30.04.2019	50	A. Makhlysheva	Security and E-health Use Cases. Introduction
v1.12	02.05.2019	52	A. Makhlysheva	Security and E-health Use Cases. Methodology
v1.2	15.05.2019	54	A. Zalitko	Finalized a first draft of the system model
v1.3	17.05.2019	55	A. Michalas	Correct format of the document and typos
v1.4	28.05.2019	55	Y. Verginadis	Structural and content improvement suggestions
v1.41	31.05.2019	64	A. Makhlysheva	Security and E-health Use Cases. Updated version based on the comments and suggestions from others
v1.5	04.06.2019	69	A. Zalitko	Added a high-level description of the architecture
v1.6	06.06.2019	80	A. Bakas	Added a detailed description of the architecture
v1.7	08.06.2019	83	A. Michalas	Review and re-wrote several parts on Section 7
v1.71	12.06.2019	86	M. Tuler, S. Olabarriaga, M. Khvastova, D. Krefting	Security and E-health Use Cases. Demonstrators 1 is ready

v1.72	13.06.2019	93	A. Makhlysheva, J.G. Bellika, M. Tuler, D. Krefting	Security and E-health Use Cases. Demonstrator 3, Demonstrator 2, changes in Demonstrator 1
v1.73	14.06.2019	94	A. Makhlysheva	Security and E-health Use Cases. Requirements prioritized list is ready
v1.74	15.06.2019	95	K.Y. Yigzaw, A. Makhlysheva	Security and E-health Use Cases. Demonstrator 3 is ready
v1.75	18.06.2019	88	A. Makhlysheva	Security and E-health Use Cases. Overall structure, references and formatting
v1.8	27.06.2019	98	A. Bakas	Changes on the description of the architecture
v1.9	03.07.2019	102	Y. Verginadis	Input on the Policy Enforcement Layer
v2.0	03.07.2019	107	A. Makhlysheva	Correct format of the document and typos
v2.0	04.07.2019	108	A. Zalitko	Integration of Security and E-health Use Cases and Acceptence Criteria into ASCLEPIOS architecture
v2.0	04.07.2019	110	A. Michalas	Review and changes to the entire document
v2.1	09.07.2019	112	E. Biliri	Description of the Security Administrative Data Analytics components
v2.1	09.07.2019	112	A. Michalas	Wrote Introduction and Conclusions
v2.2	14.07.2019	109	S. Olabarriaga	Correct format of the document and typos
v2.3	18.07.2019	109	A. Bakas	Address review comments in Sections 11.2 and 10
v2.4	14.07.2019	109	A. Zalitko	Address review comments in Section 11.1
v2.5	30.07.2019	109	A. Makhlysheva	After-review changes in Security and E-health Use Cases and Acceptance Criteria
v2.5	01.08.2019	109	A. Michalas	Last set of changes based on the review
v2.6	21.08.2019	111	A. Makhlysheva	Change history including Security and E-health Use Cases and Acceptance Criteria, formatting issues

Table 2: Deliverable Change History

Glossary

SKE	Symmetric Key Encryption
PKE	Public Key Encryption
σ	Signature
$\mathcal{C}$	Challenger
$\mathcal{A}$	Adversary
$\mathcal{M}$	Message Space
$\mathcal{K}$	Key Space
H	Hash Function
PRG	Pseudorandom Generator
PRNG	Pseudorandom Number Generator
CP-ABE	Ciphertext Policy Attribute-Based Encryption
ABAC	Attribute-Based Access Control
XACML	eXtensible Access Control Markup Language
SSE	Symmetric Searchable Encryption
FE	Functional Encryption
CPA	Chosen Plaintext Attack
CCA	Chosen Ciphertext Attack
CKA	Chosen Keyword Attack
CSP	Cloud Service Provider
RA	Registration Authority
MS	Master Authority
KeyTray	Key Tray

REV	Revocation Authority
u_i	User
TEE	Trusted Execution Environment
EHR	Electronic Health Record
EMR	Electronic Medical Record
GDPR	General Data Protection Regulation
GP	General Practitioner
HPC	High Performance Computing
HST	Home Sleep Testing
NSE	Norwegian Centre for E-health Research
SMPC	Secure Multi-Party Computation
PG	Polygraphy
PPDDM	Privacy-Preserving Distributed Data Mining
UC	Use Case

Table 3: Glossary

4 Introduction

Until recently, e-health was seen as expenditure rather than an investment. Within the past decade this has changed so drastically that e-health has moved to the top of the development agenda for both private organizations and public administration bodies. We have seen a steady increase in research focus and funding, with the aim to modernize existing healthcare systems (37) and to provide reliable and cost-effective e-health services. We are in front of a major technological upturn of the healthcare industry: a leap from relying on handwritten records to using artificial intelligence approaches, analyzing large volumes of patient data, and identifying individual treatments.

During the early stages of its development, the core idea of the e-health concept was to modernize existing medical systems by digitizing the personal health records. The transition from handwritten to digital records revealed both the importance and benefits of the concept and made a great impact on the healthcare industry. As a consequence, researchers focused on further developing the area by realizing healthcare services that went beyond data digitization. Scientists started envisioning e-health systems that enabled patients to remotely access their personal records and digitally share their medical summary with healthcare professionals (36). Furthermore, e-health researchers leverage emerging mobile and wearable devices, such as smartphones and fitness trackers. Such devices allow patients to collect fine-grained measurements of vital signs (body temperature, pulse rate, respiration rate, blood pressure), and subsequently obtain behavioural and health insights.

Even though the e-health industry started leveraging a number of new technologies in general, it has been slow in adopting IT and new, emerging technologies such as the cloud. The main reason for this is mainly due to the fear of storing sensitive data online. Lacking effective security mechanisms to protect user data from unauthorized access, sensitive patient data may leak to unauthorized third parties, such as insurance companies or potential employers. Protecting the privacy of patient data is a core challenge in the e-health domain. When it comes to protecting sensitive data, different organizations have different needs. Each use case brings its own level of risk and corresponding risk reduction once it is addressed. However, and despite the different needs of each use case, the main challenge always remains the same – preventing unauthorized access to the data that are stored in a remote location. Furthermore, it has been observed (38) that failing to provide users of e-health services with proper cybersecurity and privacy-preserving mechanisms can slow down the overall adoption of e-health.

Despite the relatively slow adoption of IT in the healthcare industry, the medical community gradually advances towards wider adoption of electronic healthcare. New computing technologies, such as cloud computing, fit squarely into this evolution. Cloud based e-health services could bring significant benefits (35). However, implementing properly a secure and robust healthcare digital infrastructure requires the design and implementation of several mechanisms that will not only ensure the privacy and protection of private data but will also provide certain guarantees about the trustworthiness of the cloud-based digital services used by patients and healthcare practitioners (32).

The main goal of ASCLEPIOS is to design and develop an e-health framework that will allow patinets to store and share their medical records in a secure and privacy-preserving way while at the same time they will be able to receive certain guarantees about the trusted state of the overall framework. In addition to that, both patients and doctors will be able to perform analytics in a privarcy-preserving way. We hope that this will not only allow patients to be sure about the privacy of their data but it will also allow healthcare practioners to better analyze and undersrtand the behaviour of certain diseases.

To achieve this, ASCLEPIOS needs to be based on a concrete, modern and flexible architecture that will not only utilize several modern techniques from the field of cryptography, machince learning and analytics but it will also support the raise of security awarness – something that is currently missing from the helathcare sector.

4.1 Scope of the Deliverable

The scope of *D.1.2: ASCLEPIOS Reference Architecture, Security and E-health Use Cases, and Acceptance Criteria* is to provide a high-level but detailed description of ASCLEPIOS's Reference Architecture. To this end, in this deliverable the overall architecture of ASCLEPIOS along with its main components, mechanisms, algorithms and models, the interconnection scheme and the specific interfaces for exchanging information among them will be designed and described. Furthermore, an analysis of the project's use cases wil be described along with the implementation scenarios of the mechanisms that will be developed within the project. Description of both architecture and use cases will be coupled with the security and health requirements that have been collected and described in *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements* (39). Use cases, as well as a suitable set of acceptance criteria for the validation of the mechanisms developed within ASCLEPIOS, will be developed in WP6.

5 Security and E-health Use Cases and Acceptance Criteria

5.2 Introduction

The first part of the document aims to identify and describe the use cases (i.e. the implementation scenarios of the mechanisms) for the ASCLEPIOS project. The use cases will define the data integrity, confidentiality, secure software execution, hardware security and secure and privacy data sharing requirements for the three demonstrators of the project that are hosted in public or private cloud computing infrastructures.

Providing a description and detailed specification of the use cases at an early stage allows all partners to get a clear understanding of the requirements for the ASCLEPIOS project. Additionally, the definition of suitable acceptance criteria per component/mechanism will be realized. This will be further used for the definition of the demonstrators' strategy for validation and evaluation of the deployed mechanisms in the pilot testbeds of the project.

The project has the following demonstrating applications developed by the project partners from the healthcare sector:

- *Data sharing for improved treatment in stroke acute care*,
- *Collaboration and analysis platform for inpatient and outpatient sleep medicine,* and
- *Privacy-preserving monitoring and benchmarking of antibiotics prescriptions*.

Further, we describe the three named above intended scenarios and demonstrators, as well as how the ASCLEPIOS framework will be applied and showcased by these implementations.

Demonstrator 1: Data sharing for improved treatment in stroke acute care

Stroke is a condition where poor blood flow in the brain results in cell death. This can lead to a part of the brain not functioning properly, with signs and symptoms appearing soon after the stroke has occurred. Time is critical in acute stroke care: within a very small-time frame, health professionals need to identify the type of stroke and severity, decide upon the treatment, transport the patient to the adequate care center, and perform the required intervention. The treatment generates and requires a large amount of data that needs to be shared between the health professionals along the whole process. Such data also represent valuable sources of evidence for medical research. Sharing of sensitive data in general raises privacy and safety issues, which can be addressed with the solutions proposed by ASCLEPIOS.

Demonstrator 2: Collaboration and analysis platform for inpatient and outpatient sleep medicine

Sleep medicine relies on the measurement of overnight multidimensional biosignals and audiovisual recordings. For inpatients, a diagnostic procedure is called polysomnography, which consists of several biosignal recordings typically performed in a certified sleep lab within a hospital. For obstructive sleep apnea, when a patient stops breathing due to an obstruction of the upper airways, the default diagnosis procedure is an unattended home sleep testing. After instructions by the technical staff of the hospital, the patient attaches the sensors to him-/herself at home and starts the recording device. The data are stored within the device and are returned by the patient the next day. In both cases, data transfer and remote execution of analysis methods are not employed for patient data in sleep medicine in the context of daily care, neither for inpatient, nor for outpatient scenarios. Solutions for storing and processing sleep-related data online are limited due to the inherent data privacy and currently used only for research.

Demonstrator 3: Privacy-preserving monitoring and benchmarking of antibiotics prescription

The increasing emergence of medication-resistant bacteria observed worldwide is lowering the success rates in infection treatment using antibiotics. Appropriate use of antibiotics by GPs can make a significant difference in the overall national consumption of these drugs. Studies have shown that feedback showing a GP's prescription pattern in comparison with his/her peers leads to behavioral improvements. However, privacy concerns and regulations limit access to health data for secondary uses, including antibiotics prescription monitoring and benchmarking. With local computations on the encrypted data, the security and privacy risks in case of an adversary compromises the server are minimized. Stronger security and privacy can increase GPs' and patients' willingness to participate in such quality improvement initiatives as reduction of unnecessary antibiotics prescription.

5.3 Methodology

This section describes the methodology for identifying and defining the project use cases and acceptance criteria for each demonstrator.

The healthcare partners analyzed their e-health applications that are demonstrators for the project and described the most relevant use cases for each of the demonstrators. The use cases have been developed by the project partners responsible for each demonstrator.

A use case illustrates how actors can use a system to meet a particular goal, showing the appropriate paths they might take to get there, as well as those situations that would cause them to fail.

The actors involved in the demonstrators were formally defined in Table 4 to create a common understanding throughout the document. An actor is a device, an application or a person. An actor can be *primary* or *secondary*. Primary actors act on the system, initiate interactions with the system, and use the system to fulfill their goals. A secondary actor is acted on/invoked/used by the system and helps the system to fulfills its goal.

Actor	Description	Demonstrator	Actor type
Stroke Patient	Unconscious patient in an acute stroke situation. Conscious patient after discharge	Demonstrator 1	Secondary/Primary
Third person	Someone who calls on the behalf of the unconscious patient	Demonstrator 1	Secondary
Emergency call center Technicians	Professionals who receive the emergency call, begin the triage process and contact the ambulance team	Demonstrator 1	Primary
Ambulance paramedics and nurses	Professionals who give first aid, continue the triage process and transport the patient to the primary or comprehensive stroke hospital.	Demonstrator 1	Primary
Researchers	Who reuse collected data to do research on acute stroke care	Demonstrator 1	Primary

Doctors and nurses	Professionals from various specialies (neurologist, interventional neuroradiologist, anesthesiologist) who treat the patient at the hospital	Demonstrator 1	Primary
Inpatient physicians	Physician in charge to decide on the diagnosis and therapy based on the PSG	Demonstrator 2	Primary
Outpatient physicians	Physician in charge to decide on the diagnosis and therapy based on the HST	Demonstrator 2	Primary
Third party physicians	Colleagues that are asked for consulting on a certain case	Demonstrator 2	Primary
In-house researchers	Researchers from within the clinics reusing the data for sleep research	Demonstrator 2	Primary
Third party researchers	Researchers from other institutions that want to use the sleep data	Demonstrator 2	Secondary
Outpatient medical staff (nurse)	Staff giving out the home sleep testing devices and monitor the measure	Demonstrator 2	Primary
Sleep Disorder Patient	Person with suspected or diagnosed sleep disorder	Demonstrator 2	Primary/secondary
PSG device	A device to measure overnight biosignals in inpatient scenario	Demonstrator 2	Secondary
HST device	A device to measure overnight biosignals in outpatient scenario	Demonstrator 2	Secondary
Video	A device to monitor visually the patient overnight (inpatient and outpatient)	Demonstrator 2	Secondary
Artefact detector	Application detecting failures in the overnight measurement	Demonstrator 2	Secondary
Sleep event detector	Application detecting events during sleep based on PSG/HST and/or video signal	Demonstrator 2	Secondary
Second opinion app	Application enabling interactive remote visualization and discussion on a case	Demonstrator 2	Secondary
Patient communication app	Application enabling direct communication between outpatient staff and patient at home	Demonstrator 2	Secondary

Biosignal storage app	Application that stores proprietary device signals as standardized interoperable biosignal recordings	Demonstrator 2	Secondary
Clinician	GP who prescribes antibiotics for his/her patients' treatment	Demonstrator 3	Primary
System administrator	A person who configures the mission scheduler with the necessary computation tasks	Demonstrator 3	Secondary
Mission scheduler	A process that initiates indicator agent and benchmark agents to compute indicators	Demonstrator 3	Secondary
Indicator agent	A process initiated by the mission scheduler to compute the group level quality indicators for antibiotic prescriptions	Demonstrator 3	Secondary
Benchmark agent	A process initiated by the mission scheduler and computes an individual level quality indicators, prepares feedback report for antibiotic prescriptions and sends a notification about it to the clinician	Demonstrator 3	Secondary

Table 4: Actors involved in the demonstrators

The demonstrator description contains the following sections:

- *Introduction*,
- *Background* with a motivation for a demonstrator and a description of the current situation (state-of-the-art) used to build the use cases, as well as the envisioned situation,
- *Use cases*, describing all use cases developed in the demonstrator and including UML use case diagram,
- *Components/mechanisms of ASCLEPIOS involved in demonstrator,*
- *Security and privacy requirements* summary for the demonstrator (all use cases),
- *Data requirements* summary for the demonstrator (all use cases), such as clinical information structure, data storage, data retrieval, data rectification, etc; and
- *Testbed,* section describing the planned infrastructure on which the demonstrator will be deployed.

Security and privacy requirements for each demonstrator, such as data integrity, confidentiality, secure execution, hardware security, and data sharing, refer to the requirements specified in deliverable *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements.*

The demonstrator template is available in Annex I. Demonstrator template.

To show an overview of each demonstrator, a use case diagram is used (Figure 1). A use case diagram models the functionality of a system using actors and use cases (1).

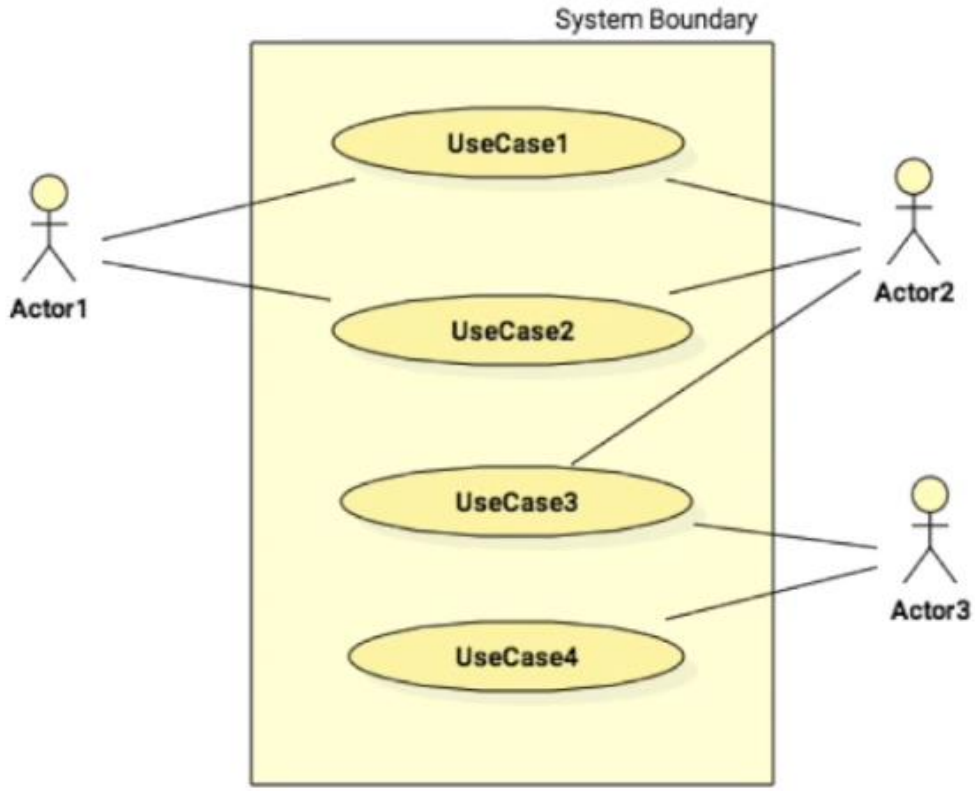

Figure 1: Demonstrator UML use case diagram

For description of use cases, we created a template - a table, summarizing all the necessary information to define the use case (Table B in Annex I. Demonstrator template). The table is inspired by the use case template used by the eDREAM project (2). The table contains actors involved in the use case, main and an alternate flow, failure scenarios, and acceptance criteria. Acceptance criteria will be defined for each use case to serve as a basis for testing the conformance of the implementation of the ASCLEPIOS components to the requirements. Criteria are likely to be different for each use case of a demonstrator. There can be similar acceptance criteria across multiple demonstrators. The acceptance criteria will be further elaborated in *WP6: Demonstrators and Performance Evaluation*.

6 Demonstrator 1: Data Sharing for Improved Treatment in Stroke Acute Care

6.4 Introduction

The availability of EMR during an emergency situation is of paramount importance. It allows healthcare professionals to access patient's data on time and properly plan the next steps to be taken during treatment. After the emergency phase, the EMRs contain valuable information for research and healthcare quality improvement. The technologies and framework developed by the ASCLEPIOS project have the potential to provide solutions to the problem of data unavailability during an emergency situation. Additionally, the ASCLEPIOS solutions can also facilitate further exploitation of available data for research in a GDPR-compliant manner. This demonstrator will illustrate how to share encrypted patient's data during an emergency situation, ensuring a treatment team to have only access to the patient's data for the time needed to complete a specific process related to the patient's treatment (e.g. transfer patient to the hospital). The demonstrator will also exploit access control and privacy-preserving techniques to perform analytics on existing EMR data for research on stroke care. The demonstrator will target the stroke acute care process ongoing in the Netherlands.

6.5 Background

In a stroke onset, patient, or a third person on behalf of the patient, is the first to contact the emergency call center. During the telephone call, the call center professionals, being trained healthcare workers, follow a triage system where a suspected stroke is determined. They send a message to the ambulance. The message is shown on a display in the ambulance and contains the patient's location and additional information collected during the phone call. Further, the ambulance is sent from one of the regional centers, with the goal to arrive within 15 minutes.

On the way to the destination hospital, the paramedics communicate with it by radio and phone, to have the hospital prepare an emergency room with the relevant experts to receive the patient.

Each entity involved in stroke treatment has its own data storage that is often not associated with other entities. This hampers the ensemble of a single electronic medical record per patient. Sufficient data might, therefore, be unavailable to support treatment and in any decision-making process by the various teams involved in emergency care.

The EMR of stroke patients constitutes a valuable source for research. Medical researchers constantly analyze the process of acute stroke treatment seeking for opportunities to reduce treatment time or improve outcome. Since patient's EMRs are spread across various organizations, obtaining access to the EMR data is very difficult. In many situations it is not allowed according to GDPR guidelines due to lack of consent from patients. Therefore, much data remains unexploited for research.

6.5.1 Motivation

The use of EMR data improves the overall quality of care received by patient: it can lead to a substantial reduction of unnecessary investigations and improvement of communication between healthcare professionals involved in the treatment. Availability of EMR, especially when a patient is under an emergency situation, is critical. For example, in stroke treatment, the phrase "*Time is brain"* conveys the idea that minutes can make the difference between life and death (3). Guarantee that the patient's EMR will be available to healthcare professionals

involved in acute stroke treatment can save time and improve the efficiency of decision-making processes, leading to quality of care for such patients.

A stroke patient stays hospitalized for a few days after treatment due to potential risk of complications. For example, delayed cerebral ischemia (DCI) is one of the most severe complications in patients who have had hemorrhagic stroke caused by subarachnoid hemorrhage. Currently it is very difficult to predict DCI. Therefore, risk assessment takes a conservative approach, potentially extending hospitalization longer than necessary. A variety of machine-learning methods can be used to develop DCI prediction models combining clinical and imaging data. The goal is to discharge sooner the patients with low DCI risk. It has been challenging to obtain large amounts of data to learn the models. Not only high-quality data are scarce, but also data controllers are reluctant to share. Moreover, organizing the computing infrastructure to learn the models requires HPC resources when imaging data are involved. Although the data are anonymized, there are still restrictions about transferring the data to infrastructures outside the hospital intranet.

6.5.2 State-of-the-art

Emergency care for acute stroke treatment involves professionals at the emergency call center, ambulance service, and primary and comprehensive stroke hospitals. All of them need to share information on a so-called “break glass” access mechanism. Break glass access (which draws its name from breaking the glass to pull a fire alarm) refers to a quick means for a person who does not have access privileges to certain information to gain access when necessary (4). Below we describe a typical scenario, providing high-level information of the involved parties as well as the basic information exchange between them.

When a patient suffers a stroke, he/she or anyone present with the patient, is the first to contact the emergency call center by phone. The call center team is composed of trained healthcare workers who are able to determine if there is an emergency situation and how to address it best. During the phone call, the call center professional follows a triage process, collecting information about the time of stroke onset, personal data (e.g. age, gender etc.) and some impressions about the patient conditions (e.g. speech capabilities).

When there is a suspicion of stroke, the call center professional contacts the ambulance service and shares the collected information about the patient. An ambulance closest to the event location is sent from the regional center, and, as soon as the ambulance arrives, the ambulance team continues triage process. They perform examinations, measurements and medical procedures at the patient’s location and during the travel to the hospital. When they arrive to the hospital, all the information about patient’s condition is orally shared with the hospital team.

During the travel, the ambulance professional communicates by phone with the proposed destination hospital. The hospital prepares the emergency room and the team with relevant experts to receive the patient (e.g. neurologist, interventional neuroradiologist, anesthesiologist, nurses). If the patient has already a record in this hospital’s local system, the hospital team can access the patient’s medical record. If there is no information about the patient, the hospital team may attempt to contact other hospitals to retrieve the patient’s medical record. Furthermore, the hospital team collects additional data about the patient, which is stored at the patient’s EMR at the treating hospital. In case of a patient with a large vessel occlusion eligible for additional endovascular treatment, the patient needs to be transported by a second ambulance to the comprehensive stroke hospital. Therefore, it is necessary for the teams in the second ambulance and second hospital to access and update the patient’s medical record.

All the collected data are potentially useful to perform analytics for research and cate quality improvement. Currently such data are shared in a conservative manner with researchers using ad-hoc solutions organized for the case at hand, and which do not necessarily comply with GDPR requirements. For example, a *"stroke dashboard"* (5) has been developed to enable visualization of timings during acute stroke treatment. The goal is to reveal bottlenecks. The data are extracted from the AMC EMR system, and exported by visualization on a stand-alone desktop application. In that solution no measure was adopted to secure the exported data.

6.5.3 Envisioned situation

The ASCLEPIOS project will develop solutions to facilitate usage of cloud infrastructure in a secure way, which will enable trusted data sharing. A cloud-based EMR offers possibilities to enhance patient information sharing during the acute phase of a patient's journey from the place of stroke onset to the care center. The AMC will apply the new encryption techniques developed in the project to design, implement and evaluate new approaches for flexible, dynamic and fine-grained data access authorization during the treatment of stroke and in research thereafter. The demonstrators built in ASCLEPIOS will explore and evaluate two cases: use of modern attribute-based encryption for dynamically grating authorization and revoking access to patient information during the acute phase (break glass situation), and privacy-preserving machine-learning scheme for large data sharing to use in predictive models.

Increasing information availability in stroke acute care can help healthcare professionals achieve faster and more effective treatment. Moreover, through the flexible access control and new privacy-preserving techniques developed in ASCLEPIOS, sharing data for research will be implemented in a GDPR compliant manner and become more appealing. Finally, the techniques for device attestation developed in ASCLEPIOS will enable usage of mobile devices in a trusted manner, facilitating mobility that is necessary during emergency care.

6.6 Use cases

Table 5 is a summary table for the use cases developed for the demonstrator.

ID	Name
D1-UC1	Store data
D1-UC2	Start emergency session
D1-UC3	Join emergency session
D1-UC4	Retrieve data from EMR
D1-UC5	Add data to EMR
D1-UC6	Leave emergency session
D1-UC7	Close emergency session
D1-UC8	Request consent for research
D1-UC9	Analytics for research

Table 5: Demonstrator 1 use cases

Figure 2 shows the use case diagram for the demonstrator.

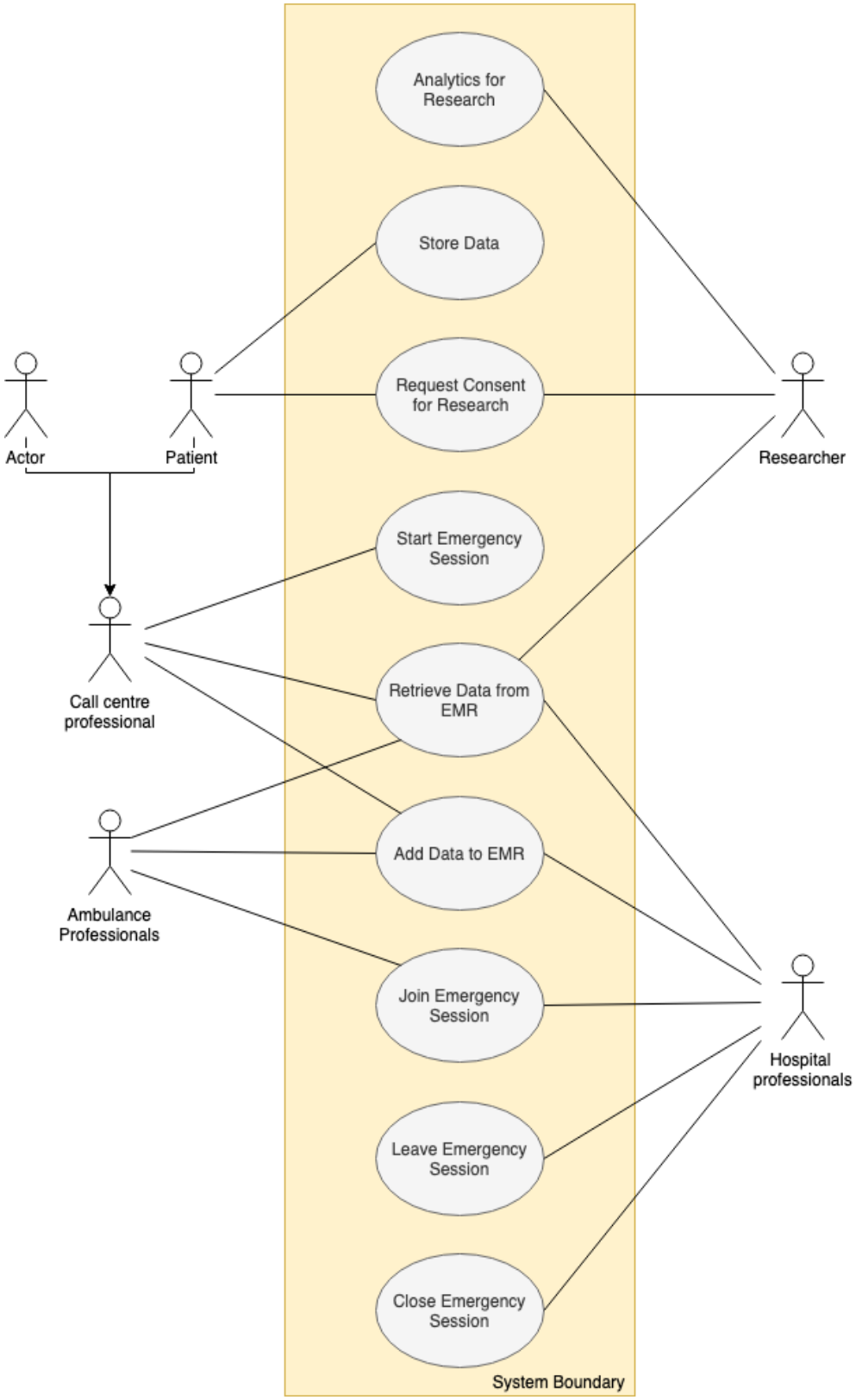

Figure 2: Demonstrator 1 use case diagram

In all the following use case descriptions, by “trusted” patient or “trusted” healthcare or call center professional, we mean that the patient/professional has been properly authenticated with the system, and that this person is entitled to legitimate use of the system.

