SafeCloud

Design and Requirements, Maxdata SafeCloud-based healthcare platform

D5.2

Project reference no. 653884

August 2016

European Commission

Horizon 2020
European Union funding
for Research & Innovation

Document information

Scheduled delivery	01.09.2016
Actual delivery	01.09.2016
Version	1.0
Responsible Partner	Maxdata Software, S.A.

Dissemination level

Public

Revision history

Date	Editor	Status	Version	Changes
01.05.2016	P.Sousa	Draft	0.1	Initial version
31.05.2016	P.Sousa	Draft	0.2	Added contribution from INESC ID
07.06.2016	P.Sousa	Draft	0.3	Added contribution from INESC TEC
13.06.2016	P.Sousa	Draft	0.4	Ready for 1st internal review
12.07.2016	P.Sousa	Draft	0.5	Added contributions and feedback from CYBER, INESC TEC, INESC ID, TUM and UniNE. Ready for final internal review.
12.08.2016	P.Sousa	Final	1.0	Final release

Contributors

Miguel Correia (INESC ID)
Bruno Ferreira (Maxdata)
Francisco Maia (INESC TEC)
Hugues Mercier (UniNE)
Bernardo Portela (INESC TEC)
Paulo Sousa (Maxdata)

Internal reviewers

Miguel Pardal (INESC ID)
Karl Tarbe (CYBER)
Sree Harsha Totakura (TUM)
Francisco Maia (INESC TEC)
João Paulo (INESC TEC)

Acknowledgements

This project is partially funded by the European Commission Horizon 2020 work programme under grant agreement no. 653884.

More information

Additional information and public deliverables of SafeCloud can be found at http://www.safecloud-project.eu

Glossary of acronyms

Acronym	Definition
EU	European Union

Table of contents

Executive summary

This deliverable describes the design and requirements of the Maxdata SafeCloud-based healthcare platform.

1 Introduction

The objective of work package 5 (WP5) is to deploy SafeCloud in order to provide its novel privacy features to new and improved products and services. Two main use cases will be demonstrated: a cloud storage platform and a healthcare platform.

This deliverable (D5.2) regards the healthcare platform. We present the design and requirements of the healthcare platform that will be built by Maxdata on top of the SafeCloud framework.

The deliverable is structured as follows.

- Section 2 revisits the SafeCloud framework fully described in deliverables D1.1, D2.1, D3.1 and D4.1.
- Section 3 presents the design of the healthcare platform. This platform will be based on CLINIdATA®, an eHealth web application owned by Maxdata. Section 3 is divided in 2 main parts:
    - Description of CLINIdATA®, including its main use cases.
    - Identification of the major deployment scenarios of CLINIdATA®, and how these scenarios will be tackled using the components of the SafeCloud framework presented in Section 2.
- Section 4 describes the relevant requirements to be addressed by the SafeCloud framework taking into consideration the use cases and scenarios presented in Section 3.
- Section 5 concludes the deliverable.

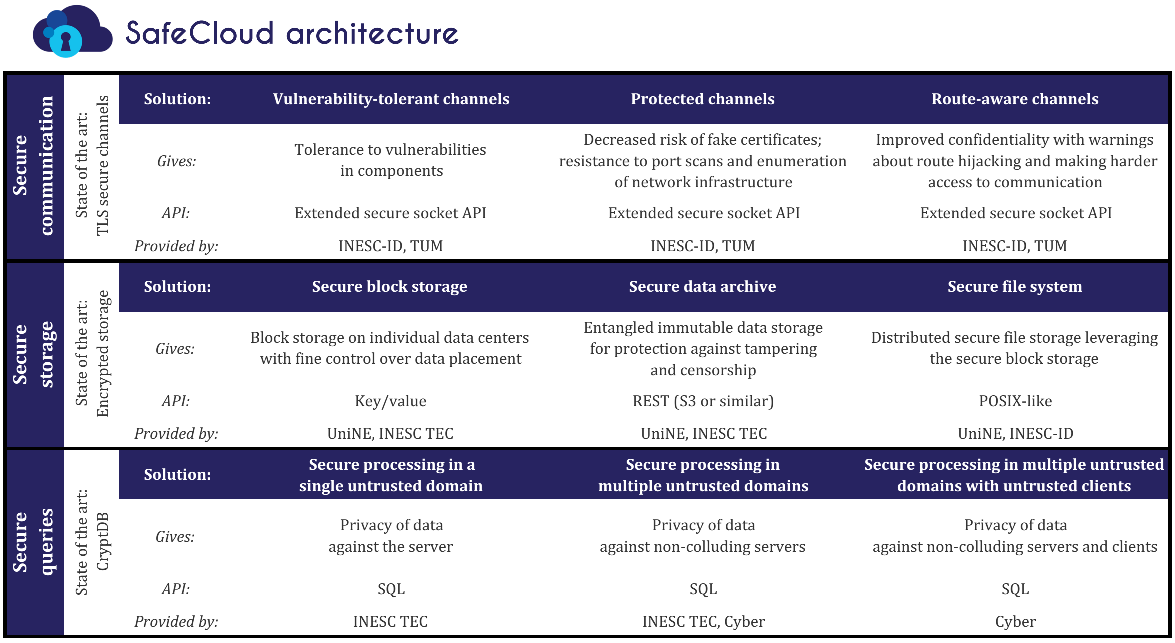
2 SafeCloud Framework

This chapter briefly describes the SafeCloud framework presented in deliverables D1.1, D2.1, D3.1 and D4.1.

2.1 SafeCloud architecture

In the current technology landscape, one of the major challenges we face concerns *data privacy*. In fact, companies, governments and even users in general are now constantly offloading the control and responsibility of their data to third-party cloud service providers. Such a trend raises a number of privacy and security problems since the actual owner of the data is no longer in control of where is it located or even of who can access it. These concerns were recently confirmed to be well justified after the recent media hype surrounding the revelations made by former NSA contractor Edward Snowden and high-profile security vulnerabilities such as the Heartbleed bug in OpenSSL.

In this context, the SafeCloud project aims at going beyond the state-of-the-art with respect to the offering of cloud-enabled data privacy solutions. It does so in a three-layer approach. As depicted in Figure 1, each layer provides a set of solutions with distinct security guarantees. In general, stricter security guarantees impose greater limitations or performance costs on the applications using the solutions.



SafeCloud architecture

Secure communication	State of the art: TLS secure channels	Solution:	Vulnerability-tolerant channels	Protected channels	Route-aware channels
		Gives:	Tolerance to vulnerabilities in components	Decreased risk of fake certificates; resistance to port scans and enumeration of network infrastructure	Improved confidentiality with warnings about route hijacking and making harder access to communication
		API:	Extended secure socket API	Extended secure socket API	Extended secure socket API
		Provided by:	INESC-ID, TUM	INESC-ID, TUM	INESC-ID, TUM
Secure storage	State of the art: Encrypted storage	Solution:	Secure block storage	Secure data archive	Secure file system
		Gives:	Block storage on individual data centers with fine control over data placement	Entangled immutable data storage for protection against tampering and censorship	Distributed secure file storage leveraging the secure block storage
		API:	Key/value	REST (S3 or similar)	POSIX-like
		Provided by:	UniNE, INESC TEC	UniNE, INESC TEC	UniNE, INESC-ID
Secure queries	State of the art: CryptDB	Solution:	Secure processing in a single untrusted domain	Secure processing in multiple untrusted domains	Secure processing in multiple untrusted domains with untrusted clients
		Gives:	Privacy of data against the server	Privacy of data against non-colluding servers	Privacy of data against non-colluding servers and clients
		API:	SQL	SQL	SQL
		Provided by:	INESC TEC	INESC TEC, Cyber	Cyber

Figure 1: SafeCloud architecture.

The first layer in the SafeCloud architecture is focused on secure communications. This layer provides solutions that improve the security of communication channels over untrusted environments. SafeCloud proposes three solutions for this layer. First, the vulnerability-tolerant channels solution provides communication channels that are built on multiple redundant security mechanisms to ensure that failure of any one mechanism does not cause a security failure in the channel. Second, the protected channels solution introduces multiple methods to reduce the risk of fake certificates used by the parties and also makes it more difficult to run port scans and do enumeration of the network infrastructure. Third, the route-aware channels solution deploys methods to improve confidentiality and detect route hijacking. All of the solutions are built on top of the Java secure socket API and they form the Private Communication Middleware Architecture.

The second layer addresses data storage security and consists of solutions that provide confidentiality and integrity guarantees for data stored in an untrusted environment. The secure block storage and the secure file system give similar secure storage benefits but with different level APIs for client applications. The secure data archive provides an entangled immutable data store for protection against tampering and censorship. It is exposed to applications as a REST API.

Finally, the third layer proposes solutions that explore trade-offs between data privacy and the ability to do useful processing over it in untrusted environments, i.e., they differ in degrees of which data is kept private and what queries can be performed in the untrusted domain. All three solutions expose a SQL interface to the user. However, each one of the solutions considers a different security model, which are intended to serve different application needs. The first solution considers a single untrusted domain while solutions two and three consider multiple untrusted domains. Furthermore, solution three distinguishes itself from solution two as it additionally considers multiple simultaneous untrusted clients.

2.2 Private communication middleware architecture

With increasing number of attacks surfacing in the Internet, Internet communications are an important source of concern for privacy and confidentiality. These concerns are addressed by the work package WP1 with its objective to provide middleware services which improve the privacy and security of Internet communications in the SafeCloud architecture. These services provide the properties of secure channels – confidentiality, integrity, and authenticity – together with availability, but assuming powerful adversaries who may be able to break some of the assumptions that make existing channels insecure.