6.6.1 Store data

The patient encrypts her EMR locally using a policy that enables access in emergency case and for research. The resulting cipher text is stored in the CSP. Table 6 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC1
Use case goal	Make patient's EMR available for emergency situations and research.
Assumptions & pre-conditions	• Patient has been registered to the system • Patient has generated the key to encrypt the EMR • Policies are defined • Device (computer) used by patient is secure • Patient is trusted
Use case initiation	The patient has his/her EMR and wants to store it in the CSP
Use case main scenario	Patient wants to share his/her EMR and has it available for further access by others. The patient is responsible for making his/her data available
Use case alternate scenario	None
Use case failure scenario	• No connectivity • No space to store data
Acceptance criteria	Encrypted EMR stored in CSP

Table 6: "Store data" use case description summary

6.6.2 Start emergency session

Through this use case, the system acknowledges the patient emergency event and begins the emergency session. This takes place when a patient, or someone on his/her behalf, contacts the call center team by phone. The call center professional received the call identifies the patient and requests access to patient's EMR. Table 7 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC2
Use case goal	Start an emergency session where authorized professionals are granted break glass access to patient's data
Assumptions & pre-conditions	• All users have been registered to the system • (Part of the) Patient data have been enabled for break glass access • Call center employee is trusted

	• Device used by call center employee is trusted
Use case initiation	Emergency call center professional answered the phone call from the patient creates a new emergency session to access data about this patient
Use case main scenario	Patient has a stroke event and contacts the emergency call center asking for treatment
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Professional is not authorized • Patient's EMR is not found • No break glass for patient data
Acceptance criteria	The emergency session is successfully initiated and the call center employee is granted break glass access to the patient's EMR

Table 7: "Start emergency session" use case description summary

6.6.3 Join emergency session

In this use case, the system includes users from a treatment team to an existing emergency session. This is initiated by some user who is already part of the emergency session. At first, the call center includes the ambulance team to join the session. Later, the ambulance team can invite the hospital team, etc. Table 8 shows the description of the use case.

Attribute	**Description**
Use case ID	D1-UC3
Use case goal	Include a treatment team to an emergency session where authorized professionals are granted break glass access to a patient's data.
Assumptions & pre-conditions	• All users have been registered to the system. • (Part of the) patient data have been enabled for break glass access • The user who starts this use case already needs to be part of the emergency session
Use case initiation	A professional from a treatment team that is already involved in the emergency session contacts the system to include next treatment team to join the emergency session.
Use case main scenario	This use case is valid in three scenarios: • Call center requests an ambulance team to transport the patient to a hospital • Ambulance team requests a hospital to receive the patient • Hospital team request an ambulance to transport the patient to another hospital

Use case alternate scenario	None
Use case failure scenario	• No connectivity • Professional is not authorized • Invalid or expired emergency session
Acceptance criteria	Users in the new team are successfully included in the emergency session and provided the key to access the patient's EMR

Table 8: "Join emergency session" use case description summary

6.6.4 Retrieve data from EMR

Once the treatment team is part of the emergency session, the members are granted access to retrieve from the CSP the cipher text containing the EMR of the patient under emergency treatment. The professionals have access to the key to decrypt the EMR only through a secure read-only application. Table 9 shows the description of the use case.

Attribute	**Description**
Use case ID	D1-UC4
Use case goal	Grant access to read a patient's EMR data
Assumptions & pre-conditions	• (Part of the) patient data have been enabled for break glass access • All the professionals of the team are part of the emergency session • The professionals were given the keys to decrypt the EMR • The application allows read-only access to the data
Use case initiation	The professional who has already the key to decrypt the EMR contacts the cloud service provider to retrieve the cipher texts
Use case main scenario	Professional retrieves the EMR before start to treat the patient
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Professional is not a member of emergency session • Patient's EMR is not found
Acceptance criteria	Cipher texts successfully retrieved from CSP and decrypted by user

Table 9: "Retrieve data from EMR" use case description summary

6.6.5 Add data to EMR

During and after a patient's treatment, all teams may upload new files to the patient EMR. These files include the report of the emergency treatment and needs to be encrypted before storage in the cloud. Table 10 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC5
Use case goal	Add new data to the patient EMR by uploading a new cipher text to the cloud storage.
Assumptions & pre-conditions	• Only the professional who treated the patient can upload new data to the patient's EMR • The professional is part of a treatment team involved in the emergency session and has key to encrypt data • The professional is authorized to write on the patient's EMR
Use case initiation	After an examination or medical procedure, the treatment records need to be stored in the cloud
Use case main scenario	Each treatment team needs to add new data to the EMR after handling the patient
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Professional is not authorized • Patient's EMR is not found • No space to store data
Acceptance criteria	New cipher texts successfully stored on CSP

Table 10: "Add data to EMR" use case description summary

6.6.6 Leave emergency session

Break glass access needs to be revoked when it is no longer necessary for patient treatment. This happens when a treatment team leaves the emergency session. The moment when the patient leaves the emergency care of the hospital defines the end of involvement of the teams of emergency care unit. The system automatically revokes access to the previous treatment teams according to specific criteria. Table 11 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC6
Use case goal	Revoke access that was granted to a treatment team in an emergency session after access is no longer necessary.
Assumptions & pre-conditions	• Patient is not under treatment team care • Treatment team already added new data to the patient's EMR
Use case initiation	The treatment team at the hospital informs the system that the previous teams need to leave the emergency session and no longer have access to the patient's EMR.
Use case main scenario	When the patient arrives to the first hospital, the call center team and the ambulance teams leave the emergency session

	If the patient needs to be transferred to a second hospital, the treatment team in the first hospital will leave the emergency session as soon as the patient arrives at the second hospital
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Patient's EMR is not found • Professional is not authorized
Acceptance criteria	All members in the team no longer have access to the patient's EMR

Table 11: "Leave emergency session" use case description summary

6.6.7 Close emergency situation

The emergency session ends when all professionals associated with it have been revoked explicit. After this, no new team is allowed to join the session anymore. Table 12 shows the description of the use case.

Attribute	**Description**
Use case ID	D1-UC7
Use case goal	End an emergency session.
Assumptions & pre-conditions	• All users have been removed from the emergency session • Patient is no longer under emergency treatment • Hospital has added data to the patient's EMR concerning the emergency treatment
Use case initiation	The hospital treatment team informs the system that the patient has been discharged from the emergency unit, i.e., the emergency treatment has ended
Use case main scenario	All treatment teams that were involved in the treatment have been removed from the emergency session
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Professional is not authorized • Patient's EMR is not found
Acceptance criteria	Emergency session successfully ended

Table 12: "Close emergency session" use case description summary

6.6.8 Request consent for research

When the patient is conscious, he/she is able to consent with the use of his/her medical records for research. Table 13 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC8
Use case goal	Researcher obtains consent from the patient to use his/her data in research
Assumptions & pre-conditions	• Patient needs to be aware about the research and decides if he/she wants or not to participate • Patient already has EMR stored in the system • Researcher is registered in the system • Patient is using a trusted device
Use case initiation	A researcher sends a request to the patient to access his/her data
Use case main scenario	A researcher wishes to process the patient's EMR for research purposes and requests the patient to grant access to his/her data
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Researcher is not authorized for this request • Request is not valid
Acceptance criteria	The researcher receives an answer from the patient: access to process the patient's EMR is either granted or denied

Table 13: "Request consent for research" use case description summary

6.6.9 Analytics for research

After obtaining consent from the patient, the researcher processes a collection of EMRs to carry out research. Analytics can be done using privacy-preserving functions or by decrypting the anonymized EMRs. Table 14 shows the description of the use case.

Attribute	Description
Use case ID	D1-UC9
Use case goal	Researchers perform analytics on patients' EMR data
Assumptions & pre-conditions	• All users have been registered to the system • Researcher has been granted access to the data by the data subjects
Use case initiation	Researcher starts data processing
Use case main scenario	After obtaining a consent from the patients, the researcher processes a collection of EMRs to carry out research. Analytics can be done using privacy-preserving functions or by decrypting the anonymized EMRs
Use case alternate scenario	None
Use case failure scenario	No connectivity

Acceptance criteria	The researcher performed analytics on patient's EMR data

Table 14: "Analytics for research" use case description summary

6.7 Components/mechanisms of ASCLEPIOS framework involved in the demonstrator

The following components/mechanisms of the ASCLEPIOS are involved in the demonstrator:

- *Use of Cloud/HPC resources* to store the EMR's and enable their processing for analytics,
- *Data sharing and revocation using SSE and ABE* to control access to the EMR during the emergency situation and afterwards for research,
- *Privacy-Preserving analytics using FE* to enable reuse of EMR data for research,
- *Medical device hardware integrity* to enable trust on mobile devices used for data access during an emergency situation,
- *Cloud provider integrity* to store EMR data in a trusted manner,
- *Increase GDPR and Security Awareness* by enabling explicit consent from patients to reuse their EMR.

6.8 Demonstrator security requirements

The following security and privacy requirements (specified in *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements*) are applied to use cases of the demonstrator. The requirement applicable to all use cases refer to generic properties of the system that will hold the EMR data (see Table 15).

Use case	Requirement	ASCLEPIOS functionalities
all	***S-CIN1*** *A cloud-based e-health framework SHALL allow administrators to define certain security profiles (such as specific software bundles and specific configurations) to be considered as trusted.*	Cloud provider integrity
all	***S-CIN2*** *A cloud-based e-health framework SHALL support integrity verification of virtualization servers.*	Cloud provider integrity
all	***S-CIN3*** *A cloud-based e-health framework SHALL implement automatic equipment identification based on a hardware root of trust (where applicable).*	Medical device hardware integrity
all	***S-CIN4*** *A cloud-based e-health framework SHALL support executing security-sensitive workloads in a dedicated (isolated) secure computing environment.*	Cloud provider integrity
all	***S-NET1*** *A cloud-based e-health framework SHALL provide support mechanisms to authenticate as well as to verify and protect the integrity of software components in the network infrastructure.*	Cloud provider integrity and medical device hardware integrity

all	***S-NET2*** *A cloud-based e-health framework SHALL support strong authentication and robust traceability of network infrastructure management.*	Master authority authentication, cloud provider integrity and medical device hardware integrity
all	***S-NET3*** *A cloud-based e-health framework SHALL support deployment of secure communication channels for in-transit protection of data between endpoints both on the same network infrastructure, as well as on network infrastructures of different institutions.*	Encryption and cloud provider integrity
all	***S-ACC3*** *A cloud-based e-health framework SHALL be adaptable to organization-specific key management schemes to support diverse authentication models and secure communication protocols.*	Searchable encryption, ABE schemes and Key Tray
all	***S-ACC4*** *A cloud-based e-health framework SHALL use a key generation algorithm that guarantees the generated keys are secure (long enough and generated using enough randomness).*	Searchable encryption and ABE schemes
all	***H-AVA1*** *A cloud-based e-health framework SHALL NOT impede the availability of data for lawful processing.*	Searchable encryption and ABE schemes
all	***H-AUD1*** *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	Cloud provider integrity
all	***H-AUD2*** *A cloud-based e-health framework SHALL specifically identify access that has overridden policies (e.g., in a medical emergency situation).*	Searchable encryption and ABE schemes
all	***H-AUD3*** *A cloud-based e-health framework SHALL protect the integrity of the audit trail.*	Encryption and cloud provider integrity
all	***H-AUD4*** *A cloud-based e-health framework SHALL enable authorized access to the audit trail.*	Encryption and cloud provider integrity
all	***H-AUD5*** *A cloud-based e-health framework SHALL ensure the audit trail maintains records of disclosures of the audit trail itself.*	Encryption and cloud provider integrity
all	***H-DRT3*** *A cloud-based e-health framework SHALL enable a data subject or legal representative to obtain access to the personal data concerning the data subject.*	Access control
all	***H-DST2*** *A cloud-based e-health framework SHALL encrypt personal data during storage.*	Data sharing and revocation using SSE and ABE
all	***H-DST3*** *A cloud-based e-health framework SHALL secure encryption/decryption keys.*	Searchable encryption and ABE schemes

all	***S-DAM1*** *A cloud-based e-health framework SHALL provide mechanisms to enforce reliable destruction of workloads and configuration in isolated enclaves.*	Use of Cloud/HPC resources
D1-UC1	***Patient must have a single identifier.*** ***H-UID1*** *A cloud-based e-health framework SHALL uniquely identify the data subject to whom a record entry belongs to.*	Registration authority
	Patient must have a single EMR, no duplications. ***H-UID2*** *A cloud-based e-health framework SHALL uniquely identify each resource, including health record entries.*	Registration authority
	Patient must encrypt his/her EMR according to framework standard mechanism. ***S-DAM2*** *A cloud-based e-health framework SHALL provide mechanisms for secure storage and sharing of data.*	Searchable encryption
	Patient must be able to store her own EMR. ***H-DST1*** *A cloud-based e-health framework SHALL store information committed by authorized users.*	Searchable encryption and ABE schemes
D1-UC2, D1-UC3	***Health professionals must decrypt the patient's EMR according to framework standard mechanism.*** ***S-DAM2*** *A cloud-based e-health framework SHALL provide mechanisms for secure storage and sharing of data.*	Searchable encryption and ABE schemes
	Professionals must be involved in emergency session to have break glass access to patient's EMR. ***S-ACC1*** *A cloud-based e-health framework SHALL support access control mechanisms to provide users access to the services that they have been specifically authorized to use.*	Searchable encryption and ABE schemes
	All professionals must authenticate themselves in the emergency session. ***S-ACC2*** *A cloud-based e-health framework SHALL support secure authentication methods to control access by remote users.*	ABE schemes and Master Authority authentication
	The access control to patient's EMR must enable break glass access.	ABE schemes and Master

	H-ACC1 *A cloud-based e-health framework SHALL enable the representation of policies that control access to health records.*	Authority authentication
	All EMRs must be controlled by the access policies. ***H-ACC2*** *A cloud-based e-health framework SHALL be able to associate a health record entry or a group of health record entries with policies that apply to them.*	ABE schemes and Master Authority authentication
	All professionals must comply as policies to have access to patient's EMR. ***H-ACC3*** *A cloud-based e-health framework SHALL enable processing of health records based on the applicable policies.*	ABE schemes and Master Authority authentication
	Professionals involved in the emergency session must be allowed to break glass access. ***H-ACC4*** *A cloud-based e-health framework SHALL NOT require the presence of explicit consent in order to process personal data, if there are other legal grounds that permit the processing.*	ABE schemes and Master Authority authentication
D1-UC4	***Professionals involved in the emergency session must be able to retrieve the patient's EMR*** ***H-DRT1*** *A cloud-based e-health framework SHALL enable authorized users to retrieve partial or complete health information based on specific criteria.*	ABE schemes and Master Authority authentication
	Professionals involved in the emergency session must be able to search for specific information in the patient's EMR. ***H-DRT2*** *A cloud-based e-health framework SHALL enable an authorized user accessing a health record entry that contains one or more links to be able to retrieve the referenced resources, such as health record entries and external resources* (e.g., images).	Searchable encryption
	Professionals involved in the emergency session have the pre-consent by law. ***H-DRT3*** *A cloud-based e-health framework SHALL enable a data subject or legal representative to obtain access to the personal data concerning the data subject.*	ABE schemes

	Professionals from different treatment team must have access to data from previous treatment team. ***H-DRT5*** *A cloud-based e-health framework SHALL enable the receiving of personal data from another [healthcare] organization* based on the appropriate legal ground.*	ABE schemes
D1-UC5	***Professionals involved in the emergency session must be able to store new data to patient's EMR.*** ***H-DST1*** *A cloud-based e-health framework SHALL store information committed by authorized users.*	Searchable encryption and ABE schemes
	The new data stored must be accessible by other treatment teams. ***H-DRT4*** *A cloud-based e-health framework SHALL enable sharing of personal data with another [healthcare] organization* based on the appropriate legal ground.*	Searchable encryption and ABE schemes
D1-UC6 D1-UC7	***The team must have the break glass access revoked when the patient is no longer under their treatment.*** ***S-ACC5*** *A cloud-based e-health framework SHALL support efficient and effective key revocation.*	Searchable encryption and ABE schemes
D1-UC8 D1-UC9	***The system must revoke access for researchers to data subject's EMR if the subject wants to be forgotten.*** ***H-DST5*** *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject choose to exercise their right to be forgotten.*	Access control
	H-DAR1 *A cloud-based e-health framework SHALL support authorized analyses of health data.*	Access control
	H-DAR2 *A cloud-based e-health framework SHALL enable authorized users to transform personal data to pseudonymized data.*	Access control
	H-DAR3 *A cloud-based e-health framework SHALL enable authorized users to transform personal data to anonymized data.*	Access control
	H-DAR4 *A cloud-based e-health framework SHALL support the implementation of privacy-preserving distributed data mining.*	Functional encryption

Table 15: Summary of the security and privacy requirements for Demonstrator 1

* [healthcare] is in close brace to indicate it as a possible organization type; it is not meant to limit the data transfer to/from healthcare organizations.

6.9 Demonstrator data requirements

All the use cases imply that patient data are stored in a central (virtual) system with an assumption of no EMR duplications exist. This would require

- ***S-CIN1*** *A cloud-based e-health framework SHALL allow administrators to define certain security profiles (such as specific software bundles and specific configurations) to be considered as trusted.*
- ***S-CIN2*** *A cloud-based e-health framework SHALL support integrity verification of virtualization servers.*
- ***S-CIN3*** *A cloud-based e-health framework SHALL implement automatic equipment identification based on a hardware root of trust (where applicable).*
- ***S-CIN4*** *A cloud-based e-health framework SHALL support executing security-sensitive workloads in a dedicated (isolated) secure computing environment.*
- ***H-UID2*** *A cloud-based e-health framework SHALL uniquely identify each resource, including health record entries.*

In the use case D1-UC9, research data are stored in a central (virtual) system. This requires

- ***H-DAR1*** *A cloud-based e-health framework SHALL support authorized analyses of health data.*
- ***H-DST5*** *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject chooses to exercise their right to be forgotten.*
- ***H-DAR4*** *A cloud-based e-health framework SHALL support the implementation of privacy-preserving distributed data mining.*

6.10 Testbed

The demonstrator will be fully implemented from scratch in a simulated environment, exploiting the testbed infrastructure and framework offered by the ASCLEPIOS project.

7 Demonstrator 2: Collaboration and Analysis Platform for Inpatient and Outpatient Sleep Medicine

7.1 Introduction

Sleep disorders are very common and many hospitals add a sleep laboratory for inpatient services. The sleep laboratory serves with polysomnography, an investigation over night with the recording of biosignals (electroencephalogram, electrooculogram, electromyogram, electrocardiogram, respiratory flow, respiratory movement, oxygen saturation, pulse rate, body position, movement of the legs), video, and sound (snoring, speaking during sleep). In addition, sleep laboratories perform daytime testing of sleepiness in order to check whether sleep fulfills the purpose of restoring performance. This is also needed to judge on people who fell asleep while driving or at their job performance, to check whether they have slept too short or have a sleep disorder. For some sleep disorders, blood and genetic testing need to be performed. Questionnaires on cognitive function are also a part of sleep medicine. In general, sleep medicine is similar to clinical neurophysiology and is characterized by a multitude of signals, images, clinical laboratory data, and textual data. Sleep medicine has challenges connected with the data variety.

Most recently patients bring data from their smartphone app monitoring their sleep and want to add these data to their EMR. This puts another challenge on medical data storage options.

7.2 Background

The interdisciplinary Center of Sleep Medicine, Charité Universitätsmedizin Berlin, treats in average 8000 patients with sleep disorders of any genesis yearly. It is mainly sleep-related respiratory disorders (e.g. sleep apnea) and difficulties initiating or maintaining sleep. At the sleep laboratory, the sleep of patients is examined both ambulatory in the home and stationary in the laboratory. The ambulatory procedure is called polygraphy (PG), also known as home sleep testing (HST). The procedure called polysomnography (PSG) is performed in the laboratory. The medical staff involves doctors (somnologists), nurses, medical-technical staff and scientists both in the outpatient clinic and at the hospital ward. The nightly sleep recording is continuously controlled and monitored by medical students.

The sleep laboratory consists of specially equipped patient rooms and an additional room where the monitoring and recording devices are located.

Polysomnography is a night-long recording of various body functions such as brain waves, eye movements, breathing, muscle tension or oxygen saturation of the blood. On the basis of these measured values, a very accurate sleep profile of the individual sleep stages (e.g. awake state, REM sleep, deep sleep, short wake-up reactions) can be recreated the following morning by special medical stuff. This sleep profile allows conclusions to be drawn about the quality of sleep and the causes that may impair sleep quality. Outpatient polygraphy, on the other hand, is a very reduced variant of the sleep examination and is always preceded by polysomnography according to the guidelines for a preliminary diagnosis.

During the day sleep labs perform sleepiness tests to check whether sleep fulfils the purpose of improving performance. Attention tests are also part of daily routine. This is necessary to assess people who have fallen asleep while driving or at work, to check whether they have slept too short or have a serious sleep disorder. Blood tests, genetic tests for specific types of sleep disorders are performed. Questionnaires on cognitive function are also part of sleep medicine to assess performance during the day.

7.2.1 Motivation

Especially for sleep-related respiratory disorders, the diagnostic process includes outpatient polygraphy as well as inpatient polysomnographic examination in the sleep medicine routine. About 2500 patients of our center receive a polygraphy and an additional polysomnography yearly. This is a long-term and cost-intensive process, both for patients and medical staff. These processes can be optimized in terms of time and costs.

Recently, patients have been asked to evaluate stored data from their smartphones that monitors sleep. Including such data in patient's EMR is another challenge for the possibilities of medical data storage.

7.2.2 State-of-the-art

A patient's journey from the initial examination to the sleep laboratory stay is prescribed according to guidelines.

First the patient introduces him-/herself to the attending physician in the sleep ambulance for an anamnesis interview. If sleep-related respiratory disorders are suspected, the nursing staff will give him/her a polygraphy device for home examinations (HST), which she/he returns the following day for evaluation. At a later meeting with the doctor, it is decided on the basis of the results and the anamnesis whether inpatient stay in the sleep laboratory is necessary.

The examinations are complex and the waiting times are long. It usually takes many months for a patient to complete the process of anamnesis and therapy.

A study is currently underway to test an alternative method of care when the complex pre-diagnostic process is taken over by an external service provider. The service provider organizes the delivery and pickup of the device to/from the patient's home. The recorded sleep data are directly transmitted digitally and retrieved by the clinicians. This procedure would have the advantage of considerable time saving in the sleep ambulances since an appointment allocation, issuing of the outpatient device, renewed presentation of the patient for the device delivery and planning of the meeting appointment, as well as the acquisition and maintenance of several polygraphy devices could be omitted. This would create resources for inpatients and sleep ambulances and significantly reduce the waiting time for appointments.

However, in both approaches of device delivery, successful home sleep testing is of high importance. This includes the correct application of the sensors and the onset of the system. A loosening of the sensors early in the night would make the measurement unusable. Additionally, the video data of big volume (5 GB) made during the night. Due to limited local resources, the video data are usually deleted for most of the patients.

During the further diagnostic procedure in sleeping laboratory, different data are created: polysomnography, hypnogram, and a patient's report. Polysomnography includes multidimensional biosignal recordings, such as electroencephalography (EEG), electrooculography (EOG), electrocardiography (ECG), blood oxygen saturation values, body movements and snoring parameters.

As sleep disorders may be very diverse, interdisciplinary teleconsultation of experts from different medical fields would increase the efficiency of sleep diagnosis. Currently it is difficult due to GDPR regulations as the PSG data would be required to transfer.

All polysomnographic data contain common labels, common channel ordering and common signal references required for view and analysis. Due to lack of standardization, different

hardware may transfer signals in a variety of data formats. These data can also be of high volume (about 300-400 MB).

The Center for Biomedical Image and Information Processing has developed a collaboration platform for sleep research, based on the eXtensible Neuroimaging Archive Toolkit (XNAT). It is able to store the data in a standardized way and different analysis tools are available to process the PSG data in the cloud.

Automated data access for cloud-based remote execution is token-based. A basic security check according the IT-Grundschutz has been applied to the platform.

7.2.3 Envisioned situation

In the ASCLEPIOS project, the goal is to extend the collaboration platform in order to address the abovementioned challenges. The extended functionalities enable direct communication between the different actors (patients, technical and medical staff, physicians and researchers) including visualization of the biosignal recordings (HST, PSG and videos). On the other hand, they need to enable fine-grained access to different patient-related data for medical staff, patients, external service providers and researchers. Furthermore, automated quality control and analysis of the biosignal data in the cloud should be enabled to allow better near-time remote monitoring and alerting. Such functionality can reduce failures in home sleep testing and produce better and faster diagnosis of sleep disorders and reduce costs for healthcare sector. Patients can be processed and treated more efficiently and effectively, reducing the time the patient suffers from untreated sleep disorders. Research on the data may allow better phenotyping of sleep disorders leading to personalized treatment options.

7.3 Use cases

Table 16 is a summary table for the use cases developed for the demonstrator.

ID	Name
D2-UC1	Start home sleep testing session
D2-UC1a	Outsource home sleep testing
D2-UC2	Perform home sleep testing
D2-UC3	Perform home sleep testing evaluation
D2-UC4	Perform inpatient diagnostics
D2-UC5	Perform teleconsultation
D2-UC6	Add data
D2-UC7	Access data
D2-UC8	Perform research

Table 16: Demonstrator 2 use cases

Figure 3 shows the use case diagram for the demonstrator.

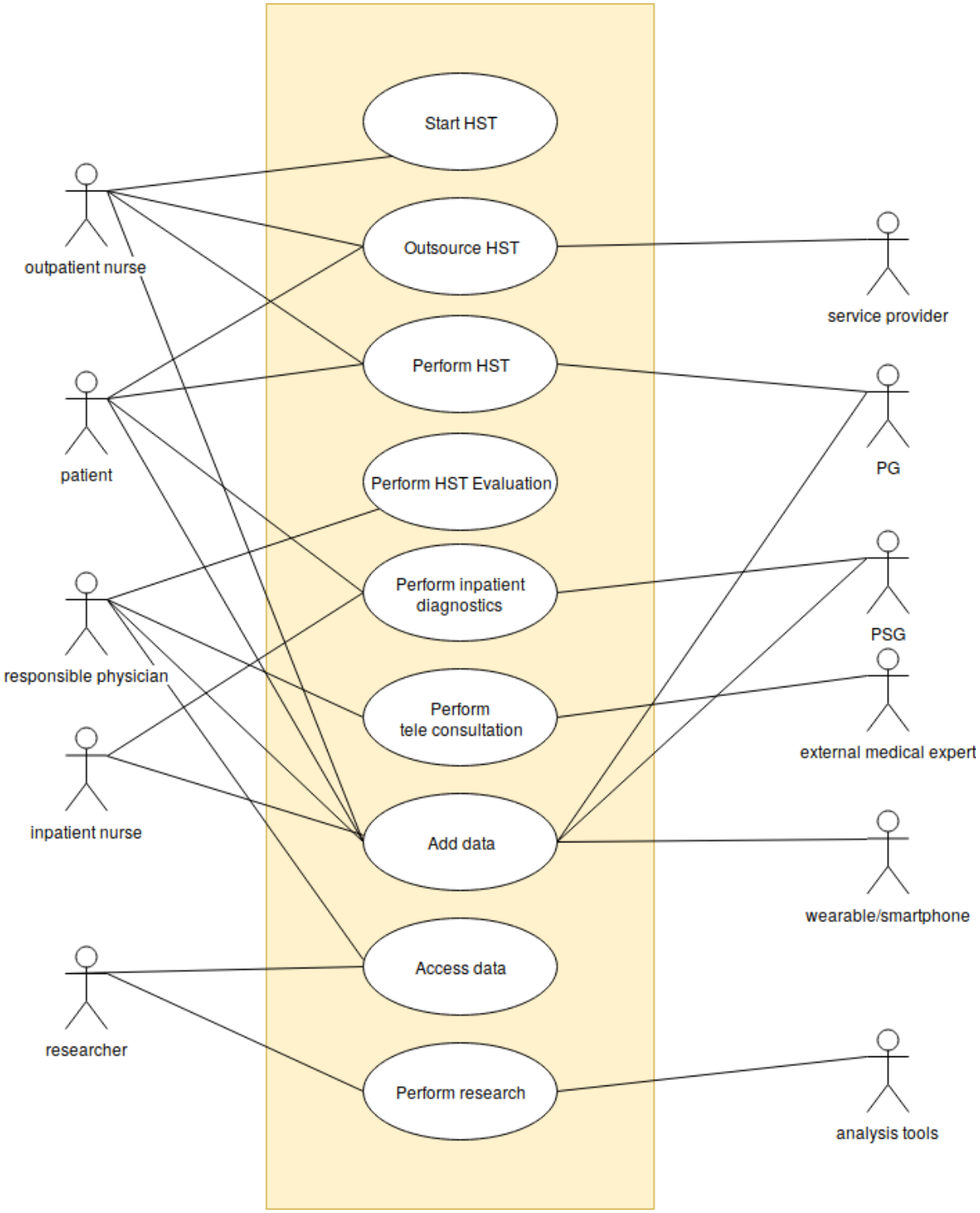

Figure 3: Demonstrator 2 use case diagram

7.3.1 Start home sleep testing session

The physician recommends the patient to have a polygraphy device at home if sleep-related respiratory disorders are suspected. Table 17 shows the description of the use case.

Attribute	Description

Use case ID	D2-UC1
Use case goal	Introduce pre-diagnostics
Assumptions & pre-conditions	• Authorization of hospital staff (sleep medicine outpatient clinic) for access to the digital patient database
Use case initiation	Outpatient nurse creates a digital data file of the patient
Use case main scenario	The nurse hands out a PG device to the patient with some instructions as well as description on how to connect the device and to the support service
Use case alternate scenario	D2-UC1a
Use case failure scenario	• No connectivity to platform • Staff is not authorized
Acceptance criteria	Patient record is created

Table 17: "Start HST session" use case description summary

7.3.2 Outsource home sleep testing

In the alternative use case, the home sleep testing procedure is organized by an external service provider. The service provider commissions a shipping company to send and collect the PG device for the nocturnal measurement to/from the patient's home. Table 18 shows the description of the use case.

Attribute	**Description**
Use case ID	D2-UC1a
Use case goal	Ship the PG device to the patient by the external provider
Assumptions & pre-conditions	• Authorization of hospital staff (sleep medicine outpatient clinic) for access to the digital patient database • Consent from the patient for the data transmission to the provider and the shipment company • Authorization of service provider and shipping company to access the required bit of data • Accessibility/presence of the patient for shipping process
Use case initiation	Outpatient nurse creates a digital data file of the patient
Use case main scenario	The outpatient nurse sends a service request to the service provider. The service provider creates a delivery order to the shipping company. Patient receives the device
Use case alternate scenario	D2-UC1
Use case failure scenario	• No connectivity • Actor not authorized • Patients are not available device cannot be delivered

Acceptance criteria	Patient record is created and all actors access only the required subset of data

Table 18: "Outsource HST" use case description summary

7.3.3 Perform home sleep testing

The patient attaches the sensors and starts the recording. There might occur problems with the handling of the device. Table 19 shows the description of the use case.

Attribute	**Description**
Use case ID	D2-UC2
Use case goal	Perform a valid PG measurement
Assumptions & pre-conditions	• Authorization of the patient to access the platform • Integrity of the device accepted
Use case initiation	The patient setups the device
Use case main scenario	The patient attaches the sensors and starts the recording. In case of problems or questions, the patient contacts the supporting actor (medical staff in D2-UC1 or service provider D2-UC1a). The support helps the patient to attach the device and connect it to the platform. It further checks remotely the signal quality and may give feedback to improve signal quality. During the night, the signal quality is monitored and the support may decide to send an alert to the patient. The PG measurement is stored in the patient's EMR
Use case alternate scenario	None
Use case failure scenario	• No connectivity • Actor not authorized • Device is defective
Acceptance criteria	PG data are stored in the patient's EMR and can be accessed by authorized actors

Table 19: "Perform HST" use case description summary

7.3.4 Perform home sleep testing evaluation

The physician has access to the digital patient data and evaluates the polygraphy results online. Table 20 shows the description of the use case.

Attribute	**Description**
Use case ID	D2-UC3
Use case goal	Physician performs evaluation of the HST results

Assumptions & pre-conditions	• Authorization of hospital staff (sleep medicine outpatient clinic) for access to the digital patient database • Consent from the patient for the digital transmission of personal data
Use case initiation	Physician has accesses the patient's EMR
Use case main scenario	Physician evaluates the data online for diagnosis
Use case alternate scenario	None
Use case failure scenario	• Physician has no access to the EMR • Physician has no access to PG data within the EMR
Acceptance criteria	A physician is able to visualize the stored PG data

Table 20: "Perform HST evaluation" use case description summary

7.3.5 Perform inpatient diagnostics

Due to the HST results the patient must be monitored in the sleep laboratory. During the diagnostic process, there are many kinds of data gathered from different biosignals, from different processes and in different formats. Table 21 shows the description of the use case.

Attribute	**Description**
Use case ID	D2-UC4
Use case goal	Collect the necessary patient's sleep data
Assumptions & pre-conditions	• Patient consent for collecting and processing personal data
Use case initiation	The patient is admitted to the sleep laboratory
Use case main scenario	The nurse is preparing the measurements and connects the devices to the platform
Use case alternate scenario	None
Use case failure scenario	• Sensors are not capable to collect data • Patient gives no consent for data collection
Acceptance criteria	The necessary data are stored to the EMR

Table 21: "Perform inpatient diagnostics" use case description summary

7.3.6 Perform teleconsultation

After the data are accessible for authorized persons, the data can be viewed remotely for digital diagnosis. It is possible to search and create different reports from the data. It must be possible to give access for data viewing for third party physicians in case of being asked to consult the case. Table 22 shows the description of the use case.

Attribute	Description
Use case ID	D2-UC5
Use case goal	Two physicians discuss the case remotely
Assumptions & pre-conditions	• The patient has given consent to share the EMR (or the particular data) for second opinion • The external colleague has access to the platform
Use case initiation	The physician invites the external expert to a tele consult session.
Use case main scenario	The external expert can view the measurements. He/she can follow the pointer of the physician and can interactively point on important events
Use case alternate scenario	None
Use case failure scenario	• The platform is not reachable • Remote viewing is not active • The person is not authorized to view the data
Acceptance criteria	The data can be interactively inspected by two authorized persons simultaneously

Table 22: "Perform teleconsultation" use case description summary

7.3.7 Add data

Upload of data from arbitrary wearable devices is increasingly demanded by patients. Table 23 shows the description of the use case.

Attribute	Description
Use case ID	D2-UC6
Use case goal	Upload collected data to platform for further diagnostics or research
Assumptions & pre-conditions	The necessary data are collected, patient consent for personal data processing is in force
Use case initiation	Collected data are ready to be uploaded
Use case main scenario	The data are uploaded to platform
Use case alternate scenario	None
Use case failure scenario	• The platform is not accessible • The actor is not authorized • The data are not available or damaged • The device is not recognized
Acceptance criteria	The data are uploaded and stored in the correct EMR

Table 23: "Add data to platform" use case description summary

7.3.8 Access data

Once the data are collected and uploaded, it must be possible for authorized persons to access it for further diagnosis or research. Table 24 shows the description of the use case.

Attribute	Description
Use case ID	D2-UC6
Use case goal	Access the data
Assumptions & pre-conditions	• Data are collected and uploaded to the platform • The patient has given consent for the planned action
Use case initiation	The actor searches for the respective data
Use case main scenario	The actor finds the data and can access it
Use case alternate scenario	None
Use case failure scenario	• Platform is not reachable • Physician is not authorized to have access • Researcher is not authorized to have access
Acceptance criteria	The authorized person has access to platform and data

Table 24: "Access data" use case description summary

7.3.9 Research process

It must be possible to access the platform for different research studies. As collected data have a big value for research, they must have access to the platform to access the data. It can be in-house or third-party research. A possibility to create reports and remote viewing must be also active. Table 25 shows the description of the use case.

Attribute	Description
Use case ID	D2-UC8
Use case goal	The researchers have access to platform, can access data, process it and create reports
Assumptions & pre-conditions	• Data are stored in platform • The researchers are authorized for processing data • Patients have given consent to the planned actions
Use case initiation	A researcher selects the data
Use case main scenario	The researcher processes the data with the available functions on a remote cloud platform and downloads the report
Use case alternate scenario	None

Use case failure scenario	• The researcher is not authorized to access the platform • Platform is not reachable • The remote viewing function for researcher is not active • Reporting is not active
Acceptance criteria	A researcher created a reproducible result by processing a dataset

Table 25: "Research process" use case description summary

7.4 Components/mechanisms of ASCLEPIOS framework involved in the demonstrator

These components/mechanisms of the ASCLEPIOS are involved in the demonstrator:

- *Use of Cloud/HPC resources* to store EMR and measurements.
- *Data sharing and revocation using SSE and ABE* to securely find and access the data by various actors.
- *Privacy-Preserving analytics using FE* to perform artefact and apnea detection as well as analytics for research.
- *Medical device hardware integrity* to integrate PG/PSG and wearables.
- *Cloud provider integrity* to fulfill standards for medical data processing.
- *Increase GDPR and Security Awareness* to enable patients to fully understand what they consent to.

7.5 Demonstrator security requirements

The security and privacy requirements of the use cases (specified in *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements*) are gathered in Table 26.

The project suggested encryption techniques will be implemented in eXtensible Neuroimaging Archive Toolkit (XNAT) to provide the secure transfer, exchange and storage of the data. The platform will be extended by the encryption methods for metadata and content encryption. The HL7 FHIR standard will be used for metadata presentation implementing.

To achieve the authorized access, for different actors the attribute-based encryption will be implemented. For this function, the Administrative actor will be added to the use cases. To use data in different research studies, the patient personal data will be pseudonymized by default. Remote visualization of PSG data will be enabled in order to avoid downloading of large volumes of data. For the prototype, only anonymized/artificial data will be used.

The names of the patients must be pseudonymized for research studies. As the gathered data considered as personal, the security methods must be implemented. There are several actors involved during diagnosis/research (see Table 4), the appropriate access control must be implemented.

Use case	Requirement	ASCLEPIOS functionalities
all	***S-ACC1*** *A cloud-based e-health framework SHALL support access control mechanisms to provide users access to the services that they have been specifically authorized to use.*	Data sharing and revocation using SSE and ABE

all	***S-ACC2*** *A cloud-based e-health framework SHALL support secure authentication methods to control access by remote users.*	Data sharing and revocation using SSE and ABE
all	***S-ACC3*** *A cloud-based e-health framework SHALL be adaptable to organization-specific key management schemes to support diverse authentication models and secure communication protocols.*	Data sharing and revocation using SSE and ABE
all	***S-ACC4*** *A cloud-based e-health framework SHALL use a key generation algorithm that guarantees the generated keys are secure (long enough and generated using enough randomness).*	Data sharing and revocation using SSE and ABE
all	***S-ACC5*** *A cloud-based e-health framework SHALL support efficient and effective key revocation.*	Data sharing and revocation using SSE and ABE
D2-UC2, D2-UC3, D2-UC4, D2-UC5, D2-UC6, D2-UC8	***S-CIN1*** *A cloud-based e-health framework SHALL allow administrators to define certain security profiles (such as specific software bundles and specific configurations) to be considered as trusted.*	Cloud provider integrity
D2-UC2, D2-UC8	***S-CIN2*** *A cloud-based e-health framework SHALL support integrity verification of virtualization servers.*	Cloud provider integrity
D2-UC2, D2-UC4, D2-UC6	***S-CIN3*** *A cloud-based e-health framework SHALL implement automatic equipment identification based on a hardware root of trust (where applicable).*	Medical device hardware integrity
D2-UC2, D2-UC8	***S-CIN4*** *A cloud-based e-health framework SHALL support executing security-sensitive workloads in a dedicated (isolated) secure computing environment.*	Cloud provider integrity
D2-UC2, D2-UC8	***S-DAM1*** *A cloud-based e-health framework SHALL provide mechanisms to enforce reliable destruction of workloads and configuration in isolated enclaves.*	Use of Cloud/HPC resources
all	***S-DAM2*** *A cloud-based e-health framework SHALL provide mechanisms for secure storage and sharing of data.*	Use of Cloud/HPC resources
D2-UC2, D2-UC3, D2-UC5, D2-UC8	***S-NET1*** *A cloud-based e-health framework SHALL provide support mechanisms to authenticate as well as to verify and protect the integrity of software components in the network infrastructure.*	Cloud provider integrity and medical device hardware integrity
all	***S-NET2*** *A cloud-based e-health framework SHALL support strong authentication and robust traceability of network infrastructure management.*	Cloud provider integrity and medical device hardware integrity
all	***S-NET3*** *A cloud-based e-health framework SHALL support deployment of secure communication channels for in-transit protection of data between endpoints both on the same network infrastructure, as well as on network infrastructures of different institutions.*	Encryption and cloud provider integrity

all	***H-UID1*** *A cloud-based e-health framework SHALL uniquely identify the data subject to whom a record entry belongs to.*	Data sharing and revocation using SSE and ABE
all	***H-UID2*** *A cloud-based e-health framework SHALL uniquely identify each resource, including health record entries.*	Data sharing and revocation using SSE and ABE
all (biosignal recordings)	***H-CIS1*** *A cloud-based e-health framework SHALL be able to represent links between resources, such as health records entries and external resources (e.g., images).*	Data sharing and revocation using SSE and ABE
all	***H-DST1*** *A cloud-based e-health framework SHALL store information committed by authorized users.*	Data sharing and revocation using SSE and ABE
all	***H-DST2*** *A cloud-based e-health framework SHALL encrypt personal data during storage.*	Data sharing and revocation using SSE and ABE
all	***H-DST3*** *A cloud-based e-health framework SHALL secure encryption/decryption keys.*	Cloud provider integrity
all	***H-DST4*** *A cloud-based e-health framework SHALL ensure minimal storage overhead.*	Use of Cloud/HPC resources
D2-UC7	***H-DST5*** *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject chooses to exercise the right to be forgotten.*	Cloud provider integrity
all	***H-DRT1*** *A cloud-based e-health framework SHALL enable authorized users to retrieve partial or complete health information based on specific criteria.*	Data sharing and revocation using SSE and ABE
D2-UC2, D2-UC3, D2-UC4, D2-UC5, D2-UC7, D2-UC8	***H-DRT2*** *A cloud-based e-health framework SHALL enable an authorized user accessing a health record entry that contains one or more links to be able to retrieve the referenced resources, such as health record entries and external resources (e.g., images).*	Data sharing and revocation using SSE and ABE
D2-UC7	***H-DRT3*** *A cloud-based e-health framework SHALL enable a data subject or legal representative to obtain access to the personal data concerning the data subject.*	Data sharing and revocation using SSE and ABE
D2-UC5	***H-DRT4*** *A cloud-based e-health framework SHALL enable sharing of personal data with another [healthcare] organization* based on the appropriate legal ground.*	Data sharing and revocation using SSE and ABE
D2-UC5	***H-DRT5*** *A cloud-based e-health framework SHALL enable receiving personal data from another [healthcare] organization* based on the appropriate legal ground.*	Data sharing and revocation using SSE and ABE
D2-UC2, D2-UC8	***H-DAR1*** *A cloud-based e-health framework SHALL support authorized analyses of health data.*	Privacy-Preserving analytics using FE

D2-UC1a, D2-UC8	***H-DAR2*** *A cloud-based e-health framework SHALL enable authorized users to transform personal data into pseudonymized data*	Data sharing and revocation using SSE and ABE
D2-UC8	***H-DAR3*** *A cloud-based e-health framework SHALL enable authorized users to transform personal data to anonymized data.*	Data sharing and revocation using SSE and ABE
D2-UC8	***H-DAR4*** *A cloud-based e-health framework SHALL support the implementation of privacy-preserving distributed data mining.*	Privacy-Preserving analytics using FE
all	***H-ACC1*** *A cloud-based e-health framework SHALL enable the representation of technically supported access policies that control access to health records.*	Data sharing and revocation using SSE and ABE
all	***H-ACC2 A*** *cloud-based e-health framework SHALL be able to associate a health record entry or a group of health record entries with policies that apply to them.*	Data sharing and revocation using SSE and ABE
D2-UC2, D2-UC8	***H-ACC3*** *A cloud-based e-health framework SHALL enable processing of health records based on the applicable technically supported access policies.*	Privacy-Preserving analytics using FE
D2-UC3, D2-UC5, D2-UC7, D2-UC8	***H-ACC5*** *A cloud-based e-health framework SHALL NOT enable the processing, except for storage, of personal data that has been restricted.*	Data sharing and revocation using SSE and ABE
all	***H-AUD1*** *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	Cloud provider integrity
all	***H-AUD3*** *A cloud-based e-health framework SHALL protect the integrity of the audit trail.*	Encryption and cloud provider integrity
all	***H-AUD5*** *A cloud-based e-health framework SHALL ensure the audit trail maintains records of disclosures of the audit trail itself.*	Encryption and cloud provider integrity

Table 26: Summary of the security and privacy requirements for Demonstrator 2

* [healthcare] is in close brace to indicate it as a possible organization type; it is not meant to limit the data transfer to/from healthcare organizations.

7.6 Demonstrator data requirements

As described above, diverse data are involved in the demonstrator, including EMR with typical database structure and massive data stored as files. The existing system has a built-in REST API for data access, and stores metadata in a postgreSQL database and the files in a specific file system structure. It is envisioned to implement a full HL7-FHIR interface to the collaboration platform. Algorithms are stored as docker containers in a public registry

7.7 Testbed

The collaboration platform is virtualized and available as a Vagrant VM (40). During the project, it should be installed in a private cloud (41). The usage of a public cloud is envisioned

since a private cloud will be not applicable due to many actors and organisations involved in the use case. Analytics tools require a container cluster or at least during the project a server providing a docker runtime environment (42).

8 Demonstrator 3: Privacy-Preserving Monitoring and Benchmarking of Antibiotic Prescription

8.1 Introduction

Rapidly learning from routine healthcare data is important to improve quality of care (6–10). However, there are privacy concerns of patients, clinicians and health institutions for the reuse of health data. Therefore, there is a legitimate need to protect the privacy and confidentiality of the people and corporate entities the data represent (11). Norwegian Centre for E-health Research (NSE) developed a privacy-preserving tool for monitoring and providing feedback to clinicians on their antibiotic prescriptions.

8.2 Background

Studies have shown that feedbacks containing the clinical performance of a clinician in comparison with peers is effective for behavioral changes leading to quality improvements (6,9,10,12). Simple questions like “What percentage of acute upper respiratory infection patients do I treat with antibiotics? What is the average of my peers?” are expected to have high impact. To sustain the achieved gains, continuous feedback to clinicians is necessary even after improvement has taken place (6,13,14). The data sources we considered for such feedbacks are structured EHR data distributed across participating health institutions. In practice, clinical performance comparisons need to be done with clinicians having similar patients, working in a similar medical domain and/or possibly from the same geographical area.

De-identification is commonly used to protect patients’ privacy. However, for the current demonstrator, de-identification is not suitable to protect the privacy of clinicians and health institutions. Privacy-preserving distributed data mining (PPDDM) allows computing on data distributed across multiple sources without revealing sensitive information apart from aggregated results (15–17). PPDDM techniques are developed based on a technique called secure multi-party computation (SMPC) (18). Practical use of SMPC protocols is limited due to their lack of efficiency and scalability required for the processing health data in practice.

8.2.1 Motivation

The increasing emergence of antibiotic resistant bacteria observed worldwide is lowering the success rates of infection treatment using antibiotics. Inappropriate antibiotics prescriptions lead to antibiotics resistance and adverse events. Every year around 700,000 individuals die because of antibiotic resistance (19). The number is increasing (20). GPs are responsible for the majority of antibiotics prescriptions; in Norway, for example, GPs prescribe around 80% of all antibiotics (21). Changes in the antibiotics prescriptions of GPs can have a significant impact for reducing inappropriate consumptions of antibiotics. Therefore, we consider GPs as the primary users of the tool.

8.2.2 State-of-the-art

We consider the cases when EHRs contains information about patients, clinicians, and health institutions. Therefore, EHR data cannot be disclosed outside the organization that originally recorded the data. Clinical performance indicators of a clinician are also sensitive information, and can only be accessed by the clinician him-/herself. However, statistics generated from combined data of a group of health institutions, such as aggregated indicators of clinicians across multiple health institutions, does not reveal sensitive information and can be disclosed publicly.

NSE has designed and deployed a system, called the Snow system (22), for the reuse of health data distributed across multiple health institutions in Norway. The Snow system contains a server installed at health institutions including GP offices (Snow server), and a coordinator server. The Emnet tool is developed on top of the Snow system for privacy-preserving distributed statistical computation on without revealing sensitive information about patients, clinicians and health institutions (15). Emnet has a coordination agent running at the coordinator, and a computation agent running at the GP offices. These agents jointly execute secure protocols for computing a given statistical function without revealing anything apart from aggregated results.

NSE developed current demonstrator based on the Snow system and Emnet. The demonstrator maintains the data at the GP offices and locally computes indicators for each GP regarding his/her antibiotics prescriptions, and uses Emnet for computing group level indicators of all participating GPs based on the data distributed across GP offices. Currently, the data stored on the Snow server is pseudonymized. However, pseudonymized data do not provide strong privacy guarantee due to re-identification risk (25).

8.2.3 Envisioned situation

The demonstrator will improve the security and privacy guarantee offered to the GPs and their patients by combining privacy-preserving distributed statistical computation with the techniques developed in the ASCLEPIOS. ASCLEPIOS enables encrypted data storage at the GP offices, in particular on the Snow servers. The local computations (such as querying the required sub-dataset using the searchable encryption and statistical computations using the functional encryption) on the Snow servers will take place on the encrypted data. This will minimize the security and privacy risks in case an outside adversary gains access to the Snow server or internal attack (19). Additionally, GPs and patients will have better access control on their data, since the attribute-based encryption makes sure that computation on the data is restricted to authorized people.

The increased security and privacy guarantee increases GPs willingness for using the tool. Increased tool use will lead to more GPs change their behavior on antibiotics prescriptions, which leads to a higher quality healthcare to individuals and the society, in general.

8.3 Use cases

Table 27 is a summary table for the use cases developed for Demonstrator 3.

ID	Name
D3-UC1	Approve data use
D3-UC2	Create dataset
D3-UC3	Compute group level quality indicators
D3-UC4	Compute individual level quality indicators
D3-UC5	Generate feedback report

Table 27: Demonstrator 3 use cases

Figure 4 shows the use case diagram for the demonstrator.

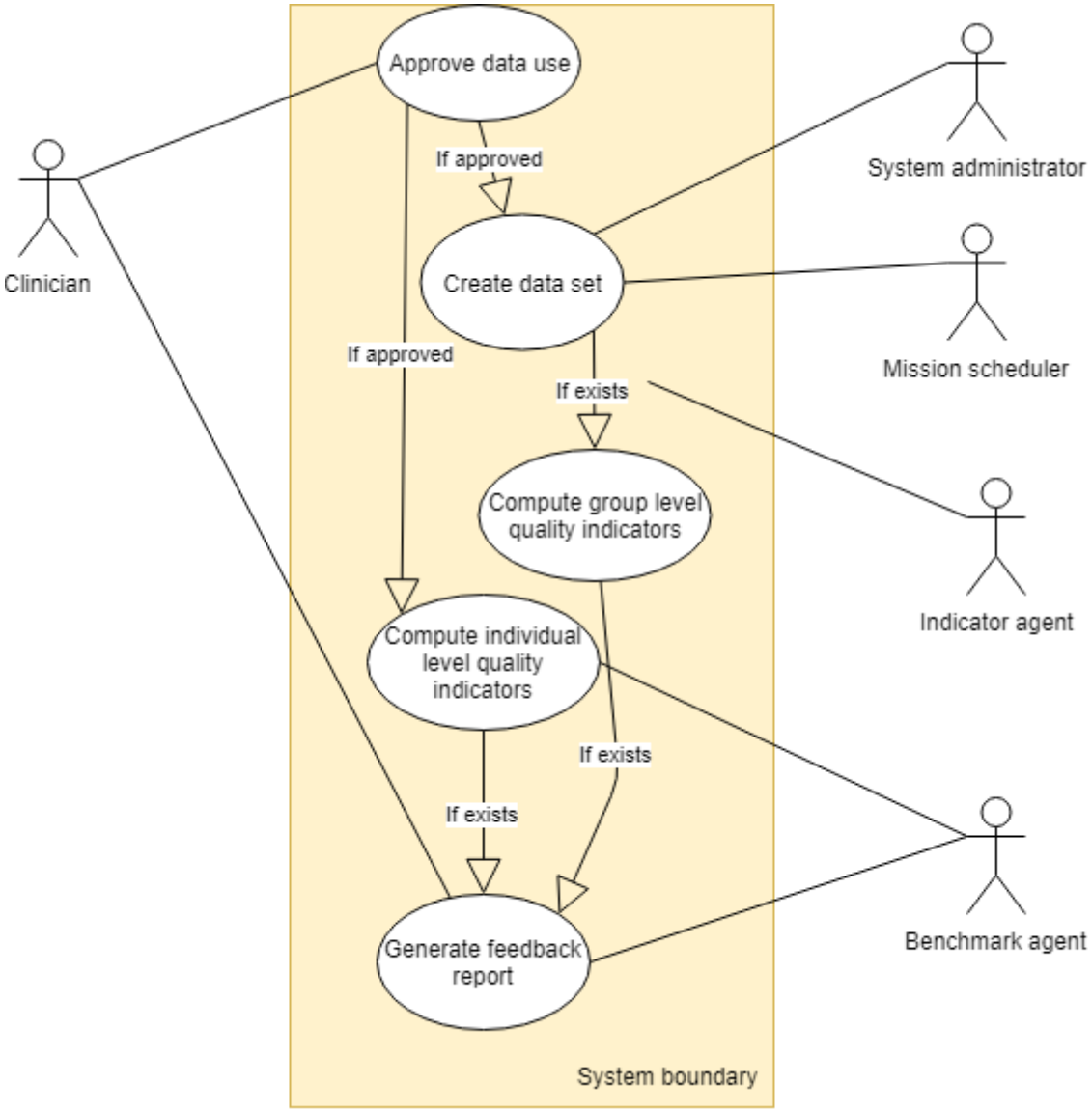

Figure 4: Demonstrator 3 use case diagram

For all the use cases described further, we assume the ASCLEPIOS framework to be up and running.

8.3.1 Approve data use

The aim of the demonstrator is categorized as quality improvement. A Norwegian regulation (26) allows the reuse of health data for quality improvements with the approval of a health institution. In Norway, each GP within a GP office is custodian for his/her patients' data which leads to the need to collect approvals from all GPs in a GP office. Computations for a particular purpose should only be performed on the subset of the data for which approval is granted. Approval of data use may be ad hoc on a case basis similar to current permissions for the reuse of data for research purposes. Table 28 shows the description of the use case.

Attribute	Description
Use case ID	D3-UC1

Use case goal	Get an approval from a GP to the use of his/her patients' data for antibiotics prescriptions benchmarking (quality improvement)
Assumptions & pre-conditions	PKI infrastructure for authenticating the GP is available.
Use case initiation	A request for access to patients' data for antibiotics prescriptions benchmarking is sent to the GP
Use case main scenario	The GP grants permission to the use of his/her patients' data
Use case alternate scenario	The GP withdraws permission to the use of his/her patients' data
Use case failure scenario	• The GP is not able to grant or reject permission for the data reuse
Acceptance criteria	• The system administrator is able to compute on the subset of the EHR data for which he/she has permission • No one is able to compute on the EHR data for which he/she does not have the responsible GP(s) permission

Table 28: "Approve data use" use case description summary

8.3.2 Create dataset

After a data reuse approval is received from the GPs across multiple GP offices, a sub-dataset for one or more indicators is defined where the snow server installed at each of the participating GP offices receive a dataset creation query. The Snow server executes the data definition, and locally creates a sub dataset on which indicators will be computed. A typical sub-dataset contains GP identifiers, patient identifiers, diagnosis codes, antibiotic prescriptions, and an encounter identifier. Table 29 shows the description of the use case.

Attribute	**Description**
Use case ID	D3-UC2
Use case goal	Create a sub-dataset containing the GP's identity, patient identity, diagnosis codes, antibiotics prescriptions, and an encounter identifier
Assumptions & pre-conditions	• GPs working in three or more participating GP offices have approved the use of EHR data for antibiotics prescriptions benchmarking (D3-UC1)
Use case initiation	The mission scheduler initiates the execution of a dataset creation query
Use case main scenario	An agent running on the coordinator server broadcasts a dataset creation query to participating GP offices. An agent running on the Snow server executes the query against the local EHR data, and creates a dataset containing a GP identifiers, patient identifiers, diagnosis codes, antibiotic prescriptions, and encounter identifier is created and locally stored
Use case alternate scenario	None

Use case failure scenario	• EHR data are not available • All GPs in a GP office have withdrawn permission to the data reuse • Network partitioning/network errors
Acceptance criteria	• A dataset containing GP identifiers, patient identifiers, diagnosis codes, antibiotic prescriptions, and encounter identifier is created at each of the GP offices that provided permission • Only approved software components can process the dataset

Table 29: "Create dataset" use case description summary

8.3.3 Compute group level quality indicators

The demonstrator computes the group level performance of peers across multiple GP offices. To that end, the tool contains an agent, called indicator agent, running at the coordinator. The agent computes group level quality indicators from the datasets distributed across multiple GP offices. A group level indicator can be the average indicator or the personal indicators distribution of all the GPs. Let us consider an indicator the percentage of acute upper respiratory infection patients treated with antibiotics in Tromsø, Norway. Emnet will compute the total number patients and the number of patients treated with antibiotics across all the GP offices. Then, the indicator agent computes the indicators from the aggregated results of Emnet. The aggregated quality indicators are accessible to anybody through a web service, called *"indicator web service"*. Table 30 shows the description of the use case.

Attribute	**Description**
Use case ID	D3-UC3
Use case goal	Compute group level quality indicators for antibiotics prescriptions
Assumptions & pre-conditions	• Use case D3-UC1 • Use case D3-UC2
Use case initiation	Mission scheduler initiates the computation of group level quality indicators
Use case main scenario	The indicator agent receives the group level quality indicators computation mission and executes the necessary distributed computations using Emnet. Then, the indicator agent computes the indicators based on the results of Emnet.
Use case alternate scenario	None
Use case failure scenario	• The Snow server in a GP office is unavailable • Network partitioning • Data reuse permission is withdrawn
Acceptance criteria	Group level indicators are computed

Table 30: "Compute group level quality indicators" use case description summary

8.3.4 Compute individual level quality indicators

The quality indicators of a GP are computed on the dataset created at respective GP office, and indicators are stored locally at the GP office. GP indicators cannot be transferred outside the GP office. Table 31 shows the description of the use case.

Attribute	Description
Use case ID	D3-UC4
Use case goal	Compute indicators for the antibiotics prescriptions of a GP
Assumptions & pre-conditions	• Use case D3-UC1 • Use case D3-UC2
Use case initiation	Mission scheduler initiates the computation of individual level quality indicators
Use case main scenario	The benchmarking agent receives a group level quality indicators computation mission and locally computes the quality indicators of a GP based on his/her data
Use case alternate scenario	None
Use case failure scenario	• The Snow server(s) in one or more GP offices become unavailable • Network partitioning • Server error • Hardware failure • Data reuse permission is withdrawn
Acceptance criteria	• Indicators of a GP are computed • GP indicators are securely stored

Table 31: "Compute individual level quality indicators" use case description summary

8.3.5 Generate feedback report

Once the necessary group level and GP antibiotics prescriptions indicators are available, a feedback for the GP can be generated. Then, the benchmark agent generates a report and sends notification emails about the availability of a feedback report to the GP. The feedback report is accessible through a secure web client. Table 32 shows the description of the use case.

Attribute	Description
Use case ID	D3-UC5
Use case goal	Produce a report containing the antibiotics prescription indicators for a GP compared to the indicators of group of GPs

Assumptions & pre-conditions	• Use case D3-UC1 • Use case D3-UC2 • GP's contact information for sending a report notification is available
Use case initiation	Benchmark agent initiates the generation of a report
Use case main scenario	Benchmark agent uses the indicators of a GP and group level indicators of all GPs to generate a feedback report. Then, it sends a notification to the GP. The GP accesses feedback report through a secure client.
Use case alternate scenario	None
Use case failure scenario	• Snow server error • Network error • Hardware failure
Acceptance criteria	Feedback report is available for a GP

Table 32: "Generate feedback report" use case description summary

8.4 Components/mechanisms of ASCLEPIOS framework involved in the demonstrator

Here are the components/mechanisms of the ASCLEPIOS framework required by the demonstrator:

- *Encrypted storage of health data*.
- *Use of Cloud/HPC resources*.
- *Data access control using SSE and ABE*.
- *Privacy-preserving analytics of health data stored at a GP office using FE*; uses FE for enabling privacy-preserving computations on data distributed across multiple GP offices.
- *Remote attestation* to check the integrity of the Snow servers installed at GP offices.
- *Increase GDPR and Security Awareness* to enable explicit consent from clinicians and patients to reuse their EHR data.

8.5 Demonstrator security requirements

The security and privacy requirements (specified in *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements*) applicable for all use cases of Demonstrator 3 are summarized in Table 33.

Use case	**Requirement**	**ASCLEPIOS functionalities**
all	***S-CIN1*** *A cloud-based e-health framework SHALL allow administrators to define certain security profiles (such as specific software bundles and specific configurations) to be considered as trusted.*	Cloud provider integrity

all	***S-CIN2*** *A cloud-based e-health framework SHALL support integrity verification of virtualization servers.*	Cloud provider integrity
all	***S-CIN4*** *A cloud-based e-health framework SHALL support executing security-sensitive workloads in a dedicated (isolated) secure computing environment.*	Cloud provider integrity
all	***S-NET1*** *A cloud-based e-health framework SHALL provide support mechanisms to authenticate as well as to verify and protect the integrity of software components in the network infrastructure.*	Cloud provider integrity
all	***S-NET2*** *A cloud-based e-health framework SHALL support strong authentication and robust traceability of network infrastructure management.*	Master Authority authentication, cloud provider integrity
all	***S-NET3*** *A cloud-based e-health framework SHALL support deployment of secure communication channels for in-transit protection of data between endpoints both on the same network infrastructure, as well as on network infrastructures of different institutions.*	Encryption and cloud provider integrity
all	***S-ACC3*** *A cloud-based e-health framework SHALL be adaptable to organization-specific key management schemes to support diverse authentication models and secure communication protocols.*	Searchable encryption, ABE schemes and key tray
all	***S-ACC4*** *A cloud-based e-health framework SHALL use a key generation algorithm that guarantees the generated keys are secure (long enough and generated using enough randomness).*	Searchable encryption and ABE schemes
all	***H-AUD3*** *A cloud-based e-health framework SHALL protect the integrity of the audit trail.*	Encryption and cloud provider integrity
all	***H-AUD4*** *A cloud-based e-health framework SHALL enable authorized access to the audit trail.*	Encryption and cloud provider integrity
all	***H-AUD5*** *A cloud-based e-health framework SHALL ensure the audit trail maintains records of disclosures of the audit trail itself.*	Encryption and cloud provider integrity
D3-UC2, D3-UC3, D3-UC4, D3-UC5	***Data about patient and health providers should not leave the institution*** ***H-DRT1*** *A cloud-based e-health framework SHALL enable authorized users to retrieve partial or complete health information based on specific criteria.* ***H-DST2*** *A cloud-based e-health framework SHALL encrypt personal data during storage.* ***H-DST3*** *A cloud-based e-health framework SHALL secure encryption/decryption keys.*	ABE schemes and Master Authority authentication Searchable encryption and ABE schemes Searchable encryption and ABE schemes

	S-ACC1 *A cloud-based e-health framework SHALL support access control mechanisms to provide users access to the services that they have been specifically authorized to use.* ***S-ACC2*** *A cloud-based e-health framework SHALL support secure authentication methods to control access by remote users.* ***H-AUD1*** *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	Searchable encryption and ABE schemes ABE schemes and Master Authority authentication Cloud provider integrity
D3-UC2, D3-UC3, D3-UC5	***Only aggregated data can leave the institution*** ***H-DRT1*** *A cloud-based e-health framework SHALL enable authorized users to retrieve partial or complete health information based on specific criteria.* ***H-DAR4*** *A cloud-based e-health framework SHALL support the implementation of privacy-preserving distributed data mining.*	ABE schemes and Master Authority authentication Functional encryption
all	***Data controller has given permission for data usage*** ***H-DAR1*** *A cloud-based e-health framework SHALL support authorized analyses of health data.* ***S-DAM2*** *A cloud-based e-health framework SHALL provide mechanisms for secure storage and sharing of data.* ***S-ACC5*** *A cloud-based e-health framework SHALL support efficient and effective key revocation.* ***H-UID1*** *A cloud-based e-health framework SHALL uniquely identify the data subject to whom a record entry belongs to.* ***H-UID2*** *A cloud-based e-health framework SHALL uniquely identify each resource, including health record entries.* ***H-ACC5*** *A cloud-based e-health framework SHALL NOT enable the processing, except for storage, of personal data that has been restricted.* ***H-DST5*** *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject chooses to exercise the right to be forgotten.* ***H-AUD1*** *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	Access control Searchable encryption Searchable encryption and ABE schemes Registration authority Registration authority ABE schemes and Master Authority authentication Access control Cloud provider integrity

D3-UC1	***Patients have a possibility to know for what purposes their data have been used with possibility to consent or withdraw consent***	
	H-ACC5 *A cloud-based e-health framework SHALL NOT enable the processing, except for storage, of personal data that has been restricted.*	ABE schemes and Master Authority authentication
	H-DST5 *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject chooses to exercise the right to be forgotten.*	Access control
	H-AUD1 *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	Cloud provider integrity

Table 33: Summary of the security and privacy requirements for Demonstrator 3

8.6 Data requirements

In addition to the requirements mentioned above, the data requirements for the use cases of Demonstrator 3 include:

- *EHR data are stored in a standardized interoperable format* (D3-UC2, D3-UC3, D3-UC4). The demonstrator assumes the data at each GP office are store stored in a relational database with a specific data model.
- *Metadata are stored in DDI format* (27) (D3-UC2).
- *Identity management system for GPs and patients is available* (D3-UC3, D3-UC4).

8.7 Testbed

The testbed for the demonstrator will consist of a coordinator server and three Snow servers simulating GP offices (at least three health institutions are required for the computation of a group level quality indicators). The GP offices will be populated with encrypted test data. On top of this setup, we will deploy the software components of the demonstrator described in the use cases, then execute each of the use cases.

9 ASCLEPIOS Requirements

Based on the requirements identified by the demonstrators, Table 34 has been created to classify the requirements in *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements* as of high, medium, and low priority. Requirements needed by the demonstrators will have higher priority than requirements that will not be used by any of the demonstrators. This list will make the prioritization for implementations in *WP2: Operations on Encrypted Health Data and Privacy-Preserving Health Data-Driven Analytics*, *WP3: Access Policies and Enforcement Middleware*, and *WP4: Isolated Execution and Medical Devices Security* to be inline with the demonstrators' requirements.

Requirement	Demonstrator	Priority
S-ACC1 *A cloud-based e-health framework SHALL support access control mechanisms to provide users access to the services that they have been specifically authorized to use.*	1, 2, 3	high
S-ACC2 *A cloud-based e-health framework SHALL support secure authentication methods to control access by remote users.*	1, 2, 3	high
S-ACC3 *A cloud-based e-health framework SHALL be adaptable to organization-specific key management schemes to support diverse authentication models and secure communication protocols.*	1, 2, 3	high
S-ACC4 *A cloud-based e-health framework SHALL use a key generation algorithm that guarantees the generated keys are secure (long enough and generated using enough randomness).*	1, 2, 3	high
S-ACC5 *A cloud-based e-health framework SHALL support efficient and effective key revocation.*	1, 2, 3	high
S-CIN1 *A cloud-based e-health framework SHALL allow administrators to define certain security profiles (such as specific software bundles and specific configurations) to be considered as trusted.*	1, 2, 3	high
S-CIN2 *A cloud-based e-health framework SHALL support integrity verification of virtualization servers.*	1, 2, 3	high
S-CIN3 *A cloud-based e-health framework SHALL implement automatic equipment identification based on a hardware root of trust (where applicable).*	1, 2	medium
S-CIN4 *A cloud-based e-health framework SHALL support executing security-sensitive workloads in a dedicated (isolated) secure computing environment.*	1, 2, 3	high
S-DAM1 *A cloud-based e-health framework SHALL provide mechanisms to enforce reliable destruction of workloads and configuration in isolated enclaves.*	1, 2	medium
S-DAM2 *A cloud-based e-health framework SHALL provide mechanisms for secure storage and sharing of data.*	1, 2, 3	high
S-NET1 *A cloud-based e-health framework SHALL provide support mechanisms to authenticate as well as to verify and protect the integrity of software components in the network infrastructure.*	1, 2, 3	high

S-NET2 *A cloud-based e-health framework SHALL support strong authentication and robust traceability of network infrastructure management.*	1, 2, 3	high
S-NET3 *A cloud-based e-health framework SHALL support deployment of secure communication channels for in-transit protection of data between endpoints both on the same network infrastructure, as well as on network infrastructures of different institutions.*	1, 2, 3	high
H-UID1 *A cloud-based e-health framework SHALL uniquely identify the data subject to whom a record entry belongs to.*	1, 2, 3	high
H-UID2 *A cloud-based e-health framework SHALL uniquely identify each resource, including health record entries.*	1, 2, 3	high
H-CIS1 *A cloud-based e-health framework SHALL be able to represent links between resources, such as health records entries and external resources (e.g., images).*	2	low
H-DST1 *A cloud-based e-health framework SHALL store information committed by authorized users.*	1, 2, 3	high
H-DST2 *A cloud-based e-health framework SHALL encrypt personal data during storage.*	1, 2, 3	high
H-DST3 *A cloud-based e-health framework SHALL secure encryption/decryption keys.*	1, 2, 3	high
H-DST4 *A cloud-based e-health framework SHALL ensure minimal storage overhead.*	2	low
H-DST5 *A cloud-based e-health framework SHALL enable the erasure of personal data with undue delay if the data subject chooses to exercise the right to be forgotten.*	1, 2, 3	high
H-DRT1 *A cloud-based e-health framework SHALL enable authorized users to retrieve partial or complete health information based on specific criteria.*	1, 2, 3	high
H-DRT2 *A cloud-based e-health framework SHALL enable an authorized user accessing a health record entry that contains one or more links to be able to retrieve the referenced resources, such as health record entries and external resources (e.g., images).*	1, 2	medium
H-DRT3 *A cloud-based e-health framework SHALL enable a data subject or legal representative to obtain access to the personal data concerning the data subject.*	1, 2	medium
H-DRT4 *A cloud-based e-health framework SHALL enable sharing of personal data with another [healthcare] organization* based on the appropriate legal ground.*	1, 2	medium
H-DRT5 *A cloud-based e-health framework SHALL enable receiving personal data from another [healthcare] organization* based on the appropriate legal ground.*	1, 2	medium

H-DAR1 *A cloud-based e-health framework SHALL support authorized analyses of health data.*	1, 2, 3	high
H-DAR2 *A cloud-based e-health framework SHALL enable authorized users to transform personal data into pseudonymized data.*	1, 2	medium
H-DAR3 *A cloud-based e-health framework SHALL enable authorized users to transform personal data to anonymized data.*	1, 2	medium
H-DAR4 *A cloud-based e-health framework SHALL support the implementation of privacy-preserving distributed data mining.*	1, 2, 3	high
H-ACC1 *A cloud-based e-health framework SHALL enable the representation of technically supported access policies that control access to health records.*	1, 2	medium
H-ACC2 A *cloud-based e-health framework SHALL be able to associate a health record entry or a group of health record entries with policies that apply to them.*	1, 2	medium
H-ACC3 *A cloud-based e-health framework SHALL enable processing of health records based on the applicable technically supported access policies.*	1, 2	medium
H-ACC4 *A cloud-based e-health framework SHALL NOT require the presence of explicit consent in order to process personal data, if there are other legal grounds that permit the processing.*	1	low
H-ACC5 *A cloud-based e-health framework SHALL NOT enable the processing, except for storage, of personal data that has been restricted.*	1, 2, 3	high
H-AVA1 *A cloud-based e-health framework SHALL NOT impede the availability of data for lawful processing.*	1	medium
H-AUD1 *A cloud-based e-health framework SHALL maintain an audit trail of personal data processing.*	1, 2, 3	high
H-AUD2 *A cloud-based e-health framework SHALL specifically identify access that has overridden policies (e.g., in a medical emergency situation).*	1, 3	medium
H-AUD3 *A cloud-based e-health framework SHALL protect the integrity of the audit trail.*	1, 2, 3	high
H-AUD4 *A cloud-based e-health framework SHALL enable authorized access to the audit trail.*	1, 3	medium
H-AUD5 *A cloud-based e-health framework SHALL ensure the audit trail maintains records of disclosures of the audit trail itself.*	1, 2, 3	high

Table 34: A prioritized list of security, privacy and data requirements

* [healthcare] is in close brace to indicate it as a possible organization type; it is not meant to limit the data transfer to/from healthcare organizations.

10 Cryprographic Primitives

In this section, we present the cryptographic primitives needed to describe the reference architecture. However, since we are still in an ealy phase of the project, other primitives may also be used. In this case, they will be thoroughly described in the corresponding deliverable.

10.1 Symmetric Key Encryption (SKE)

Symmetric encryption is a method of cryptography where a single key is responsible for encrypting and decrypting data. The involved parties share that key and they can use it to decrypt or encrypt any messages they want. The most common algorithms used for symmetric cryptography include:

- Triple Des, which applies the DES algorithm three times with different keys, and
- Advanced Encryption Standard (AES) recommended by the US National Institute of Standards and Techonology

Symmetric key-ciphers, or the algorithms used to perform encryption and decryption, appeal to organizations because they are inexpensive despite the level of protection they can afford.

However, symmetric encryption is not perfect. Keys in this cryptographic method live on forever. This means organizations must invest in logging and auditing of the keys over their lifecycle. Moreover, if a symmetric key is lost, organizations cannot recall it. Instead, they must encrypt and decrypt data with a different key once they recover their data in an unencrypted form. A symmetric key encryption scheme is defined as follows (28):

Definition (SKE). A symmetric (secret) key encryption E scheme supporting a message domain $\mathcal{M}$ consists of the following polynomial time algorithms:

1. **E.KeyGen(1^{λ}).** A Key generation algorithm that takes as input a security parameter λ and outputs a key K from the key space $\mathcal{K}$,
2. **E.Enc(K, m).** An encryption algorithm that takes as input a key K and a message $m \in \mathcal{M}$ and outputs a ciphertext c,
3. **E.Dec(K, c).** A decryption algorithm that takes as input a key K and a ciphertext c and outputs m.

Correctness: A symmetric key encryption scheme E is correct if and only if, for all λ and all $m \in \mathcal{M}$:

$$\Pr[E.Dec(K, E.Enc(K, m)) \neq m | K \leftarrow E.KeyGen(1^{\lambda}) = negl(\lambda)$$

An encryption scheme provides data confidentiality. So, it should prevent an adversary from learning which message is encrypted in a ciphertext. The security of E is formally defined by the following security game:

Definition (IND-CPA security of E). Security is depicted by the following games between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

1. The challenger runs the E.KeyGen algorithm to obtain a key K from the key space $\mathcal{K}$,
2. The challenger also choses a random bit $b \in \{0, 1\}$,
3. Whenever the adversary provides a pair of messages (m_0, m_1) of her choice, the challenger replies with E.Enc(K, m_b),
4. The adversary finally outputs its guess b'.

The advantage of adversary in the above game is:

$$Adv(\mathcal{A}) \coloneqq \Pr[b' = b] - \frac{1}{2}$$

A symmetric key encryption scheme E is said to have indistinguishability security under chosen plaintext attack if there is no polynomial time adversary $\mathcal{A}$ which can win the above game with probability non-negligible in λ .

10.2 Public Key Encryption (PKE)

Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related, but not identical, keys - a public key and a private key. Unlike symmetric key algorithms, that rely on one key to both encrypt and decrypt, here each key performs a unique function. The public key is used to encrypt, and the private key is used to decrypt.

It is computationally infeasible to compute the private key based on the public key. Because of this, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, ensuring only the owners of the private keys can decrypt content and create digital signatures.

Since public keys need to be shared but are too big to be easily remembered, they are stored on digital certificates for secure transport and sharing. Since private keys are not shared, they are simply stored in the software or operating system of the user, or on hardware (e.g., USB token, hardware security module) containing drivers that allow it to be used with the user's software or operating system. A public key encryption scheme is defined as follows (28):

Definition (PKE). A PKE is a generalization of symmetric key encryption where anyone with the public key of the receiver can send encrypted messages to the receiver. A PKE scheme supporting a message domain M consists of the following algorithms:

1. **PKE.KeyGen(1^{λ}).** A key generation algorithm that takes as input a security parameter λ and outputs a public/secret key pair (pk, sk),
2. **PKE.Enc(pk, m).** An encryption algorithm that takes as input a public key pk and a message $m \in \mathcal{M}$ and ouputs a ciphertext c, which is an encryption of m under sk,
3. **PKE.Dec(sk, c).** A decryption algorithm that takes as input a secret key sk and a ciphertext c and outputs the decrypted message m.

Correctness: A public key encryption scheme PKE is correct if and only if for all λ and $m \in \mathcal{M}$

$$\Pr\left[PKE.Dec\left(sk, PKE.Enc(pk, m)\right) \neq m \middle| (pk, sk) \leftarrow PKE.KeyGen\left(1^{\lambda}\right)\right] = negl(\lambda)$$

A PKE scheme provides confidentiality to the encrypted message. The security of PKE is formally defined by the following security game:

Definition (IND-CCA2 security of PKE). Consider the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

1. $\mathcal{C}$ runs the PKE.KeyGen algorithm to obtain a key pair (pk, sk) and gives pk to the adversary,
2. $\mathcal{A}$ provides adapitively chosen ciphertexts c and gets back PKE.Dec(sk, c),
3. $\mathcal{A}$ provides two messages m_0, m_1 to $\mathcal{C}$,

4. $\mathcal{C}$ then runs $c^* =$ PKE.Enc(pk, m_b), for some $b \leftarrow \{0,1\}$. $\mathcal{C}$ provides c^* to $\mathcal{A}$,
5. $\mathcal{A}$ continues to provide adaptively chosen c in order to get back PKE.Dec(sk, c),
6. $\mathcal{A}$ outputs its guess b'.

The advantage of the adversary $\mathcal{A}$ in the above game is:

$$Adv(\mathcal{A}) := \Pr[b' = b] - \frac{1}{2}$$

A public key encryption scheme is said to have indistinguishability security under adaptively chosen ciphertext attack if there is no polynomial time adversary $\mathcal{A}$ which can win the above game with probability non-negligible in λ .

10.3 Digital Signatures

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. A digital signature is the digital equivalent of a handwritten signature or stamped seal, a digital signature offers far more inherent security, and it is intended to solve the problem of tampering and impersonation in digital communications.

Digital signatures can provide the added assurances of evidence of origin, identity and status of an electronic document, transaction or message and can acknowledge informed consent by the signer.

In many countries, signatures are considered legally binding in the same way as traditional document signatures. A digital signature scheme is defined as follows (28):

Definition (Signature Scheme). A digital signature scheme S supporting a message domain M consists of the following polynomial time algorithms:

1. **S.KeyGen(1^λ).** A key generation algorithm that takes as input a security parameter λ and outputs a signing key sk and a verification key vk,
2. **S.Sign(sk, m).** A signing algorithm that takes as input a signing key sk and a message $m \in$ M and outputs the signature σ,
3. **S.Verify(vk, σ, m).** A verification algorithm that takes as input a verification key vk, a signature σ and a message m and outputs either 0 or 1.

The first two algorithms are probabilistic whereas the verification algorithm is deterministic.

Correctness: A signature scheme S is correct if for all $m \in$ M,

$$\Pr[\mathrm{S.Verify}(vk, S.Sign(sk, m), m) = 0 | (sk, vk \leftarrow S.KeyGen(1^\lambda)]$$

Signatures provide authenticity. So, an adversary without the signing key should not be able to generate a valid signature. The security of S is formally defined by the following security game:

Definition (EUF-CMA security of S). Consider the following game between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

1. The challenger runs the S.KeyGen algorithm to obtain the signing/verification key pair (sk, vk), and provides the verification key vk to the adversary,

2. Initialize query = { },

3. Now, whenever the adversary provides a query with a message m, the challenger replies with S.Sign(sk, m). Also, query = query ∪ m,
4. Finally, the adversary outputs a forged signature σ* corresponding to a message m.

The advantage of $\mathcal{A}$ in the above security game is:

$$Adv(\mathcal{A}) = \Pr[S.Verify(vk, \sigma^*, m^*) = 1 | m^* \notin query]$$

A signature scheme S is said to be existentially unforgeable under chosen message attack if there is no polynomial time adversary which can win the above game with probability non-negligible in λ.

10.4 Cryptographic Hash Functions

Informally, a cryptographic hash function is employed to produce a short descriptor of a message. A descriptor is analogous to a fingerprint for human identification (see Figure 5). We proceed with the definition of a hast function as well as their properties as defined in (30).

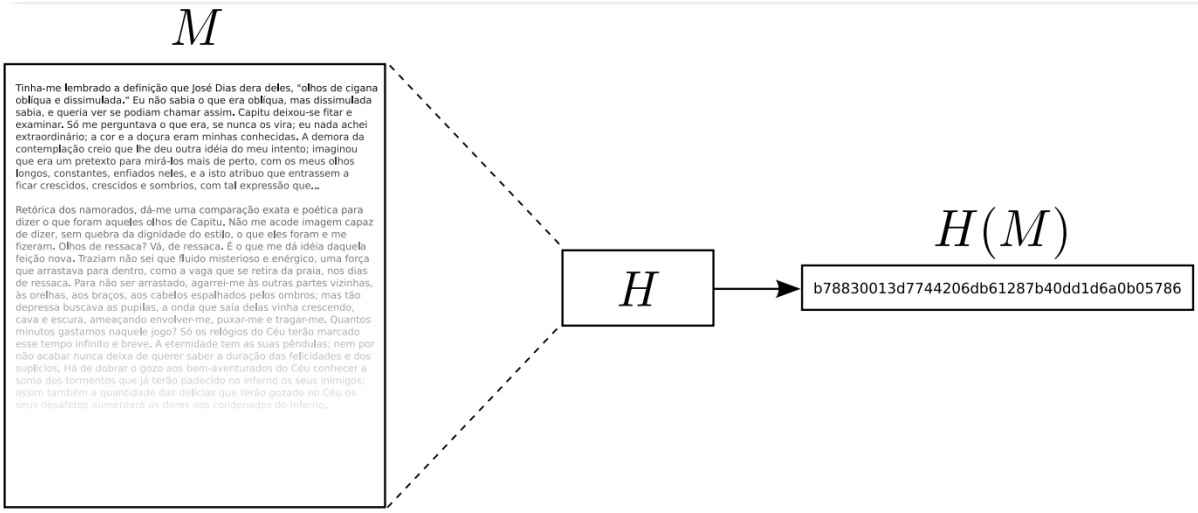

Figure 5: Hash Function

Definition (Cryptographic hash function). A cryptographic hash function maps messages from a set $\mathcal{X}$ to hash valuer or authenticators in a set $\mathcal{Y}$. In this first case, it is denoted by $H: \mathcal{X} \to \mathcal{Y}$. In the second, it is parameterized by a key $K \in \mathcal{K}$ and represented by $H: \mathcal{X} \to \mathcal{Y}$. If $\mathcal{X}$ is finite, H is also called a **compression function.**

Hash functions have many different applications. We list some of them below:

- Password Storage ($store\ H(s)$ instead of s),
- Key derivation ($k = h(g^{xy} mod\ p)$, $k_i = H(k_{i-1})$),
- Integrity verification ($y = H(x)$),
- Digital Singatures (sign $h(m)$ instead of just m),
- Message Authentication Codes (MACs) ($y = H_K(x)$).

10.4.1 Properties of Hash Functions

In this section, we list the most important properties of hash functions.

- **Preimage resistance.** Given hash y, it should be computationally infeasible to find x such that $y = H(x)$,
- **Second preimage resistance.** Given hash y and a message x such that $y = H(x)$, it should be computationally infeasible to find $x' \neq x$ such that $H(x') = H(x) = y$,
- **Collision resistance.** It should be computationally infeasible to find x, x' such that $H(x) = H(x')$.

10.5 Pseudorandomness

A pseudorandom process predicts outcomes given information which is typically difficult to acquire. In general, a truly random process generates unpredictable outcomes, that is for any single event no particular outcome can be predicted in advance. For example, consider an unbiased coin which on any given flip, the outcome is either "*heads"* or *"tails".* On a single flip no outcome is certain. Recording a number of flips in a logbook provides a sequence of pseudorandom outcomes; in possession on the logbook, each outcome is known for certain. However, a person without the logbook sees only a random string of *"heads"* or *"tails".*

Randomness is therefore a condition which holds a sequence *relative* to the information available to the predictor. Pseudorandomness indicates that sufficient information to predict the next outcome may be acquired by the predictor.

In cryptography, random values are often required. The intuition behind this, is to make a message as hard to break as possible. Pseudorandom sequences are deterministic and reproducible; all that is required in order to discover and reproduce a pseudorandom sequence is the algorithm used to generate it and the initial seed.

10.5.1 Definitions

A pseudorandom generator (PRG) for a class of statistical tests is a deterministic procedure that maps a random seed to a longer pseudorandom string such that no statistical test in the class can distinguish between the output of the generator and the uniform distribution. The random seed is typically a short binary string drawn from the uniform distribution. It is not known if secure pseudorandom generators exist. Proving that they exist is equivalent to proving that $P \neq NP$, which is widely believed but a famously open problem. The existence of secure pseudorandom generators is widely believed as well and they are necessary for many cryptographic applications.

Definition (Statistical Test). A statistical test is a class of functions $\mathcal{A} = \{ \mathcal{A}: \{0,1\}^n \rightarrow \{0,1\}^* \}$. These functions are the ones that the pseudorandom generator will try to "fool" and they are usually algorithms. Most of the times, these statistical tests are called adversaries of distinguishers.

Definition (PRG). A function $\mathcal{G}: \{0,1\}^{\ell} \rightarrow \{0,1\}^n$, where $l \leq n$
Is a pseudorandom generator against $\mathcal{A}$ with bias ϵ if for every $A \in \mathcal{A}$, the statistical distance between the distributions $A(G(U_{\ell}))$ and $A(U_n)$ is at most ϵ, where U_k is the uniform distribution on $\{0,1\}^k$.

The quantity ℓ is called the *seed length* and the quantity $n - \ell$ is called the *stretch* of the pseudorandom generator.

A pseudorandom generator against a family of adversaries$(\mathcal{A}_n)_{n\in\mathbb{N}}$ with bias $\epsilon(n)$ is a family of pseudorandom generators $(\mathcal{G}_n)_{n\in\mathbb{N}}$, where $\mathcal{G}_n:\{0,1\}^{\ell(n)} \to \{0,1\}^n$ is a pseudorandom generator against $\mathcal{A}_n$ with bias $\epsilon(n)$ and seed length $\ell(n)$.

Apart, from the PRGs we also need to look at pseudorandom number generators (PRNG). A requirement for a cryptographic PRNG is that an adversary, who does not know the seed, has only negligible advantage in distinguishing the generator's output sequence from a random sequence. In other words, a cryptographically secure PRNG must pass all the statistical tests that are restricted to polynomial time in the size of the seed. In general, years of review may be required before an algorithm can be certified as a cryptographically secure PRNG.

Some examples of cryptographically secure PRNGs are the following:

- Stream ciphers,
- Block ciphers running in counter,
- Microsoft's Cryptographic Application Programming Interface function (CryptGenRandom).

Most PRNG algorithms produce sequences which are uniformly distributed by any of several tests. It is an open question whether there is any way to distinguish the output of a high-quality PRNG from a truly random sequence. In this setting, the adversary knows that either the known PRNG algorithm was used – but not the initialization state- or a truly random algorithm was used and has to distinguish between the two. The security of most cryptographic algorithms and protocols using PRNGs is based on the assumption that it is infeasible to distinguish between the use of a cryptographically secure PRNG and a truly random sequence. The simplest examples of this dependency are stream ciphers, which they often *xor* the plaintext of a message with the output of a PRNG in order to produce a ciphertext.

Definition (PRNG). A pseudorandom number generator is a function that, once initialized with some random value (the seed) outputs a sequence that appears random, in the sense that an observer who does not know the value of the seed cannot distinguish between the output of the PRNG and the of truly random bit generator.

10.6 Ciphertext-policy Attribute-based Encryption (CP-ABE)

Attribute-based encryption (ABE) is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key issue is, that someone should only be able to decrypt a ciphertext if the person holds a key for *"matching attributes"* where user keys are always issued by some trusted party.

In ciphertext-policy attribute-based encryption (CP-ABE) a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunction or, disjunctions For instance, let us assume that the universe of

attributes is defined to be {A,B,C,D} and user 1 receives a key to attributes {A,B} and user 2 to attribute {D}. If a ciphertext is encrypted with respect to the policy (A∧C)∨D ((A AND C) OR D), then user 2 will be able to decrypt, while user 1 will not be able to decrypt.

Let us for example define two different polices, $\mathcal{P} = \{engineer \wedge female\}$ and $\mathcal{Q} = \{nurse \vee male\}$. Moreover, let Lois be and engineer and Peter be a lawyer. Then the following holds (Figure 6):

User	Attributes	Policy	Decryption
Lois	Female, Engineer	P	✓
Lois	Female, Engineer	Q	✗
Peter	Male, Lawyer	P	✗
Peter	Male, Lawyer	Q	✓

Figure 6: CP-ABE example

CP-ABE allows to realize implicit authorization, i.e., authorization is included into the encrypted data and only people who satisfy the associated policy can decrypt data. Another nice feature is that users can obtain their private keys after data has been encrypted with respect to policies. So, data can be encrypted without knowledge of the actual set of users that will be able to decrypt, but only specifying the policy which allows to decrypt. Any future users that will be given a key with respect to attributes such that the policy can be satisfied will then be able to decrypt the data.

In order to provide a concrete and reliable solution for the problem described in the previous section, we need to build a protocol through which newly encrypted data will not be decryptable by a user if her access has been revoked. Additionally, we want to allow users with certain access rights to be able to search directly over encrypted data. To this end, we will be using a CP-ABE scheme. In a CP-ABE scheme every secret key (e.g. user key) is generated based on a public and a private key as well as on a concrete *list of attributes A.* Then, every ciphertext is associated with a policy P. Decryption will only be possible if $P(A) = True$ – if the attributes on a key satisfy the policy on the ciphertext. From now on we will refer to the space of attributes as $\Omega = \{\alpha_1, \dots, \alpha_n\}$, while the space of policies will be denoted as $\mathcal{P} = \{p_1, \dots p_n\}$.

We now proceed with the definition of a CP-ABE scheme (30).

Definition (Ciphertext-policy ABE). A CP-ABE scheme is a tuple of the following four algorithms:

1. **CPABE.Setup.** A probabilistic algorithm that takes as input a security parameter λ and outputs a master public key MPK and a master secret key MSK. We denote this by $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}(1^{\lambda})$,

2. **CPABE.Gen.** A probabilistic algorithm that takes as input a master secret key, a set of attributes Ω and the unique identifier of a user and outputs a secret key which is bind both to the corresponding list of attributes and the user. We denote this by $sk_{\{A,u_i\}} \leftarrow \mathrm{Gen}(\mathrm{MSK}, \mathrm{A}, u_i)$,

3. **CPABE.Enc.** A probablistic algorithm that takes as input a master public key, a message m and a policy $P \in \mathcal{P}$. After a proper run, the algorithm outputs a ciphertext c_p which is associated to the policy P. We denote this by $c_p \leftarrow \mathrm{Enc}(\mathrm{MPK}, m, P)$,

4. **CPABE.Dec.** A deterministic algorithm that takes as input a user's secret key and a ciphertext and outputs the original message m if the set of attributes A that are associated with the underlying secret key satisfies the policy P that is associated with c_p. We denote this by $\mathrm{Dec}(sk_{\{A,u_i\}}, c_p) \rightarrow m$.

We want the CP-ABE scheme to be secure against chosen plaintext attacks. We denote this by CPA-security.

Definition (CPA security of CP-ABE). Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ an adversary that tries to break the scheme. The CPA security of the CP-ABE scheme is defined as follows:

1. **Setup.** The challenger runs CPABE.Setup and gives MPK to the adversary,

2. **Phase 1.** The adversary makes private key queries for discrete set of attributes $A_1, \dots, A_n$,

3. **Challenge.** The adversary submits two messages of equal length M_0, M_1 to the challenger. Moreover, the adversary submits a set of attributes A_i such that $i \notin [1, n]$. The challenger flips a coin b, where $b \in \{0, 1\}$ and encrypts, M_b under MPK. The resulted ciphertext c is returned to the adversary,

4. **Phase 2.** The adversary continues to make private key queries for discrete sets of attributes $A_{\{n+1\}}, \dots A_m$with the restricition $i \notin [n+1, m]$,

5. **Guess.** The adversary outputs a guess b' of b.

The advantage of the adversary $\mathcal{A}$ is:

$$\Pr[b' = b] - \frac{1}{2}$$

We say that the CP-ABE scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above CP-ABE game.

10.7 Dynamic Symmetric Searchable Encryption (DSSE)

Symmetric searchable encryption (SSE) is a promising technique that allows users to search directly over an encrypted database. In a cloud environment, an SSE scheme provides security against internal attacks, since a storage server cannot be regarded as a trusted entity.

SSE is a tokenized encryption technique, in which users generate tokens - search token, add token and delete token- which are then sent to the server. Upon reception, the server will perform the corresponding operation directly on the encrypted data, without decrypting them. The definition of a Dynamical SSE scheme as specified in (30) is presented below:

Definition (Dynamic Index-based SSE). A dynamic index-based symmetric searchable encryption scheme is a tuple of nine polynomial algorithms, SSE = (Gen, Enc, SearchToken, AddToken, DeleteToken, Search, Add, Delete, Dec) such that:

1. **SSE.Gen.** A probabilistic key-generation algorithm that takes as input a security parameter λ and outputs a secret key $\boldsymbol{K}$. It is used by the client to generate her secret key,

2. **SSE.Enc.** A probabilistic key generation algorithm that takes as input a secret key $\boldsymbol{K}$ and a collection of files $\boldsymbol{f}$ and outputs an encrypted index $\boldsymbol{\gamma}$ and a sequence of ciphertexts $\boldsymbol{c}$**.** It is used by the client to get ciphertexts corresponding to her files as well as an encrypted index which are then sent to the storage server,

3. **SSE.SearchToken.** A (possibly probabilistic) algorithm that takes as input a secret key K and a keyword w and outputs a search token $\tau_s(w)$. It is used by the client in order to create a search token for some specific keyword. The token is then sent to the search server,

4. **SSE.AddToken.** A (possibly probabilistic) algorithm that takes as input a secret key K and a file f and outputs an add token $\tau_a(f)$ and a ciphertext c_f. It is used by the client in order to create an add token for a new file as well as the encryption of the file which are then sent to the storage server,

5. **SSE.DeleteToken.** A (possibly probabilistic) algorithm that takes as input a secret key K and a file f and outputs a delete token $\tau_d(f)$. It is used by the client in order to create a delete token for some file which is then sent to the storage server,

6. **SSE.Search.** A deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts c and a search token $\tau_s(w)$ and outputs a sequence of file identifiers $\boldsymbol{I_w} \subset \boldsymbol{c}$. This algorithm is used by the storage server upon receiving a search token in order to perform the search over the encrypted data and determine which ciphertexts correspond to the searched keyword and thus should be sent to the client,

7. **SSE.Add.** A deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts $\boldsymbol{c}$, an add token $\tau_a(f)$ and a ciphertext c_f and outputs a new encrypted index γ' and a new sequence of ciphertexts $\boldsymbol{c}'$. This algorithm is used by the storage server upon receiving an add token in order to update the encrypted index and the ciphertext vector to include the data corresponding to the new file,

8. **SSE.Delete.** A deterministic algorithm that takes as input an encrypted index γ, a sequence of ciphertexts $\boldsymbol{c}$ and a delete token $\tau_d(f)$ and outputs a new encrypted index γ' and a new sequence of ciphertexts $\boldsymbol{c}'$. This algorithm is used by the storage server upon receiving a delete token in order to update then encrypted index and the ciphertext vector to delete the data corresponding to the deleted file,

9. **SSE.Dec.** A deterministic algorithm that takes as input a secret key K and a ciphertext c and outputs a file f. It is used by the client to decrypt the ciphertexts that she gets from the storage server.

A dynamic SSE scheme is correct if for all possible security parameters and file collections, and for secret keys, encrypted indexes and ciphertexts created using the respective algorithms and for any sequences of add, delete and search operations handled using the respective algorithms, it holds that the search operation always returns the correct set of indices corresponding to the searched keyword and the returned ciphertexts can be correctly decrypted.

On an intuitive level, a good security notion for searchable encryption would be to require that nothing is leaked to the storage server beyond the outcome of the search (also known as the access pattern). Unfortunately, practical SSE schemes normally leak more information than that. They also leak whether two queries were for the same keyword or not, which is called the search pattern. The search pattern is leaked if the tokens are deterministic, which is the case in the most efficient schemes. Given this, a reasonable definition of security for searchable encryption is requiring that nothing is leaked beyond the access and search patterns. We should mention that some dynamic schemes, also leak information during the add and delete operations.

The leakage functions associated to index creation, search, addition and delete operations are denoted as $\mathcal{L}_I, \mathcal{L}_S.\mathcal{L}_A, \mathcal{L}_D$ respectively. The security is defined using the following security game:

Definition (Dynamic CKA2-Security). Let SSE = (Gen, Enc, SearchToken, AddToken, DeleteToken, Seach, Add, Delete, Dec) be a dynamic index-based symmetric searchable encryption scheme and $\mathcal{L}_I, \mathcal{L}_S.\mathcal{L}_A, \mathcal{L}_D$ be leakage functions: The following experiments are considered:

- $Real_{\mathcal{A}}()$. The secret key K is generated by running $Gen(1^{\lambda})$. The adversary $\mathcal{A}$ choses a file collection $\boldsymbol{f}$ and then receives an encrypted index γ and the ciphertexts $\boldsymbol{c}$ such that $(\gamma, \mathbf{c}) \leftarrow Enc(\mathrm{K}, \boldsymbol{f})$. The adversary $\mathcal{A}$ can make a polynomial number of adaptive queries to get search, add and delete tokens. The tokens are generated using the respective algorithms of SSE (the ciphertext is also generated in the case of an addition) and given to the adversary. Finally, $\mathcal{A}$ outputs a bit b indicating whether she thinks this is the real or the ideal experiment,

- $Ideal_{\{\mathcal{A},\mathcal{S}\}}()$. The adversary $\mathcal{A}$ chooses a file collection $\boldsymbol{f}$**.** The simulator $\boldsymbol{\mathcal{S}}$ only gets $\mathcal{L}_I(\boldsymbol{f})$ and has to simulate an encrypted index γ and ciphertexts $\boldsymbol{c}$ to send to the adversary. The adversary $\mathcal{A}$ is again allowed to make adaptive queries to get search, add and delete tokens, but the simulator has to generate the tokens (and also the ciphertexts in the case of addition) to send to the adversary given only the leakage from $\mathcal{L}_S.\mathcal{L}_A$ or $\mathcal{L}_D$. Finally, $\mathcal{A}$ outputs a bit b indicating whether she thinks this is the real or the ideal experiment,

SSE is $(\mathcal{L}_I, \mathcal{L}_A, \mathcal{L}_S, \mathcal{L}_D)$ - secure against adaptive dynamic chosen-keyword attacks if for all probabilistic polynomial time adversaries $\mathcal{A}$, there exists a probabilistic polynomial time simulator $\boldsymbol{\mathcal{S}}$ such that:

$$|Real_{\mathcal{A}}() = 1 - Ideal_{\{\mathcal{A},\mathcal{S}\}}() = 1| \leq negl().$$

The intuition behind this definition is that if every adversary cannot distinguish whether the encrypted index, ciphertexts and tokens given to her were generated using the real data and the scheme SSE or by a simulator which only gets as input the information specified by the leakage functions, then SSE only leaks the information specified by the leakage functions.

Using this security definition, the leakage of the SSE scheme can be formally defined. As dynamic index-based symmetric searchable encryption schemes should leak as little information as possible, a good example would be: $\mathcal{L}_I$ leaking only the number of files and unique keywords, the identifiers of the files and size of the files, $\mathcal{L}_S$ leaking only the search and access patterns, $\mathcal{L}_A$ leaking only the size and identifier of the added files as well as the updated number of unique keywords and $\mathcal{L}_D$ leaking only the updated number of unique keywords.

10.8 Functional Encryption (FE)

A functional encryption scheme is an encryption scheme that allows to release so-called *"functional decryption"* keys sk_f (indexed by some function f) such that decrypting a ciphertext $c = Encrypt(pk, m)$ under the secret key sk_f, produces as a result $f(m)$ (rather than just m, as would a normal decryption algorithm.) The ability to reveal only partial information $f(m)$ about a message m make functional encryption a very powerful tool. Traditional public key cryptography, corresponds to a system which supports only the identity function $f(m)$. The wider the class of supported functions, the more expressive the associated functional encryption scheme. We proceed with the definition of an FE scheme as defined in (31).

Definition (FE). A functional encryption scheme is a tuple of the four following algorithms:

1. **FE.Setup.** A probabilistic algorithm that takes as input a security parameter λ and outputs a master public/secret key pair $(\mathrm{PP}, \mathrm{msk})$,

2. **FE.Keygen.** A probabilistic algorithm that takes as input the master secret key msk and a key k and outputs a secret key sk_k for k,

3. **FE.Enc.** A probabilistic algorithm that takes as input the public key PP and a message m and outputs a ciphertext c,

4. **FE.Dec.** A probabilistic algorithm that takes as input sk_k, for some k, and a ciphertext c and outputs a part of the ciphertext in plaintext.

Let $\mathcal{F}$ be a functional encryption scheme. We need to define security against an adaptive adversary $\mathcal{A}$ that repeatedly asks for secret keys sk_k for $k \in \mathcal{K}$. The problem is how to define the challenge ciphertext in a semantic security game. As usual, once the attacker obtains all the secret keys she desires, she outputs two challenge messages m_0, m_1 and expects to get back an encryption c of either m_0 or m_1 chosen at random by a challenger $\mathcal{C}$. If the attacker has a secret key sk_k for some $k \in \mathcal{K}$ such that $F(k, m_0) \neq F(k, m_1)$ then she can easily answer the challenge by outputting 0 if $(sk_k, c) = F(k, m_0)$ and 1 otherwise.

Hence, for the definition to be satisfiable, we must restrict the attacker's choice of m_0 and m_1 and require that:

$$F(k, m_0) = F(k, m_1) \; \forall \; k \; such \; that \; the \; attacker \; knows \; sk_k.$$

Definition (Security of a FE scheme). We define the following security game:

1. **Setup.** run (PP, msk) ← $\mathrm{setup}(1^\lambda)$ and give PP to the adversary,

2. **Query.** The adversary adaptively submits queries $k_i \in \mathcal{K}$ for $i = 1, 2 \ldots$ and is given $sk_i \leftarrow \mathrm{KeyGen}(msk, k_i)$,

3. **Challenge.** The adversary submits two messages m_0, m_1 to the challenger. The challenger picks $b \in \{0, 1\}$ and encrypts m_b under PP,

4. **Guess.** The adversary continues to issue key queries as before and eventually outputs a bit $b' \in \{0, 1\}$. The adversary wins the game is $b' = b$.

For $b \in \{0,1\}$, let W_b be the event that the adversary outputs 1 in the above game. The advantage of the adversary in the above game is:

$$FE_{\{ADV\}}[\mathcal{F},\mathcal{A}](\lambda) \coloneqq |\Pr[W_0] - \Pr[W_1]|$$

FE scheme is secure if for all probabilistic polynomial time adversaries $\mathcal{A}$, $FE_{\{ADV\}}[\mathcal{F},\mathcal{A}](\lambda)$ is negligible.

11 Overview of ASCLEPIOS Reference Architecture

In this section, we provide a description of ASLEPIOS Reference Architecture. More precisely, we describe and discuss the eight layers of ASCLEPIOS Reference Architecture as well as its main components and their general relationships. The section is divided into two main parts. First, we provide a high-level overview of the ASCLEPIOS architecture while in the second part we describe a more in-depth and technical presentation of the main components, their functionality and their interaction.

11.9 High-Level Overview of ASCLEPIOS Architecture

The goal of this section is twofold. First, we provide a high-level functional description of ASCLEPIOS main layers. This will allow the reader to understand the basic functionalities that will be offered by the ASCLEPIOS information-driven security framework. Second, we show how ASCLEPIOS fits into the life of the users and demonstrators.

Figure 7 illustrates a high-level overview of ASCLEPIOS architecture by showing its main components as well as how they interact with each other.

The ASCLEPIOS architecture consists of the following eight discrete layers:

1. Trusted Cloud Provider,
2. Crypto Layer,
3. Analytics Layer,
4. Policy Enforcement Layer,
5. Registration Authority,
6. Users,
7. Attestation Layer,
8. Revocation Layer.

More precisely, we show how ASCLEPIOS uses several cryptographic schemes to offer a secure cloud storage while at the same time allows users to share data in an efficient, secure and privacy-preserving way. In addition to that, ASCLEPIOS will improve the trustworthiness of electronic medical services by allowing healthcare professionals to verify the integrity of patients' devices by incorporating software-based remote attestation techniques.

Before we move on to a detailed description of each of the layers and its components, we first provide a high-level overview of the functionality that each layer offers. Furthermore, we will briefly describe how these layers and the underlying components interact with each other to successfully complete the core functionality that will be offered by ASCLEPIOS.

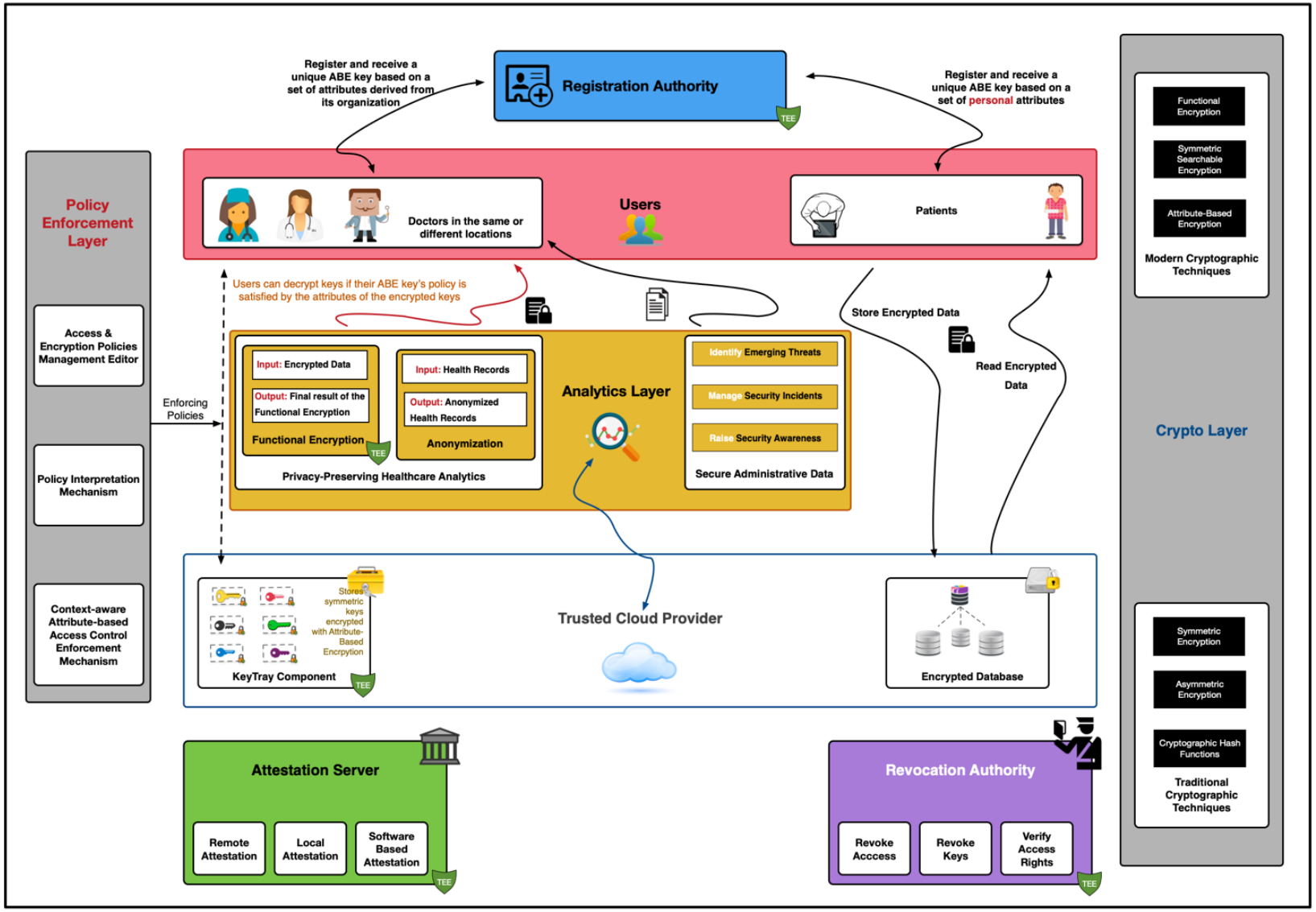

Figure 7: ASCLEPIOS Architecture

11.9.1 Cloud Service Provider

One of the common models of a cloud computing platform is Infrastructure-as-a-Service (IaaS). In its simplest form, such a platform consists of cloud hosts which operate virtual machine guests and communicate through a network. Often a cloud middleware manages the cloud hosts, virtual machine guests, network communication, storage resources, a public key infrastructure and other resources. The cloud middleware creates the cloud infrastructure abstraction by weaving the available resources into a single platform. In our system model, we consider a cloud computing environment based on a trusted IaaS provider similar to the one described in (32). The IaaS platform consists of cloud hosts which operate virtual machine guests and communicate through a network. Furthermore, the cloud service provider must support a TEE since core entities of the protocol will be running in an isolated environment.

As it can be seen in Figure 7, Trusted Cloud Provider contains two main components: an encrypted database and the KeyTray.

Encrypted Database. The encrypted database(s) will be built on top of the IaaS and can host multiple outsourced databases. This component will be responsible for storing and maintaining the data of all users that will be using of the ASCLEPIOS framework. Even though this database will be stored in the CSP, users' data will be protected from both internal and external attacks. This is because all data will be symmetrically encrypted with keys that will not be known to the CSP. Hence, a malicious CSP will not be able to learn any information about users' health records.

ASCLEPIOS Data Interaction

- Only registered users are able to send valid requests to the CSP in order to search for certain keywords over encrypted data.
- Only registered users are allowed to perform any of the operations of read/write/update on stored encrypted files.

Key Tray (KeyTray). Key Tray is key storage existing in the CSP and responsible for storing ciphertexts of all the symmetric keys that have been generated by different data owners and have been used to encrypt medical records. Only registered users can contact the KeyTray directly and request access to the stored ciphertexts. Moreover, the symmetric keys are encrypted with a CP-ABE scheme. Hence, only users with certain access rights are able to recover it. KeyTray is running on a TEE. Hence, anyone that is interacting with this component can verify its integrity. Thus, KeyTray is considered as a trusted entity in ASCLEPIOS architecture.

ASCLEPIOS Data interaction

- Only registered users can decrypt a symmetric key that is stored in the KeyTray based on a policy that is attached to the ciphertext.
- Only registered users can contact the KeyTray directly and request access to the stored ciphertexts based on the assigned set of attributes by their organisation.

11.9.2 Cryptographic Layer

The crypto layer will be implemented as a collection of cryptographic algorithms and will form one of the key components for both the security and the main functionality of ASCLEPIOS. This layer will provide a complete cryptography toolkit that will be used to protect stored data and secure the communication between connected components and entities in the system.

The novelty of this layer is the implementation of three modern encryption techniques: **Dynamic Symmetric Searchable Encryption scheme**, **Attribute-Based Encryption scheme**, a **Functional Encryption scheme** as well as a list of various standard cryptographic schemes and functions.

Dynamic (SSE). This is a symmetric encryption scheme that will allow registered users of ASCLEPIOS to generate locally a symmetric encryption key and encrypt their data (locally) before sending them to the CSP where they will be stored in a remote location. The SSE scheme that will be used in ASCLEPIOS will allow users with certain access rights to *search* directly over their encrypted data, *delete* existing and *update* encrypted files and *add new* ciphertexts of their medical records. The SSE scheme of ASCLEPIOS will be designed and implemented in such a way that the specific needs of the healthcare industry will be met. Moreover, in the healthcare industry a large number of the generated medical data is not text-based (e.g. images). For this reason, ASCLEPIOS SSE scheme will not only allow to search for keywords over text-based encrypted files but also search for keywords of non-text files' encrypted metadata. (e.g. picture's size, title, timestamp of creation or modification).

CP-ABE. This is an asymmetric encryption scheme that allows users with different keys to decrypt the same ciphertext. More precisely, in a CP-ABE scheme, a user can encrypt data based on a policy. In addition to that, each user gets a unique private key that also contains a list of personal attributes. By acquiring this private key, the user can decrypt the generated ciphertext *if and only if* the attributes that are attached to her key satisfy the policy that is bound to the underlying ciphertext. In ASCLEPIOS, CP-ABE will play an important role in ***storing*** and *managing the symmetric keys* that will be generated by the users and will be used for the encryption of their medical records with an SSE scheme. While the main CP-ABE library will be part of the Crypto Layer, other components, such as the KeyTray, will be extensively use this scheme and take advantage of the properties that has to offer.

FE. This is a modern encryption technique where a decryption key enables a user to learn a specific function of the encrypted data and nothing else. In ASCLEPIOS, a Registration Authority (trusted authority) generates and holds a master secret key only known to the authority. Users provide the description of some function f as input to RA. Then, RA uses its master secret key to generate a derived secret key sk_f linked with f. Now anyone holding sk_f can compute $f(x)$ from encryption of any x encrypted under the master public key [33]. In other words, users (healthcare practitioners and researchers), with specific access rights, will be capable of decrypting some part(s) of encrypted data without learning anything about the actual content in a privacy-preserving manner. Furthermore, FE will be part of the Crypto Layer running in an isolated execution environment, mainly due to its nature.

11.9.3 Analytics Layer

The project foresees that a massive volume of data will be generated and shall be processed with ASCLEPIOS framework, and therefore the analyses will result in insights with potential significance.

To this end, a separate layer that will be responsible for conducting analytics based on a wide range of health data will complement the ASCLEPIOS architecture. The Analytics Layer will allow authorized stakeholders of a healthcare organization to perform analytics based on stored medical data. ASCLEPIOS analytics functions will be designed based on the specific requirements that will be defined by the project's healthcare experts. Several techniques for analyzing big data will be incorporated and offered by this layer. More precisely, machine learning and deep learning techniques, such as regressions, support vector machines, and *k*-means clustering will be used for analyzing medical records and output important conclusions.

The Analytics Layer will be based on two main components: **Privacy-Preserving Health Data Analytics** Component, **Security Administrative Data** Analytics Component.

Privacy-Preserving Health Data Analytics. This component will be strictly operating in an isolated environment and will be used only by authorized individuals who can access the data to perform health data analytics in a privacy-preserving way. More precisely, this component will use FE and be able to perform several statistical algorithms on the given ciphertexts. At the end of the process, the user that initiated the process will learn the final result of the computation without learning anything about the actual content of the individual data. Hence, any private data that might be contained in the ciphertexts will be protected. (e.g. EMR data).

Security Administrative Data Analytics. This component will employ various data exploration and analysis methods and apply them strictly on data pertaining to the system's security, e.g. access logs and decryption activity logs. Its analytics engine, which will be built on well-known data analytics frameworks and libraries, will provide simple statistical computations, but also advanced machine learning algorithms targeting the identification of events pointing to abnormal access patterns (and behaviours in general) and/or possible security threats and incidents.

ASCLEPIOS Data Interaction

Privacy-Preserving Health Data Analytics entails the following tasks:

- Authorized users of ASCLEPIOS will be able to provide lists of encrypted data (e.g. patients' data) as input to an FE scheme that will be running in a TEE. Upon reception, FE will apply one of the implemented data-mining algorithms to the data. Users will then learn the final result of the computation without learning anything about the actual content of the individual data.
- Authorized data analysts will be able to provide a list of plaintext data as input (e.g. patient's data). Upon reception, the analytics engine will perform complex queries, run advanced machine learning algorithms and finally produce important results based on the analysis oft he given data.

Security Administrative Data Analytics entails the following main tasks:

- Authorised users of ASCLEPIOS will be able to query and analyse data related to infrastructure security, including access logs and monitored encryption and decryption activities. The underlying analytics engine will apply appropriate techniques, ranging from statistical analysis to complex event processing and machine learning-based anomaly detection to identify normal, abnormal and possibly unsecure behaviours and assess the status of the system in terms of security. The users will be able to parameterise and narrow down the exploration to specific data of interest, explore how these are accessed and monitor system security to timely identify incidents and security threats.
- Data owners that use the ASCLEPIOS framework will be able to review how their data are being managed and accessed and for which purposes. Metrics related to how their data are encrypted, decrypted, queried and used will be calculated by the analytics engine and provided to the users in a visual way that will help them assess the efficiency of the access policies they define and gain a deeper understanding of how they should safeguard their data.

11.9.4 Policy Enforcement Layer

The policy enforcement layer will introduce the appropriate methods and tools to implement an efficient and flexible access policies enforcement middleware, adequate for modern cloud-enabled healthcare systems. Essentially this involves fine-grained authorisation along with security awareness capabilities. These capabilities refer to the context-aware privacy preservation of sensitive data while the latter corresponds to data owners' education about the security competence of the ASCLEPIOS solution along with their cultivation with respect to potentially compromising behaviours. The main novelty of the Policy Enforcement Layer is that it enables the construction of a concrete list of policies that can be used with ABE and ABAC schemes based on the specific needs of healthcare organisations.

The policy enforcement layer will be based on three main components: **Access & Encryption Policies Management Editor**, **Policy Interpretation Mechanism** and **Context-aware Attribute-based Access Control Enforcement Mechanism**.

Access & Encryption Policies Management Editor. The editor will support policy editing functionalities for ABAC and ABE in the healthcare application domain. The editor will be based on a model for formally capturing the background knowledge for enabling the ABAC and ABE paradigms. It will support the development of the semantics of the ASCLEPIOS context-based access control and ABE related policies.

Policy Interpretation Mechanism. This software component will provide a mechanism capable of translating the developed policies (i.e. instantiated ASCLEPIOS models) and cope with their run-time updates (made by the editor) into the appropriate format for enabling either the ABAC enforcement or the realization of ABE. It will essentially undertake the task of automatically translating the instantiated ASCLEPIOS models (that express the creation or update of authorization policies) to an acceptable input for the Context-aware Attribute-based Access Control Enforcement Mechanism.

Context-aware Attribute-based Access Control Enforcement Mechanism. This software will implement a mechanism for enforcing efficient authorization on sensitive medical data access requests by considering contextual information. Well-known XACML authorization engines will be examined (e.g. Balana, AuthzForce, PaaSword's Drools-based engine) for identifying, reusing and extending the most appropriate one based on the trade-off between flexibility, expressivity support and efficiency. The mechanism should be able to cope with the enforcement of both ABAC and ABE policies. In addition, it should be able to accommodate run-time changes of any authorization policy deployment without any downtime in order to grant, deny and manage any incoming access request. The mechanism will implement the essential decoupling between the access decisions and the points of use (i.e. Policy Enforcement Points (PEP) of the XACML standard specification). In addition, this mechanism will build on the authorization decisions along with the security awareness aspects of the ASCLEPIOS Context model in order to inform and educate the healthcare actors with respect to possible security issues concerning certain types of access attempts.

ASCLEPIOS Data Interaction

Access & Encryption Policies Management Editor, Policy Interpretation Mechanism and Context-aware Attribute-based Access Control Enforcement Mechanism entail the following main tasks:

- Using the editor, Cloud Application Operators (DevOps) of ASCLEPIOS will be able to develop context-based access control policies, as well as static policies concerning the manner in which medical data can be decrypted according to the ABE paradigm.
- With the mechanisms, Cloud Application Operators (DevOps) of ASCLEPIOS will be able to enable the context-aware ABAC enforcement.

- The Context-aware Attribute-based Access Control Enforcement Mechanism will also raise the involved actors' security awareness by indicating the logic of each authorization decision.

11.9.5 Registration Authority

This layer is responsible for the registration of medical personnel as well as patients to the underlying electronic healthcare service. Furthermore, RA will be responsible for generating and distributing ABE keys to the registered users based on a set of attributes derived from its organization or based on the set of personal attributes for individuals. In addition to that, RA will be responsible for the distribution of necessary public and private key parameters that will be used to establish secure communication channels between different components. Furthermore, RA will not just only distribute ABE keys to the registered users, but also secret functional keys sk_f, for variety functions f, to users in order to perform functional decryption for a function f, where function f, will be represented as a program running in an isolated environment. Thus, RA is considered as a single trusted authority and should be running on a TEE. Please see example of attributes in Figure 6.

ASCLEPIOS Data Interaction

- Medical personnel from local or remote locations able to register and receive a unique ABE key based on a set of attributes derived from its organisation.
- Patients are able to register and receive a unique ABE key based on a set of personal attributes.

11.9.6 Attestation Server

Attestation Server will allow certain users to validate server's integrity and identify possible unauthorized modifications. Therefore, the use of attestation will be the main tool that will be used in ASCLEPIOS to provide certain guarantees to users regarding the trustworthiness of the overall service that will be offered. The main goal of this layer is to allow users to take a decision regarding the trustworthyness of the target prior to usage. This will be accomplished by collecting enough information about the target, such as hardware, software, and configuration data to ensure its code integrity.

ASCLEPIOS attestation server will offer two main different types of attestation: **Hardware-Based Attestation** that can be either remote or local and **Software-Based Attestation** that will be done remotely.

- **Hardware-Based Attestation.** Integrity, confidentiality and trustworthiness of the workload execution will be implemented by using the Hardware-Based Attestation of this layer. This logical architecture layer will expose an aggregated pool of isolated execution capabilities, based on hardware security features (both bare-metal and virtualized), available on the ASCLEPIOS cloud platform. In the hardware-based attestation users will rely on a hardware roots of trust available in ASCLEPIOS underlying servers. Users will then be able to perform a remote or a local attestation. For the remote attestation, the aforementioned trusted hardware will be used to present reliable evidence to remote parties about the software that it is running. In the local attestation setting, two or more entities that run on the same host collaborate with each other and authenticate each other locally by using special encryption keys. The output of this process is a verification that the counterpart is running on the same platform by applying a proof based on a key-exchange protocol. The successful result of local attestation will offer a protected channel between the involved entities and will guarantee the confidentiality and integrity of the future communication between the involved parties. In ASCLEPIOS, local attestation will be used to verify the integrity

and establish a secure communication between two or more entities that are on the same platform and are running in a TEE.

- **Software-Based Attestation.** In contrast to Hardware-Based Attestation, Software-Based attestation enables the integrity verification of untrusted devices without requiring any particular hardware. In ASCLEPIOS, we will be using a software-based attestation protocol to verify the memory contents of some medical devices used by patients. This will allow a doctor to establish the absence of malicious changes to the memory contents of a patient's data before they start a remote session (e.g. in a telemedicine setup). Finally, the software-based attestation will run remotely and will be able to detect with high probability changes in memory contents, thus detecting possible malicious files or unexpected configuration settings.

11.9.7 Revocation Authority

This component is responsible for the revocation of users (users that might have been compromised or just lost their access rights). In ASCLEPIOS the revocation authority will be running in a TEE and will maintain a list with a mapping of all the compromised users with the corresponding keys. Furthermore, in ASCLEPIOS, we separated the revocation mechanism from the underlying CP-ABE scheme. This strategy, results in a more efficient, more reliable and more secure revocation mechanism.

ASCLEPIOS Data Interaction

- The data owner will contact the revocation authority and ask to disable the access of a certain user to some or all of her files.
- The data owner will contact the revocation authority and ask to modify the access of a certain individual or a group of individuals to her files (e.g. change from read/write to read).

11.10 Detailed Description of ASCLEPIOS Architecture

In this section we will provide a detailed description of the different layers that constitute ASCLEPIOS architecture. Each component is described by analyzing the following five fields:

- **Function** describes the purpose and main role of the component within the ASCLEPIOS architecture.
- **Sub-Components** lists the components (if applicable) that are running inside the described component.
- **Sources** lists the components that provide data or any other input to the described component.
- **Consumer** lists the components that feed from the activities or data produced by the described component.

11.10.1 Cryptographic Layer

The cryptographic layer contains all the traditional cryptographic primitives as well as modern cryptographic primitives such as Symmetric Searchable Encryption, Attribute-Based Encryption and Functional Encryption. It is considered as the fundamental layer of ASCLEPIOS since it will be used by all entities to communicate securely, as well as for storing and managing data and access rights in a secure and privacy-preserving way. The main algorithms that will constitute this layer will be discussed in detail in *WP2: Operations on Encrypted Health Data and Privacy-Preserving Health Data-Driven Analytics*.

Function. The cryptographic layer will use traditional cryptography to secure message exchanges between different entities and also to offer a variety of other facilitations such as secure storage, verification of the integrity of the stored files and authentication. Moreover, modern cryptographic techniques such as Symmetric Searchable Encryption, Ciphertext-Policy Attribute-Based Encryption and Functional Encryption will be used to provide efficiency, protection against both internal and external attacks and data analysis in a privacy-preserving way. Furthermore, a list of traditional cryptographic algorithms and schemes will be part of this layer. More precisely, a wide range of private and public encryption schemes will be offered as well as a list of important protocols that will allow two or more entities to establish a secure communication channel. Furthermore, cryptographic hash functions and key derivation functions will provide a more complete cryptographic toolkit for ASCLEPIOS.

Sub-Components. The Crypto Layer is divided into two sub-components. Namely, traditional cryptography (Figure 8) and modern cryptography (Figure 9).

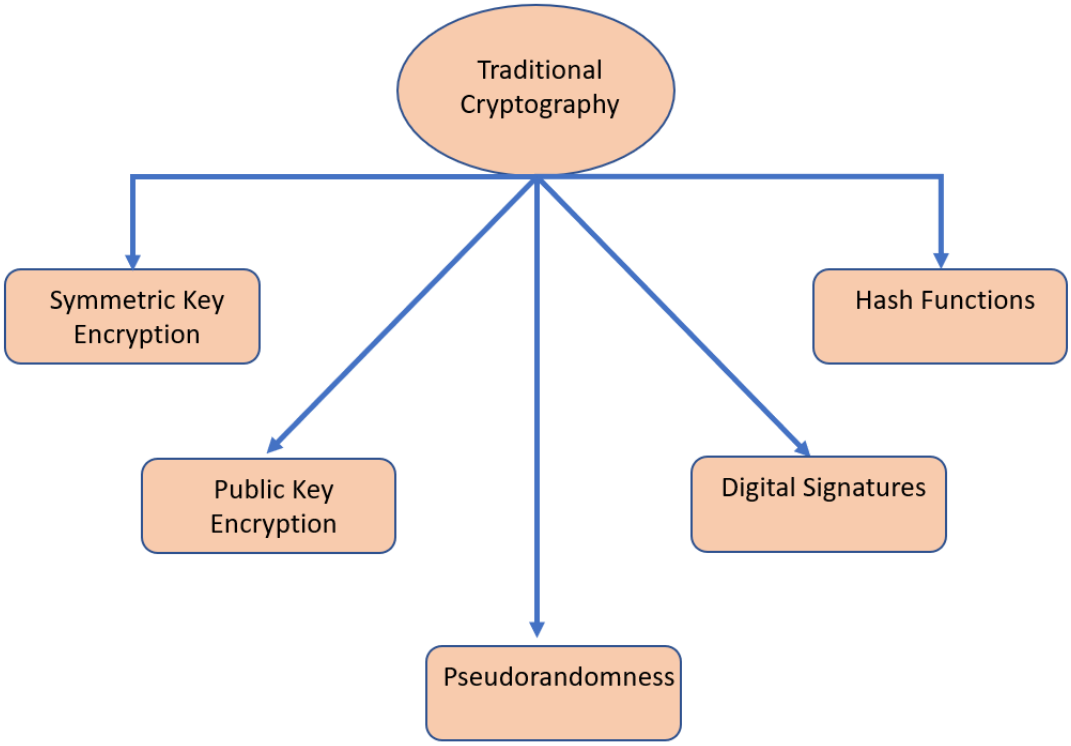

Figure 8: Traditional Cryptography

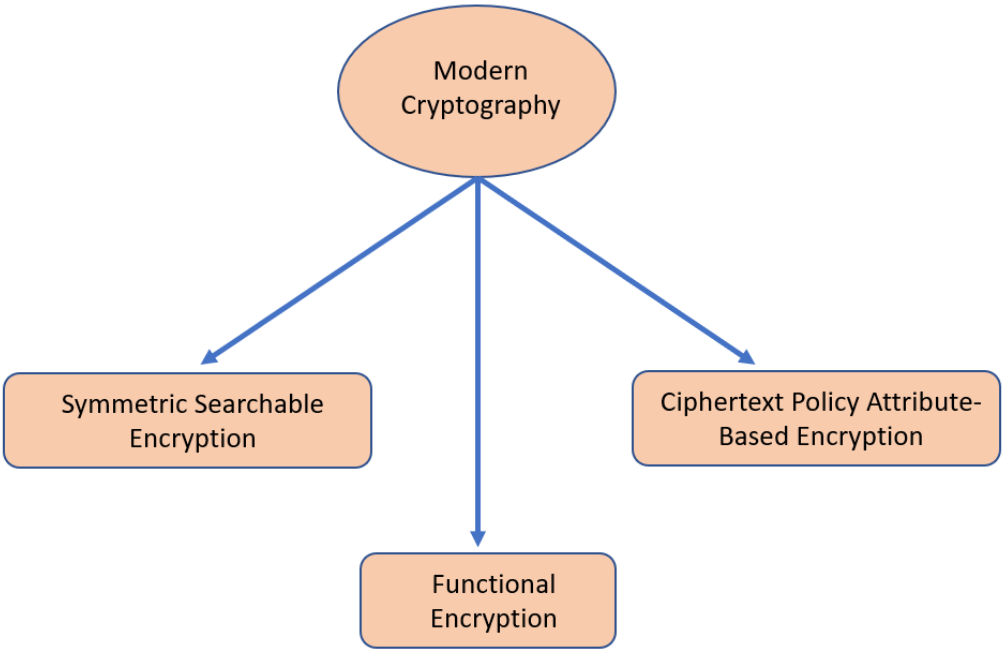

Figure 9: Modern Cryptography

Sources. Every communication in ASCLEPIOS will be protected using cryptography. As a result, every message exchange will by processed by the cryptographic layer. Moreover, file storing should be done in a privacy-preserving way. To do so, each user that wishes to store files on the cloud will first have to encrypt them. Finally, each time a doctor or a researcher wishes to perform privacy-preserving analytics, the cryptographic layer will make sure that the patients' privacy will not be violated.

Table 35 summarizes the sources of the Cryptographic layer.

Sources
Users
CSP
Analytics Layer
Registration Authority
Revocation Authority
Attestation Server
Policy Enforcement Layer

Table 35: Crypto Sources

Consumer. Similarly, all the entities will consume data output from the cryptographic layer.

Table 36 provides a collective list of all the consumers of the Cryptographic layer.

Consumer

Users
CSP
Analytics Layer
Registration Authority
Revocation Authority
Attestation Server
Policy Enforcement Layer

Table 36: Crypto Consumers

11.10.2 Policy Enforcement Layer

The work regarding the policy enforcement layer involves the research and development of methods and tools that enable the efficient and flexible policies enforcement with respect to authorizing access to sensitive healthcare data. This layer exploits the background knowledge, formally modelled as contextual circumstances under which access requests to healthcare data should be permitted or denied.

Function. The core functionalities of the policy enforcement layer involve fine-grained authorization of any incoming access requests to sensitive data along with security awareness capabilities. In order to accomplish this, several mechanisms will be developed with the following functionalities: i) based on ASCLEPIOS models, create and edit policies that can be used for ABE and ABAC authorization; ii) interpret ABE and ABAC authorization policies into enforceable XACML ABAC and CP-ABE policies; iii) authorize incoming access requests based on the ABAC scheme; iv) match key attribute values with CP-ABE policies for permitting the decryption of the symmetric key(s) used for encrypting the sensitive data and v) provide security awareness according to the defined and matched authorization policies. Summarizing, the policy enforcement layer will introduce capabilities that refer to the context-aware privacy preservation of sensitive data while it will augment the security awareness of ASCLEPIOS users.

Sub-Components. The Policy Enforcement Layer is divided into three sub-components. Namely, Access & Encryption Policies Management Editor, Policy Interpretation Mechanism and Context-aware Attribute-based Access Control Enforcement Mechanism.

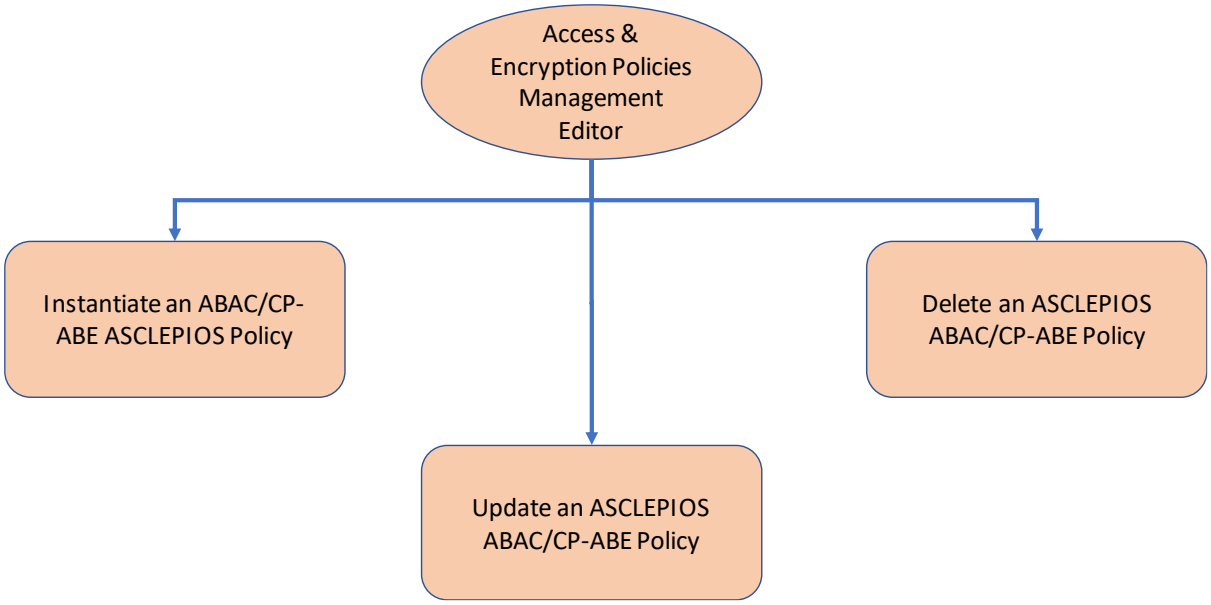

Figure 10: Access & Encryption Policies Management Editor

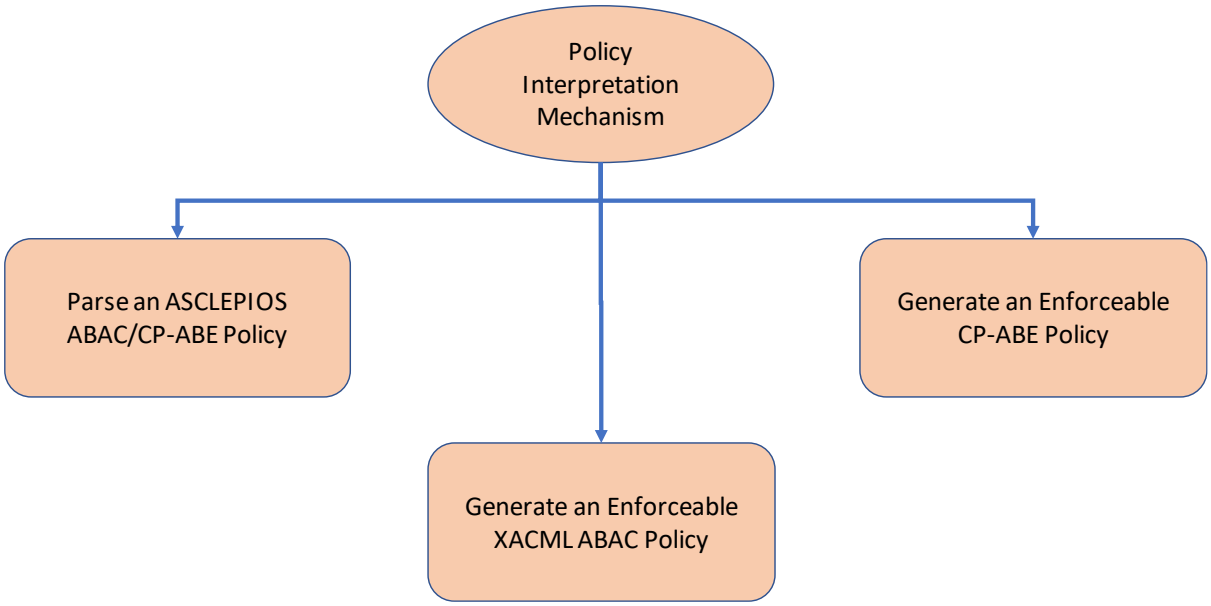

Figure 11: Policy Interpretation Mechanism

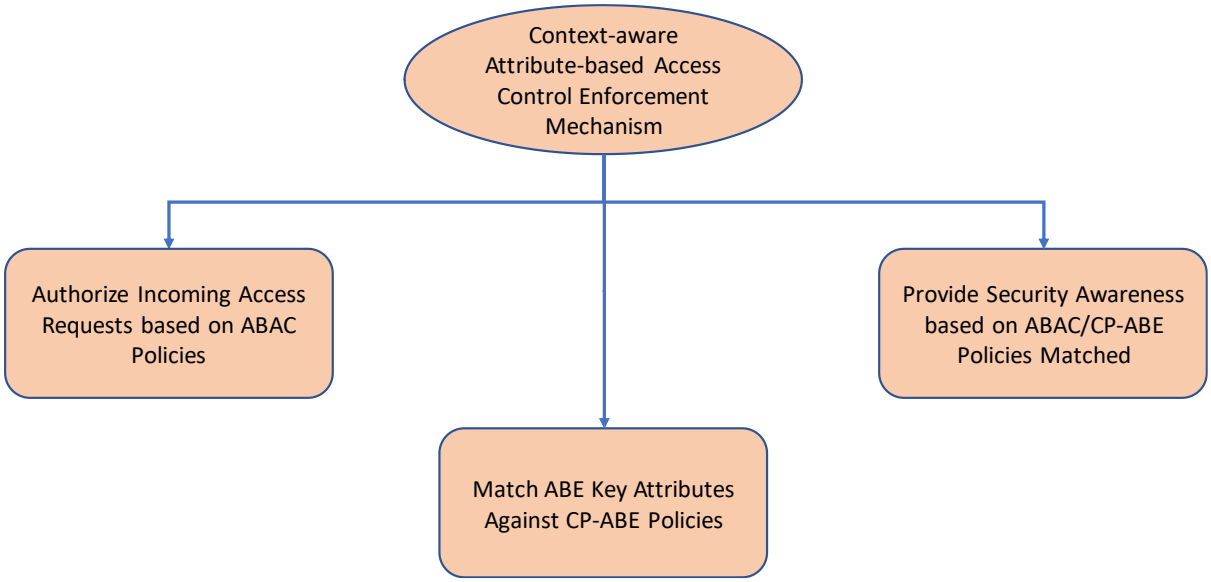

Figure 12: Context-aware Attribute-based Access Control Enforcement Mechanism

Sources. The ASCLEPIOS users may provide input to this layer in the sense that may instantiate the ASCLEPIOS Context and Policy models in order to describe the contextual circumstances that should provide access to healthcare data (e.g. reading the data, performing analytics processing over the data in privacy or non-privacy preserving way) and/or allow the decryption of the symmetric key used for encrypting this data. We foresee a necessary communication with the Crypto Layer with respect to the ABE scheme, while input from the Registration Authority is also considered in the form of unique ABE keys of the users. The input from the Revocation Authority is related to establishing the validity of the user that just posted an access request (i.e. her key has not been revoked).

Table 37 summarizes the sources of the policy enforcement layer.

Sources
Users
CSP (KeyTray Component)
Crypto Layer
Registration Authority
Revocation Authority

Table 37: Policy Enforcement Layer Sources

Consumer. Similarly,

Table **38** provides a collective list of all the entities that we expect to consume the output of the policy enforcement layer.

Consumer
Users
CSP (Encrypted Database)
Crypto Layer
Analytics Layer

Table 38: Policy Enforcement Layer Consumers

11.10.3 Analytics Layer

The analytics layer will be mainly used by healthcare professionals or researchers in the healthcare section. Analysis of data can uncover correlations and patterns and provide answers and insight to some questions of uttermost importance such as:

- What are the causes, development and effects of diseases? (research purposes)
- What interventions can improve prevention, diagnostics and therapies (methods, procedures and treatments)? (research purposes)
- Continuous evaluation of safety, effectiveness, efficiency, accessibility and quality of interventions (quality assurance/quality improvement work purposes)

These questions are of major importance in healthcare as their answers can prevent deseases and save lives.

Function. The analytics layer aims at providing doctors and researchers with valuable insights on healthcare data. In Table 39, we list the main functions of the analytics layer.

Functions
Privacy-Preserving Health Data Analytics
Security Administrative Data Analytics

Table 39: Analytics Functions

Sub-Components. We divide the analytics layer into Privacy-Preserving Health Data Analytics and Security Administrative Data Analytics.

- **Privacy-Preserving Health Data Analytics** is further divided into two categories.

  The first category involves the data that will be accessed through a FE scheme that will be further explored and finalized in WP2. With the use of the functional encryption scheme, we can ensure that the privacy of the medical records will be preserved, but at the same time doctors and researchers will be able to perform data analysis. Analytics will be computed in a trusted execution environment to ensure that all processed data remain protected. In particular, to perform privacy preserving analysis on requested data we will make use of at least three different TEE-enabled environments, one of which will be part of the *RA* and the rest will constitute the *Decryption Node,* which involves *Decryption Environment* and *Function Environment*.

  Decryption Environment has the role of transferring the secret decryption key to authorized programs running within environemnts on the same platform which we refer as function environemnts. The Decryption environemntwill verify the code running inside a function environemnt via local attestation. The local attestation also establishes a secure channel from the Decryption environemnt to the function environemnt. If the verifications of the local attestation and the signature pass, then Decryption environemnt transfers the secret key to the function environemnt over the secure channel.

  The Functional decryption for a function f is performed inside a ***Function Environement*** that loads the function upon initialization. A user's application authorized to decrypt f can operate the function environemnt either locally or remotely.

If the application is running locally on the decryption node, then it can directly call into the function environemnt, providing as input a vector of ciphertexts and a signature. A remote client application will need to establish a secure channel with the environemnt via remote attestation. If the signature is valid, the Function environemnt will contact the Decryption environemnt in order to receive the secret key needed for decryption. It will then decrypt the ciphertexts and passed the vector of decrypted plaintexts as input to the user-defined function and records the output.

The second category involves ***Anonymization Techniques*** that will also be specified in WP2. These techniques will enable interested parties (such as researchers or healthcare professionals) to perform statistical analysis on data stored on the CSP without leaking ownership details. The results can be used for research, public health, and quality improvement purposes

- **Security Administrative Data Analytics** is also divided into two categories based on their scope.

  The first category will offer cybersecurity, encryption and access analytics for CSP operation and Healthcare Providers systems used within the ASCLEPIOS framework.These will be based on statistical analysis, machine learning and complex event processing. The exact algorithms and methods to be applied, as well as the libraries and frameworks to be leveraged for the implementation, will be explored and documented in the context of WP2 activities. Techniques ranging from anomaly detection, both supervised and unsupervised to computation of targeted security-related metrics will be applied on activity and access logs to model behaviour patterns (from the perspective of infrastructure security) and to identify abnormal activity.

  The second analytics category increases security awareness by addressing the need of data owners, i.e. patients, to understand in which context and by whom their data are being accessed and used. This involves providing insights on both granted and denied access requests, on data encryption and decryption activities, on data analysis methods that were applied on the data etc. Through preconfigured and custom metrics, data owners will be able to assess and check the efficiency of the policies they define in the ABE schemes and get evidence-based insights on how to better safeguard their data.

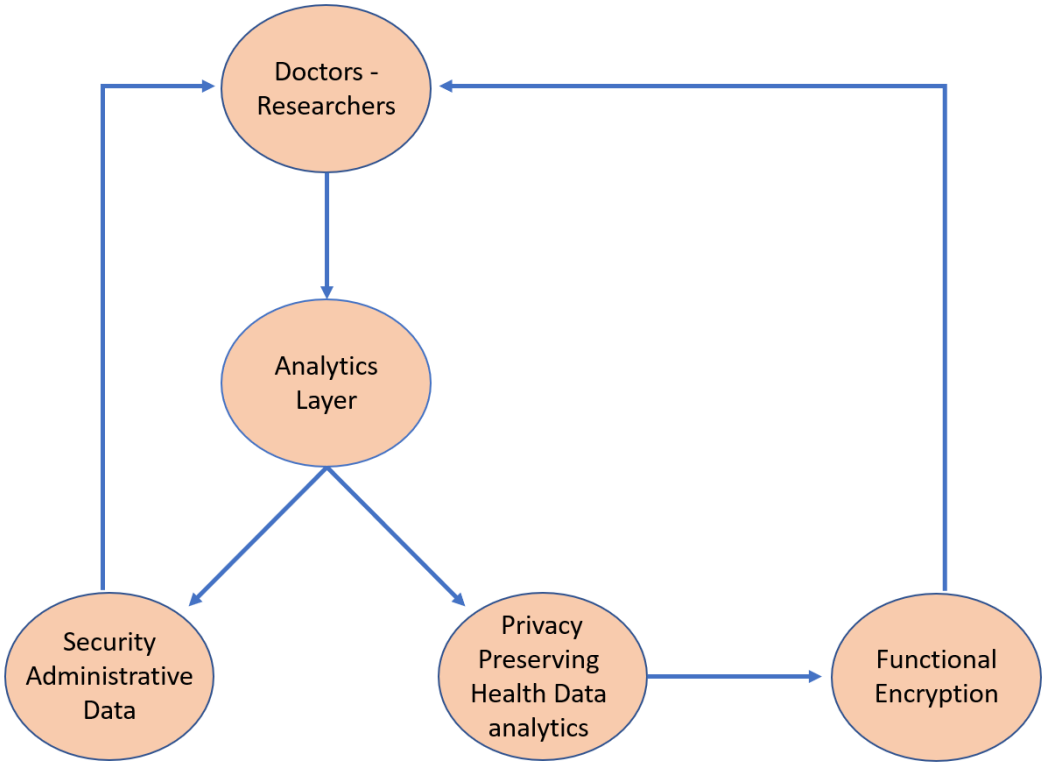

Figure 13: Analytics Layer

Sources. The Analytics Layer will receive input from users (doctors and researchers). For *Privacy-Preserving Health Data Analytics,* the users will provide the analytics layer with an authorization issued by the *RA* for a specific function. Apart from that, the *RA* will send the master functional encryption secret key $\boldsymbol{msk}$ to the analytics layer, so that the data will be decrypted. Finally, the CSP will be responsible for providing the analytics layer with specific data that will be used for analytics.

In

Table **40**, we list all the sources of the Analytics Layer.

Sources
Users
CSP
Registration Authority

Table 40: Analytics Sources

Consumer. The output of the Analytics Layer will be given to the same users who have initialized the process. In particular, to those users who have initialized a Function environemnt.

Table 41 presents a list with all the consumers of the Analytics Layer.

Consumer
Users

Table 41: Analytics Consumers

11.10.4 Attestation Server

The attestation server allows to authenticate software and hardware configuration of specific devices.

Function. The attestation server has three different functionalities. ***Remote Attestation*** in which any remote party which is TEE-enabled can verify the integrity and the trustworthiness of an entity. This is a key factor to ASCLEPIOS, since it enables us to diminish restrictions based on location. On the other hand, ***Local Attestation*** enables TEE-enabled devices that reside on the same platform, to attest each other. Finally, since not all personal devices can be TEE-enabled, the Attestation Server needs to support ***Software-Based Attestation*** as well**.** Software-Based Attestation will allow the verification of the integrity and the trustworthiness of an entity based on measurements of the software and not the hardware.

Hardware-Based Attestation. Integrity, confidentiality and trustworthiness of the workload execution will be implemented by using the Hardware-Based Attestation of this layer. This logical architecture layer will expose an aggregated pool of isolated execution capabilities, based on hardware security features (both bare-metal and virtualized), available on ASCLEPIOS cloud platform. In the hardware-based attestation users will rely on a hardware that will be installed in ASCLEPIOS underlying servers and will be inherently trusted. Users will then be able to perform a remote or a local attestation. For the remote attestation, the aforementioned trusted hardware will be used to present reliable evidence to remote parties about the software it is running. In the local attestation setting, two or more entities that run on the same host collaborate with each other and authenticate each other locally by using ~~special~~ encryption keys. The output of this process is a verification that the counterpart is running on the same platform by applying a proof based on a key-exchange protocol. The successful result of local attestation will offer a protected channel between the involved entities and will guarantee the confidentiality and integrity of the future communication between the involved parties. In ASCLEPIOS, local attestation will be used to verify the integrity and establish a secure communication between two or more entities that are on the same platform and are running in a TEE.

Software-Based Attestation. In contrast to Hardware-Based Attestation, Software-Based attestation enables the integrity verification of untrusted devices without requiring any particular hardware. In ASCLEPIOS, we will be using a software-based attestation protocol to verify the memory contents of medical devices that will be used by patients. This will allow a doctor to establish the absence of malicious changes to the memory contents of a patient's before they start a remote session (e.g. in a telemedicine setup). Finally, the software-based attestation will run remotely and will be able to detect any changes in the memory contents with high probability, thus detecting possible malicious files or unexpected configuration settings.

Sub-Components. The attestation server is a single component.

11.10.5 Cloud Service Provider

Function. CSP is responsible for storing patients' encrypted medical data along with the searchable indexes needed to perform search, add and delete operations. Moreover, an isolated encironment called Key Tray will run on the CSP in order to enable the sharing of the symmetric keys used to encrypt the patients' data.

Table 42 provides a collective list of all main functions that will be supported by the CSP.

Functions
Store encrypted files
Store searchable indexes
Store encrypted keys
Search on the encrypted database
Update the encrypted database
Provide the Analytics Layer with ciphertexts
Share the encrypted keys

Table 42: Cloud Service Provider

Sub-Components. The CSP should support two different sub-components. The Encrypted database and the Key Tray.

- **Encrypted database.** This is a database that stores the ciphertexts of the patients' data, along with the searchable indexes. In particular, a patient will encrypt her data and generate the indexes using Symmetric Searchable Encryption. The encrypted data will only be available to authorized users.

- **Key Tray.** KeyTray is a key storage that exists in the CSP and stores ciphertexts of all the symmetric keys that have been used to encrypt data. A patient, will first generate a symmetric key to encrypt her data. As a next step, she will encrypt the symmetric key using CP-ABE and she will send it to the KeyTray. The Key Tray will also be responsible for sharing the ciphertexts of the symmetric keys to authorized doctors.

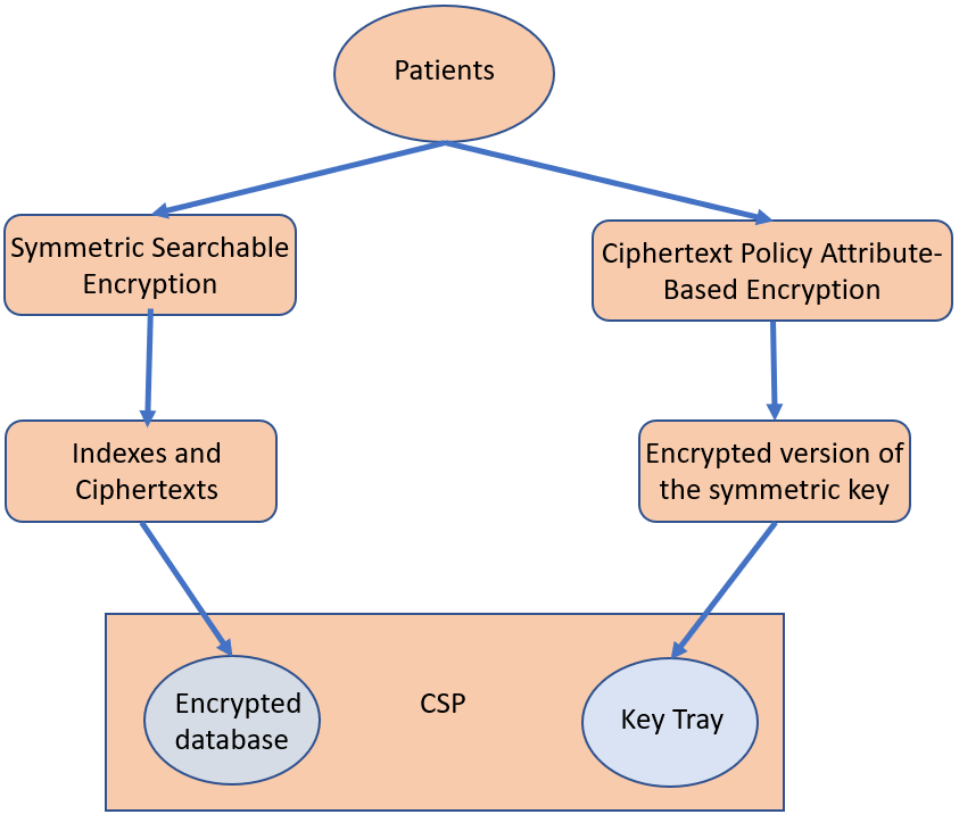

Figure 14: CSP

Sources. The CSP gets input either directly from users or through the Analytics Layer. In the first case, patients contact the CSP in order to store their encrypted medical data, add or delete files and finally, store the symmetric key they used for encrypting their data. Furthermore, doctors and researchers contact the CSP in order to receive the symmetric keys that was stored in the Key Tray by the patients.

On the other hand, authorized users can contact the CSP through the Analytics Layer. In particular, after a user gets authorized by the registration authority to run Privacy-Preserving Analytics, she sends a request to the Analytics Layer. This request will be forwarded to the encrypted database of the CSP in order to receive back the files needed for the data analysis.

Table 43 summarizes all CSP sources.

Sources
Users
Analytics Layer

Table 43: CSP Sources

Consumer. The CSP will output to the users' files that have been searched for and the encrypted version of the symmetric key that is needed for accessing the encrypted database. Moreover, the CSP will provide the Analytics Layer with specific files that are required to run Privacy Preserving Analytics.

Table 44 presents all CSP consumers.

Consumer
Users
Analytics Layer

Table 44: CSP Consumers

11.10.6 Registration Authority

RA is a single component running in a TEE. RA is responsible for the registration of users. Moreover, it is responsible for setting up all the necessary public and private parameters that are needed for establishing secure channels between different components. Furthermore, RA will generate and distribute ABE keys to the registered users and secret functional keys sk_f, for different functions f, for the users to perform functional decryption for a function f. The function f will be represented as an program running in an isolated environment.

Function. Provide users with valid credentials in order to enable the communication of the users with different components of ASCLEPIOS's architecture. These credentials will also include a secret CP-ABE key $\boldsymbol{sk}_{\mathcal{A},u_i}$ where $\boldsymbol{u_i}$ is the identifier of a user and $\boldsymbol{\mathcal{A}}$ denotes the set of $\boldsymbol{u_i}$'s attributes. Each time a legitimate user wishes to perform analytics, she first needs to request authorization for a function $\boldsymbol{f}$ from RA. RA will then issue a secret functional key $\boldsymbol{sk_f}$ that will be sent back to the user in order for her to proceed with data analysis and it will also send to the analytics layer Master functional encryption secret key.

Table 45 presents a collective list of the functions that will be supported by the RA.

Functions
Setup of all public and private parameters
Register users
Provide users with ABE keys
Provide users with FE keys
Provide the Analytics Layer with the master functional secret key

Table 45: RA Functions

Sub-Components. The RA consists of a single component.

Figure 15 shows the UML diagram for RA.

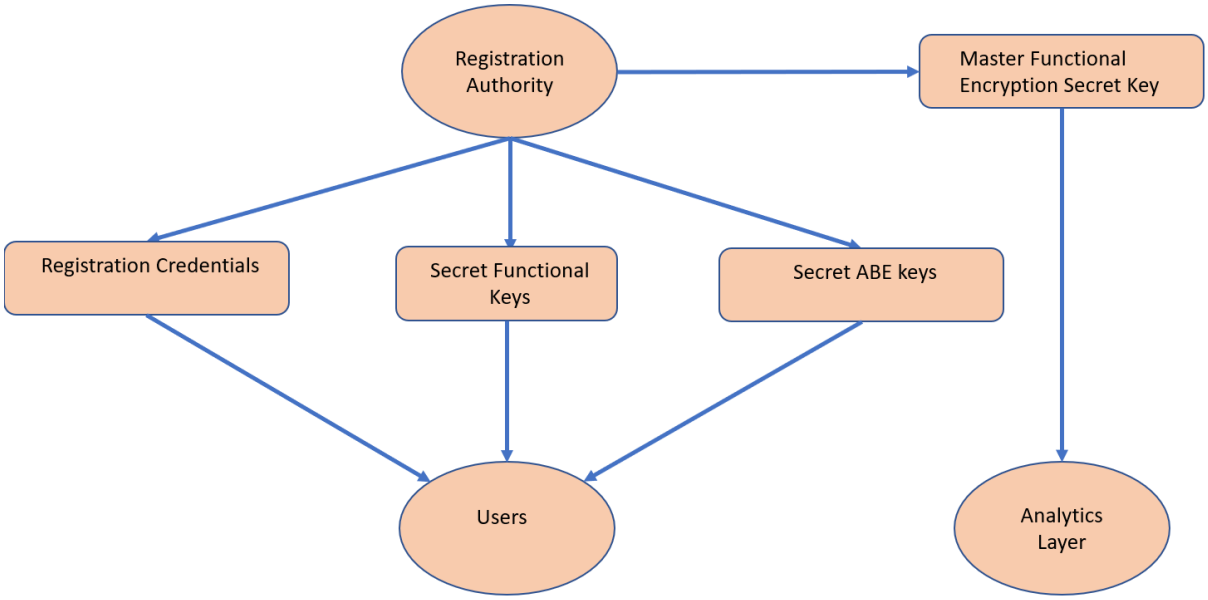

Figure 15: Registration Authority

Sources. RA gets input from either new users that wish to register, or from already registered users that want to receive a secret ABE key or get authorization for a secret FE key. Table 46 summarizes the sources of RA.

Sources
Users

Table 46: RA Sources

Consumer. Secret CP-ABE keys and FE keys are sent directly to the users that issued the corresponding request. Moreover, after a request for a specific function gets validated, the RA will send the master functional encryption secret key to the Analytics Layer. In Table 47, we list all RA consumers.

Consumer
Users
Analytics Layer

Table 47: RA Consumers

11.10.7 Revocation Authority

REV is responsible for maintaining a revocation list (*rl*) in plaintext with the unique identifier of the users that have been revoked. Every time that a user is revoked, REV needs to update *rl*. This will prevent revoked users from accessing ciphertexts that are not authorized anymore. Moreover, REV will control the access rights of the users.

Function. Upon reception of a request from a data owner (a patient), the revocation authority will revoke the access to a specific user. Moreover, for a registered user to start interacting with the CSP, she first needs to get an authorization from the RA. This authorization will prove to the CSP that the user's access rights are still valid. In

Table **48**, we list all the functions of the RA.

Functions
Revoke users
Provide users with proof of their access rights

Table 48: REV Functions

Sub-Components. The Revocation Authority is a single component.

Sources. The Revocation Authority gets input both from patients and doctors or researchers. The patients provide the Revocation Authority with the identities of the users to be revoked while the doctors and researchers request a verification of their access rights. In Table 49, we list all the sources of the Revocation Authority.

Sources
Users

Table 49: REV Sources

Consumers. Every time a user is revoked, the revocation authority updates its revocation list. Hence, it does not need to send a notification to the newly revoked user. The revoked user will learn that she is revoked, once she tries to get a verification of her access rights. Table 50 presents a list of all the consumers of the Revocation Authority.

Consumers
Users

Table 50: REV Consumers

11.10.8 Users

Users are the cornerstone of ASCLEPIOS. ASCLEPIOS aims at protecting patients' data when stored online, while at the same time enabling doctors to access them either for proposing treatments or perform analytics.

Each user has a unique identifier i. A user u_i will be referred as patient when she is the one who generates a certain file. Each u_i has a public\private key pair. The private key is kept secret, while the public key is shared with the rest of the community. These keys will be used to secure message exchanges. Hence, the communication lines between parties are assumed to be secure. It is also assumed that the public keys of all entities in the system model are known to all registered users.

Function. Within ASCLEPIOS there is a constant interaction between users and the cloud.

Table 51 summarizes users' functions.

Functions
Register to the service

Generate encryption keys to safely protect her data
Store data on the cloud
Share data with other users by creating certain policies using a Ciphertext-Policy Attribute-Based Encryption scheme
Perform Analytics
Revoke access to other users

Table 51: Users' Functions

Sub-Components. We divide users into two different categories. The first category consists of patients who will upload their medical data on the cloud, where they will be stored in a privacy preserving way. On the other hand, the second category consists of doctors and researchers that wish to access theses files without violating the patients' privacy.

Sources. The users interact will all the different components. As a result, every component provides them with data. In

Table **52**, we list all the users' sources.

Sources
Users
CSP
Analytics Layer
Registration Authority
Revocation Authority
Attestation Server
Policy Enforcement Layer

Table 52: Users' Sources

Consumer. Similarly, the users can contact and provide data to all the components.

Table 53 presents a list of users' consumers.

Consumer
Users
CSP
Analytics Layer
Registration Authority
Revocation Authority
Attestation Server
Policy Enforcement Layer

Table 53: Users' Consumers

11.11 ASCLEPIOS's Architecture Walkthrough

ASCLEPIOS aims to create a cloud platform in where will be able to upload and share their medical files with other users. To this end, the patient first encrypts her data using an SSE scheme. The intuition behind this, is to protect the content of the files by both external (e.g. hackers) and internal (e.g. malicious servers) attacks. As a next step, the patient will encrypt the secret SSE key she generated for the encryption of her files using a CP-ABE scheme. The result would be a ciphertext bound by a policy specified by the patient.

The use of the CP-ABE scheme will allow efficient sharing of the symmetric key between multiple users without the need of contacting the patient. For example, a doctor can request the ciphertext of the key, but she will be able to decrypt it if and only if her attributes satisfy the policy specified by the patient. In this way the patient does not need to be online every time a new user registers to the application. By the time the doctor decrypts the ciphertext of the symmetric key - and thus, acquiring the symmetric key – she can start searching directly over the encrypted files, for those containing specific keywords.

Apart from that, ASCLEPIOS will be equipped with a revocation mechanism that will control the access rights of the users depending on their attributes. For example, we assume that a certified doctor will be able to upload new files concerning a certain patient, but a medical student or a researcher will only be able to read files that are already stored online by someone else.

One of the biggest advantages of such an application is that it would allow efficient cross border data sharing. For example, one can think of a scenario in which a patient that is travelling and is in critical health condition, wishes to share her medical history with authorized doctors. The doctors will be able to access the encrypted database and propose a treatment based on the patient's medical records.

Finally, ASCLEPIOS will provide mechanisms enabling analytics to all interested (legitimate) users to perform privacy preserving analytics in order to extract insights from the encrypted data. This will be achieved using an FE technique. Any interested user will contact the corresponding authority requesting for a secret FE key. This process will take place in an isolated environment and thus, all sensitive information will be protected.

We proceed with showing how the different entities communicate with each other. For better understanding we divide the main protocol into five different phases: ***Setup***, ***Initialization, Key Sharing*** and ***Storing***, ***Edit, Revocation***, and ***Analytics***.

Setup Phase. In this phase each entity receives a public/private key pair (pk, sk) for a CCA2 secure public cryptosystem PKE. In addition to that, the entities running in a TEE generate a signing and a verification key pair. Finally, MS runs CPABE.Setup to acquire the master public/private key pair (MPK, MSK) and FE.Setup to acquire the corresponding Functional Encryption public and private parameters (PP, msk).

Initialization Phase. As a first step, a user u_i contacts the RA and requests a secret CP-ABE key (Figure 16). Upon reception of the request, RA authenticates u_i and checks if the user is eligible for receiving such a key (i.e. is not a compromised user, has not generated such a key in the past, etc.). If so, MS generates a CP-ABE key sk_{A,u_i}, encrypts it under pk_i and sends it back to u_i. Now that u_i has successfully received $sk_{\mathcal{A},u_i}$ she can start using the CSP to store files remotely. To do so, she first sends her credentials $cred_i$ and a store request *StoreReq* to the CSP. The CSP authenticates u_i as legitimate and sends back an authorization *Auth* as. At this point, u_i generates a symmetric SSE key K_i to encrypt her files and she send them to the CSP. The initialization phase concludes with u_i encrypting K_i under MPK to get c_K and sends it to KT where it will be stored.

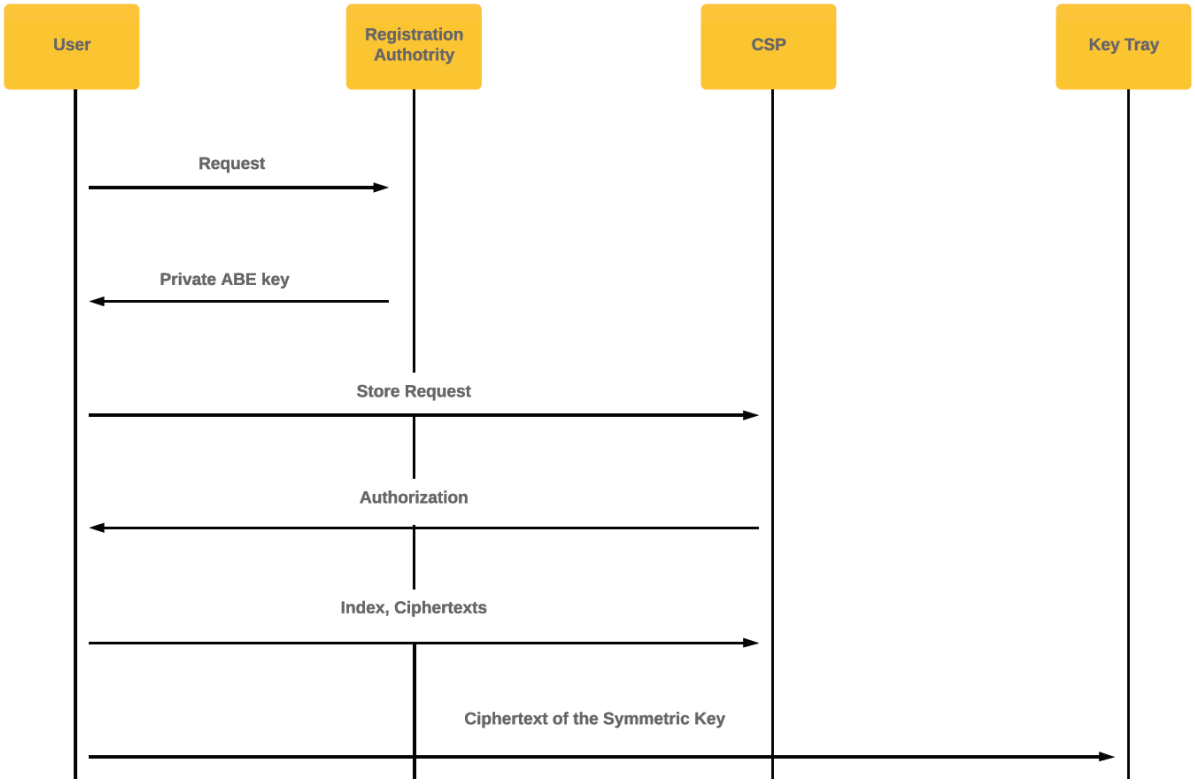

Figure 16: Initialization Phase

Key Sharing Phase. The goal of this phase (Figure 17) is to share data between legitimate users. We assume that the *Initialization Phase* has been successfully completed (i.e. patient u_i has stored encrypted files in a remote location operated by the CSP). This phase commences with a user u_j (e.g. a doctor) wishing to access files stored by u_i in the CSP. At first, u_j has to be verified as a legitimate user by the REV. If u_j is indeed legitimate, she receives back a token τ_{ks} that she will then forward to KT. This way, KT will know that u_i is legitimate and it will reply with c_K. The user will be able to decrypt c_K if and only if her attributes match the policy bounded to c_K. In more detail, the Key Sharing Phase begins with u_i proving that she is not revoked by executing. To this end, u_j contacts REV who will check whether $u_j \in rl$ or not. Assuming that $u_j \notin rl$ REV replies with a verification message that will be forwarded to KT. KT will then send the ciphertext of the symmetric c_K key to u_j who will decrypt it using $sk_{\mathcal{A},u_i}$. Now that u_j has $\mathrm{K_i}$, she can start searching for files stored on the CSP. To do so, she locally runs SSE.SearchToken to generate $\tau_s(w)$ and then sends it to the CSP. Upon reception, CSP will run SSE.Search with $\tau_s(w)$ as input. The output $\boldsymbol{I_w}$ is sent back to u_j.

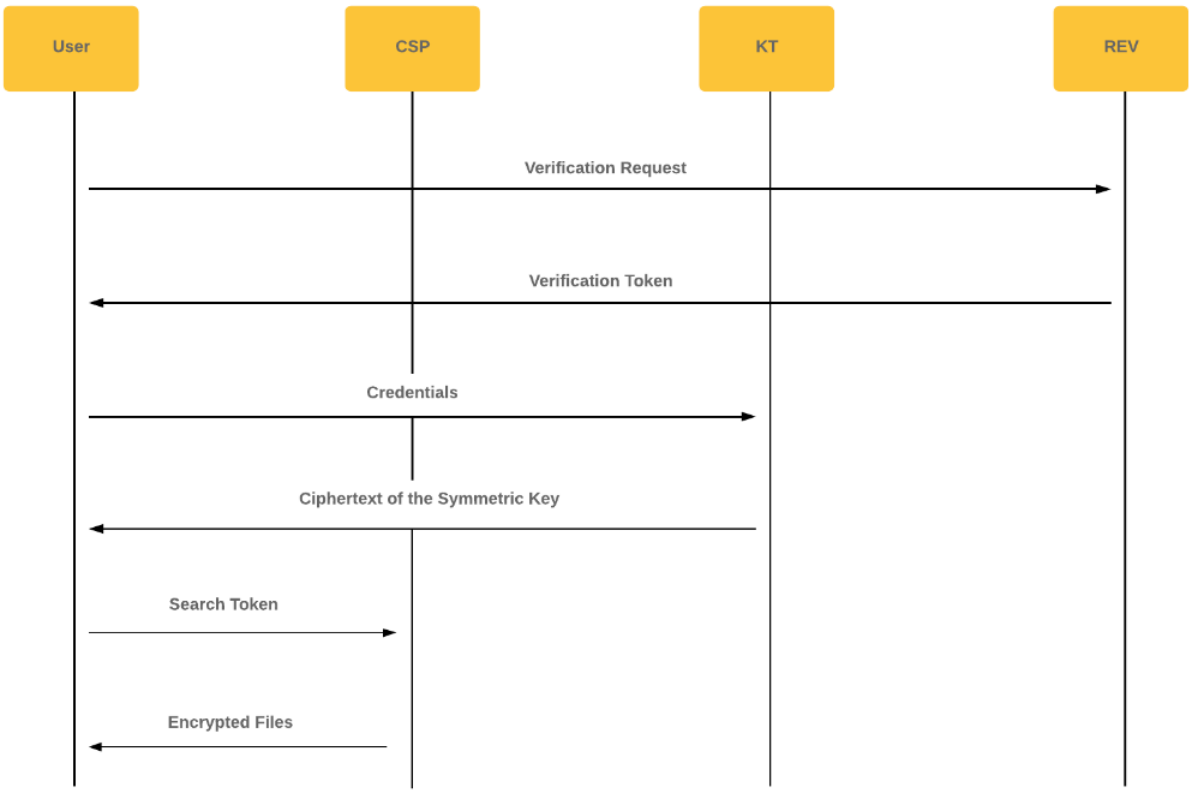

Figure 17: Key Sharing and Search

Editing Phase. In this phase, registered users can add files to the database and data owners can also delete files. In other words, during the editing phase, u_i executes update and delete operations. In particular, to update the database, a registered user u_i first generates an add token by running $(\tau_\alpha(f), c_f) \leftarrow$ SSE.AddToken(K_i, f). This token will be sent to the CSP. Finally, the CSP will execute SSE.Add$(\gamma_i, c_i, \tau_\alpha(f))\rightarrow(\gamma', c_i')$. Similarly, to delete a file from the database, u_i will generate a delete token by running $\tau_d \leftarrow$ SSE.Delete(K_i, f) that will be sent to the CSP. The CSP will proceed by deleting the file specified by the delete token.

Revocation Phase. The last phase of our protocol focuses on the revocation of users (Figure 18) More precisely, we consider the scenario where a data owner u_i wishes to revoke access to a user u_j to whom she had granted permission to access certain files in the past. To this end, u_i first contacts KT in order to prove that she is the owner of K_i , and thus the owner of the files encrypted under K_i. After u_i is authenticated as the owner of K_i, KT sends an acknowledgement to REV containing the identity of u_j. REV will then add u_j to the revocation list *rl*.

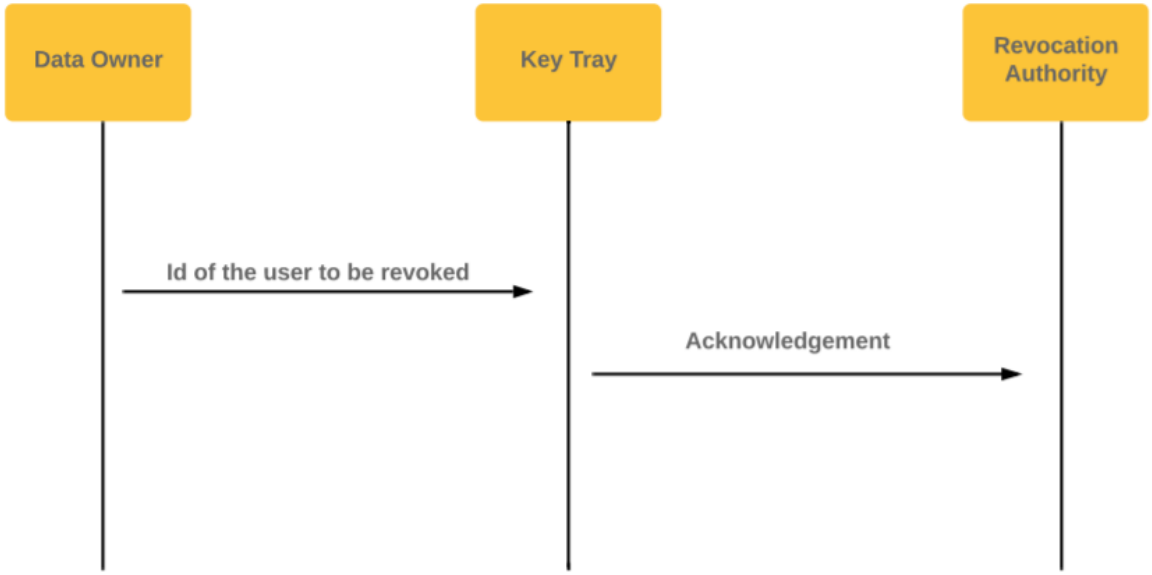

Figure 18: Revocation

Privacy-Preserving Health Data Analytics. For the analytics layer, we make use of a Functional encryption scheme, in order to preserve the privacy of the data processed. In particular, we will make use of the isolated decryption ennvironment (DE) and the isolated functional environment (FE). These two environemnts ideally reside on the same platform, but there are no any serious restrictions concerning that. Upon initialization, DE will establish a secure channel with MS in order to receive the functional encryption secret key (see Figure 19).

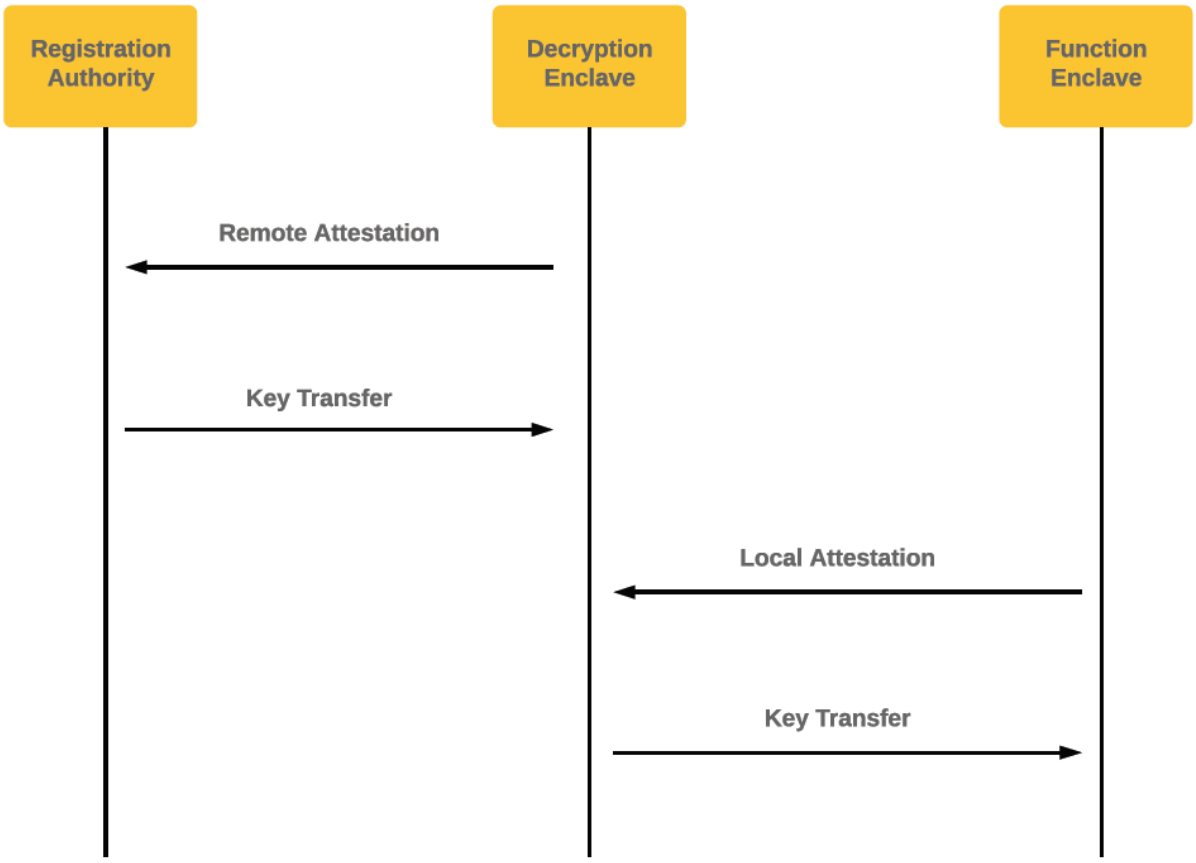

Figure 19: Key Transfer

We assume each user/patient consents that some of her files can be used for analytics. For these files, the user needs to download them, decrypt them and re-encrypt them with a functional encryption key. The re-encrypted files will be sent to the CSP again. As a next step, any user that wishes to perform analytics will contact MS in order to receive authorization for a particular function of her choice. The user can now send the authorization of the function to FE and a request to the CSP for the files are meant to be used for analytics. The CSP will transfer these files over a secure channel to the DE. At that moment, FE locally attests to DE and establishes a secure channel. DE will respond by sending the ciphertexts along with the functional encryption secret key. FE will decrypt the ciphertexts, apply the function on the plaintext, and return the output to the user (Figure 20).

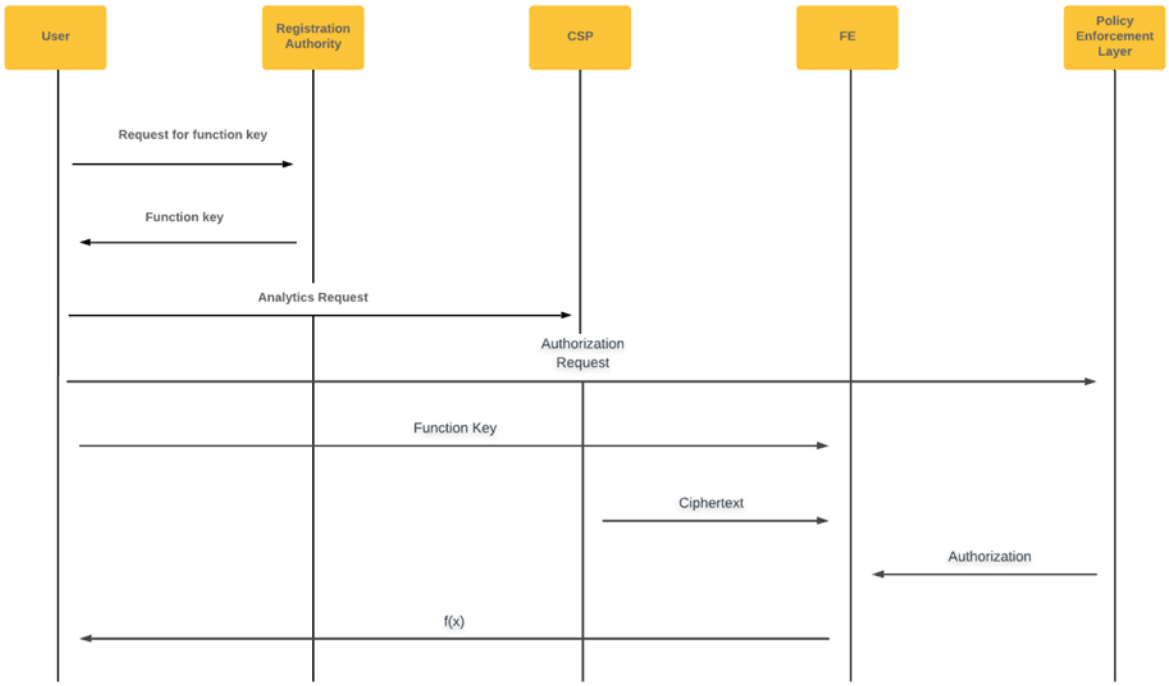

Figure 20: Functional Decryption

11.11.1 Trusted Execution Environment

The security of our construction heavily depends on the use of TEE. More precisely, we are using TEE to create isolated environments and launch trusted entities that would allow a third party to attest and verify their integrity. In the next paragraphs, we provide some basic information regarding the TEE as well as the functionality needed in for secure communication of the different entities.

Isolation. TEE should support the creation of isolated execution environments, in which small pieces of code can be executed in isolation from the rest of the system. Software developers can use these environemnts to create TEEs during OS execution. Such isolated environments rely for their security on the platform's trusted computing base (TCB) code and data loaded at initialization creation time, processor firmware and processor hardware. *Program execution within an isolated environemnt is transparent to both the underlying operating system and other isolated environments.* Multiple mutually distrusting isolated environments should be able to operate on the platform.

Sealing. The TEE should also support sealing of the data stored in the isolated environments. In other words, these envrionments should be able to encrypt and authenticate data that are

stored in untrusted memory. We also require the TEE to be able to recover data even after an isolated environment is destroyed and restarted on the same platform.

Attestation. Attestation can be done either *remotely* or *locally*. The main idea of the remote attestation is that any remote party can verify the integrity and the trustworthiness of an entity. On the other hand, local attestation is taking place between isolated environments that are part of the same platform. We assume that each time such environments communicate with each other, they first run a remote or local attestation protocol depending on whether they are on the same platform or not.

We now proceed with a formal definition of the hardware (HW) as described in (31).

Definition (Secure HW functionality). A secure hardware functionality HW for a class of probabilistic polynomial time programs $\mathcal{P}$ consists of the following interface:

1. **HW.Setup(1^{λ}).** HW.Setup takes as input a security parameter λ and it generates the secret keys $sk_{\{quote\}}$ and $sk_{\{report\}}$. Finally. It generates and outputs public parameters $params$.

2. **HW.Load($params, Q$).** This loads a program into an isolated environment. HW.Load takes as input a program $Q \in \mathcal{P}$ and some global parameters $params$. It first creates an isolated envronment and loads Q and generates a handle h that will be used to identify the environment running Q.

3. **HW.Run($h\ in$).** This runs an program in an isolated environemtn. It takes in a handle h corresponding to the environment running the program Q and an input in and outputs an output out.

4. **HW.Run&Report(h, in).** This executes a program in an isolated environment and also generates an attestation of tis output that can be verified by another program on the same HW platform. It takes as input a handle h for an environment running a program Q and input for Q. The algorithm first executes Q on in to get out. HW.Run&Report outputs the tuple $report := (md_h, tag_Q, in, out, mac)$ where md_h is the metadata associated with the isolated environment, tag_Q is a program tag that can be used to identify the program running inside the environement (it can be a cryptographic hash of the program code Q) and mac is a cryptographic MAC produced using $sk_{\{report\}}$.

5. **HW.Run&Quote(h, in).** This executes a program in an isolated environment and also generates an attestation of tis output that can be publicly verified. It takes as input a handle h for an environment running a program Q and input for Q. The algorithm first executes Q on in to get out. HW.Run&Report outputs the tuple $quote := (md_h, tag_Q, in, out, \sigma)$ where md_h is the metadata associated with the environment, tag_Q is a program tag that can be used to identify the program running inside the enviornment and σ is a signature on (md_h, tag_Q, in, out).

6. **HW.ReportVerify($h', report$).** This is the report verification algorithm. It takes as input a handle h' for an isolated environment and a $report := (md_h, tag_Q, in, out, mac)$. It uses $sk_{\{report\}}$ to verify the MAC. If MAC is valid, it outputs 1. Otherwise it outputs 0.

7. **HW.QuoteVerify($params, quote$).** This is the quote verification algorithm. It takes as input $params$ and a $quote := (md_h, tag_Q, in, out, \sigma)$.It outputs 1 if the verification of σ succeeds and 0 otherwise.

12 Conclusions

The scope of this deliverable was to provide a high-level but detailed description of the ASCLEPIOS Reference Architecture along with its main components, mechanisms, algorithms and models, the interconnection scheme and the specific interfaces for exchanging information among them. ASCLEPIOS Reference Architecture aims to satisfy the different types of requirements that have been formulated during the requirements analysis that took place in D1.1 (39). More specifically, deliverable *D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements* highlighted specific functional and security requirements by evalutating the needs of both healthcare and technical partners. By analyzing these requirements, the goal of the current deliverable was to identify all stakeholders and as many as possible functionalities that would be required towards the formulation of ASCLEPIOS services. To this end, specific roles have been identified and specific functionalities have been described.

Furthermore, an analysis of the project's use cases have been described along with the implementation scenarios of the mechanisms that will be developed within the project. Both the description of the architecture and of the use cases has been nicely coupled with the security and health requirements that have been collected and described in D1.1.

Finally, it is worth mentioning that this deliverable will be used as our guide during the design, development and implementation of core functions of ASCLEPIOS.

13 References

1. Cockburn, A. Use Case fundamentals [Internet]. Available from: http://alistair.cockburn.us/Use+case+fundamentals

2. Cavadenti, A. eDREAM project. WP2 – User Requirements, Use Cases and System Specification. DELIVERABLE: D2.2 Use Case Analysis and application scenarios description V1 [Internet]. Available from: http://edream-h2020.eu/wp-content/uploads/2018/09/eDREAM.D2.2.ASM_.WP2_.V1.0.pdf

3. Saver, J.L. Time is brain - quantified. Stroke. 2006;37.1:263–6.

4. HIPAA. Break Glass Procedure: Granting Emergency Access to Critical ePHI Systems [Internet]. Available from: https://hipaa.yale.edu/security/break-glass-procedure-granting-emergency-access-critical-ephi-systems

5. Gerrits, van der Werf, Kloppenburg, de Ruijter, Hard voor Hersenen. Tijdregistratie Herseninfarct. 2019.

6. Bradley, E., Holmboe, E., Mattera, J., Roumanis, S., Radford, M., Krumholz, H. Data feedback efforts in quality improvement: lessons learned from US hospitals. Qual Saf Health Care. 2004;(13):26–31.

7. van der Veer, S.N., de Keizer, N.F., Ravelli, A.C.J., Tenkink, S., Jager, K.J. Improving quality of care. A systematic review on how medical registries provide information feedback to health care providers. Int J Med Inf. 2010;79:305–23.

8. WHO International Working Group for Drug, WHO Collaborating Centre for Drug Statistics Methodology. Introduction to drug utilization research. 2003.

9. Meeker, D., Linder, J.A., Fox, C.R., Friedberg, M.W., Persell, S.D., Goldstein, N.J., Knight, T.K., Hay, J.W., Doctor, J.N. Effect of Behavioral Interventions on Inappropriate Antibiotic Prescribing Among Primary Care Practices: A Randomized Clinical Trial. JAMA. 2016;(315):562–70.

10. Eden, A.R., Hansen, E., Hagen, M.D., Peterson, L.E. Physician Perceptions of Performance Feedback in a Quality Improvement Activity. Am J Med Qual. 2018;33:283–90.

11. Holmes, J., Soualmia, L., Séroussi, B. A 21st Century Embarrassment of Riches: The Balance Between Health Data Access, Usage, and Sharing. Yearb Med Inf. 2018;27:005–6.

12. Gjelstad, S., Høye, S., Straand, J., Brekke, M., Dalen, I., Lindbæk, M. Improving antibiotic prescribing in acute respiratory tract infections: cluster randomised trial from Norwegian general practice (prescription peer academic detailing (Rx-PAD) study). BMJ. 2013;347:f4403.

13. Gerber, J.S., Prasad, J.S., Fiks, A.G., Localio, A.R., Bell, L.M., Keren, R., Zaoutis, T.E. Durability of Benefits of an Outpatient Antimicrobial Stewardship Intervention After Discontinuation of Audit and Feedback. JAMA. 2014;312:2569–70.

14. Linder, J.A., Meeker, D., Fox, C.R., Friedberg, M.W., Persell, S.D., Goldstein, N.J., Doctor, J.N. Effects of Behavioral Interventions on Inappropriate Antibiotic Prescribing in Primary Care 12 Months After Stopping Interventions. JAMA. 2017;318:1391–92.

15. Aggarwal, C.C., Yu, P.S. A General Survey of Privacy-Preserving Data Mining Models and Algorithms. In: Aggarwal, C.C., Yu, P.S., editor. Privacy-Preserving Data Mining. New York, USA: Springer; 2008. p. 11–52.

16. Kantarcioglu, M. A survey of privacy-preserving methods across horizontally partitioned data. In: Aggarwal, C.C., Yu, P.S., editor. New York, USA: Springer; 2008. p. 313–35.

17. Vaidya, J. A survey of privacy-preserving methods across vertically partitioned data. In: Aggarwal, C.C., Yu, P.S., editor. Privacy-Preserving Data Mining. New York, USA: Springer; 2008. p. 337–358.

18. Lindell, Y., Pinkas, B. Secure multiparty computation for privacy-preserving data mining. J Priv Confidentiality. 2009;1:5.

19. Čižman, M. The use and resistance to antibiotics in the community. Int J Antimicrob Agents. 2003;21(4):297–307.

20. O'Neill, J. Tackling drug-resistant infections globally: final report and recommendations, Review on Antimicrobial Resistance. 2016; Available from: https://amr-review.org/sites/default/files/160525_Final%20paper_with%20cover.pdf

21. Ministry of Health and Care Services. Handlingsplan mot antibiotikaresistens i helsetjenesten. Oslo, Norway; 2015.

22. Bellika, J. G., Henriksen, T. S., Yigzaw, K. Y. The Snow system - a decentralized medical data processing system. In: Data Mining in Clinical Medicine. Llatas, C. F., García-Gómez, J. M. Springer; 2014.

23. Hailemichael, M. A., Yigzaw, K. Y., Bellika, J. G. Emnet: a tool for privacy-preserving statistical computing on distributed health data. In Linköping, Sweden; 2015. p. 33–40.

24. Yigzaw, K.Y., Hailemichael, M.A., Skrøvseth, S.O., Bellika, J.G. Secure and scalable computation of the mth-ranked element on distributed data. In [under revision]; 2018.

25. El Emam, K., Jonker, E., Arbuckle, L., et al. A Systematic Review of Re-Identification Attacks on Health Data. PLoS ONE. 2011;6:e28071.

26. Ministry of Health and Care Services. Lov om behandling av helseopplysninger ved ytelse av helsehjelp (Act on processing of health information when performing healthcare) [Internet]. 2014. Available from: https://lovdata.no/lov/2014-06-20-42/§6

27. International Household Survey Network. DDI metadata standard [Internet]. Available from: http://www.ihsn.org/projects/DDI-standard

28. Fisch B, Vinayagamurthy D, Boneh D, Gorbunov S. IRON. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17 [Internet]. ACM Press; 2017; Available from: http://dx.doi.org/10.1145/3133956.3134106

29. Papaioannou A. 2005, Cryptography, Athens:NTUA

30. Michalas A. The lord of the shares. Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing – SAC '19 [Internet]. ACM Press; 2019; Available from: http://dx.doi.org/10.1145/3297280.3297297

31. Boneh D, Sahai A, Waters B. Functional Encryption: Definitions and Challenges. Lecture Notes in Computer Science [Internet]. Springer Berlin Heidelberg; 2011;253–73. Available from: http://dx.doi.org/10.1007/978-3-642-19571-6_16

32. Paladi N, Gehrmann C, Michalas A. Providing User Security Guarantees in Public Infrastructure Clouds. IEEE Transactions on Cloud Computing [Internet]. Institute of Electrical and Electronics Engineers (IEEE); 2017 Jul 1;5(3):405–19. Available from: http://dx.doi.org/10.1109/tcc.2016.2525991

33. Boneh D, Sahai A, Waters B. Functional encryption. Communications of the ACM [Internet]. Association for Computing Machinery (ACM); 2012 Nov 1;55(11):56. Available from: http://dx.doi.org/10.1145/2366316.2366333

34. Coker G, Guttman J, Loscocco P, Herzog A, Millen J, O'Hanlon B, et al. Principles of remote attestation. International Journal of Information Security [Internet]. Springer Science and Business Media LLC; 2011 Apr 23;10(2):63–81. Available from: http://dx.doi.org/10.1007/s10207-011-0124-7

35. C. Chatman, "How cloud computing is changing the face of health care information technology," J Health Care Compliance, vol. 12, pp. 37–70, 2010.

36. Antonis Michalas and Noam Weingarten. "HealthShare: Using Attribute-Based Encryption for Secure Data Sharing Between Multiple Clouds". Proceedings of the 30th IEEE International Symposium on Computer-Based Medical Systems (CBMS'17), Thessaloniki, Greece, 2017.

37. Antonis Michalas, Nicolae Paladi and Christian Gehrmann. "Security Aspects of e-Health Systems Migration to the Cloud". Proceedings of the 16th IEEE International Conference on E-health Networking, Application & Services (Healthcom), October 15 - 18, 2014, Natal, Brazil.

38. Kassaye Yitbarek Yigzaw, Antonis Michalas and Johan Gustav Bellika. "Secure and Scalable Deduplication of Horizontally Partitioned Health Data for Privacy-Preserving Distributed Statistical Computation". Journal of Medical Informatics and Decision Making (BMC), 2017

39. Y. Kassaye et.. al. ASCLEPIOS Deliverable D1.1: ASCLEPIOS Technical, Security, Healthcare and Data Privacy Requirements, 2019.

40. HashiCorp. Vagrant [Internet]. Available from: https://www.vagrantup.com/

41. Mell, P., Grance, T. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. 2011. Available from: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

42. Docker. Enterprise Container Platform for High-Velocity Innovation [Internet]. Available from: https://www.docker.com/

14 Annex I. Demonstrator template

14.1 Introduction

<Provide introduction to the section.>

14.2 Background

<Provide sufficient details about the demonstrator, including the involved actors (from Table 1), and specifications of the medical devices used if there is any.>

14.2.1 Motivation

<Describe a short description of the motivation of the demonstrator>

14.2.2 State-of-the-art

<Describe the current situation before the demonstrator is implemented>

14.2.3 Envisioned situation

<Describe the envisioned situation after the implementation of the demonstrator. Describe as well an expected impact of the ASCLEPIOS platform to patients, healthcare organizations, and public>

14.3 Use cases

Table A is a summary table for the use cases developed for the demonstrator.

ID	Name

Table A: Demonstrator Y use cases

Figure A shows the use case diagram for the demonstrator.

<Figure A>

Figure A: Demonstrator Y use case diagram

14.3.1 <use case title>

<Please use the subsection structure for describing each of the use cases>
<Add introduction to the use case and reference to the use case description in Table B.>

Attribute	Description
Use case ID	<ID uniquely identifying the use case>
Use case goal	<An attainable goal for the use case>
Assumptions & pre-conditions	<Assumptions generally do not change during the execution and should be true to successfully terminate the use case. Pre-

	conditions define all the conditions that must be met to meaningfully cause the initiation of the use case>
Use case initiation	<Potential triggers or events initiating the use case>
Use case main scenario	<Description of steps for the defined use case in a clear concise manner>
Use case alternate scenario	<Alternative scenario for the use case if any>
Use case failure scenario	<Various error conditions that can happen and the actions to resolve them>
Acceptance criteria	<The criteria in order to accept the use case as complete>

Table B: Use case description summary

<Provide additional description here if you want to provide more information than what is possible in the table.>

14.4 Components/ mechanisms of ASCLEPIOS framework involved in demonstrator

Components/mechanisms of the ASCLEPIOS involved in the demonstrator:

- *Use of Cloud/HPC resources*
- *Data sharing and revocation using SSE and ABE*
- *Privacy-Preserving analytics using FE*
- *Medical device hardware integrity*
- *Cloud provider integrity*
- *Increase GDPR and Security Awareness*

14.5 Demonstrator security requirements

The security and privacy requirements of the use cases are summarized in Table C.

<Create a table containing the security and privacy requirements (i.e., confidentiality, integrity, secure software execution, hardware security, and private data sharing) for all of the use cases with a reference to the requirements specified in D1.1.>

Use case	**Requirement**	**ASCLEPIOS functionalities**
<Use case ID>	Requirement 1	
	Requirement 2	

Table C: Summary of the security and privacy requirements for all use cases

14.6 Demonstrator data requirements

<Describe the data requirements of the demonstrator, such as the data structure (e.g., relational database, HL7 FHIR). Identify the data requirements that applies to the demonstrator from D1.1.>

14.7 Testbed

<Describe the planned infrastructure on which the demonstrator will/should be deployed, for example public cloud, private cloud, across servers installed in a simulated environment.>