In month 6, WP1 provided Deliverable D1.1 that presents a preliminary design of the secure communication middleware: the threats it assumes, the service it will provide, and its architecture. In that document it is possible to grasp some of the details on the three services the middleware will provide: (1) Vulnerability-tolerant channels that will provide secure communication even if vulnerabilities are discovered in some of their mechanisms; (2) Protected services, which extend solution (1) with protected service provisioning; and (3) Route-aware channels, which include solution (2) and extend it with route monitoring and multi-path communication.

2.3 Storage architecture

As described in D2.1 and delivered at Month 6, the storage architecture provides three solutions in increasing levels of sophistication: secure block storage, a secure data archive, and a secure file system.

The **secure block storage** solution consists of a data store that can store raw data under a given key. In that respect, it behaves akin to a key/value store but provides additional mechanisms to ensure data security and integrity (using cryptographic techniques), as well as explicit placement of data items. As a matter of fact, being able to place different parts of data items in various geographical locations within distinct administrative domains is key to providing privacy in the SafeCloud platform. The secure block storage operates locally, on a per node basis, and there are typically several instances per data centre. Orchestration between these instances is performed by separate components that can explicitly place data items on individual instances of the block storage.

The **secure data archive** builds on top of the secure block storage and supports secure storage of documents over many distributed instances of data stores. Documents are redundantly stored and protected against tampering using coding and entanglement techniques, i.e., they are encoded and combined with previous documents to ensure that no party can modify or delete them (without affecting a significant portion of all other documents). The data stored in the system is immutable, as the key idea behind entanglement is to persist documents over the long term. In other words, this solution supports archival storage. Modifications to existing data can be implemented on top of the long-term distributed encrypted document storage by the means of a versioning API, i.e., by inserting a new version of a previous document under the same name but a with different version identifier.

The **secure file system** provides a file system API on top of the secure block storage. It supports secure reading and writing of files that are geographically distributed across data centres, and hence deals with mutable data. The file system is optimized in terms of latency and throughput for sequential accesses. Placement of data in the file system can be guided by policies that express security or dependability requirements (e.g., replication degree, disaster tolerance against whole data centre failure, geo-localisation in a given set of countries, etc.). As the file system is accessible locally from clients, it requires a local component to execute directly on the client machines.

2.4 Privacy-preserving storage and computation architecture

Modern application and service deployments can fall into one of three major categories: (1) In-house deployments, where everything from IT infrastructure to software is designed, developed and maintained within the company or organization facilities; (2) Remote deployments, where everything is handed over to a Cloud or a set of Cloud providers, and the organization/company manages its infrastructure remotely; or (3) A compromise between the previous two, in which critical systems and services remain within trusted premises, and complementary services are off-loaded to convenient Cloud providers.

The latter is the “best of two worlds” solution, which allows for the system to benefit from the maximum potential of the Cloud provider, without jeopardizing information security. This option, however, requires a case-specific analysis of security and functional requirements. The general architecture for SafeCloud solutions to privacy-preserving storage and computation is heavily rooted in the Cloud Computing paradigm, focused in exploring the approach of this third category, comprising a set of comprehensive solutions to private and secure data storage and computation.

The SafeCloud architecture for privacy-preserving storage and computation detailed in Deliverable D3.1 is comprised of two domains: the trusted deployment and the untrusted deployment, communicating with the client via a client interface, and with each other via a proxy. These distinctions allow for clear specification of client API (either SQL or NoSQL), computational requirements associated with computation performed in the local and remote components, and how the client side interacts with the cloud provider(s). SafeCloud evades the problems inherent to a single-scenario focused approach by proposing several solutions meeting different functional and security requirements. All solutions will provide the end-user with a SQL interface. However, for meeting certain performance levels without sacrificing security guarantees, the actual language coverage of each respective solution will be individually specified. Additionally, solutions will also be devised to support specific classes of applications that benefit from NoSQL interfaces, therefore extending the scope of our framework. More precisely, we aim to provide a system that offers a full ANSI SQL API, built on top of a NoSQL database in a scalable design.

The first solution considers a single untrusted domain, i.e., a single cloud provider. This comprises the simplest deployment possibility, either employing standard cryptographic techniques to securely store remote information (albeit effectively disabling the possibility of executing any meaningful computations over this data), or applying different property-preserving cryptographic techniques according to varying levels of data sensitivity. This solution is expected to provide full ANSI SQL, as it translates SQL queries into NoSQL statements, to be processed by the underlying database system.

The second solution considers the possibility of accessing multiple untrusted domains. This opens the possibility to employ more complex cryptographic mechanisms, allowing for several cloud providers to execute computations over encrypted data without any individual provider gaining any information about the associated plaintext values (assuming these providers do not collude). The trusted deployment is expected to be very similar to the one described in the previous solution, providing full ANSI SQL.

The third solution allows for untrusted clients to access multiple remote domains. This model allows for various users to upload their sensitive data over several cloud providers, and then jointly perform computations over the full dataset. Contrary to the second solution, this is not restricted to a single data provider, therefore enabling for the deployment of solutions such as joint medical data analysis, which are typically performed over sensitive data. This final solution is not expected to implement the full SQL standard, since some parts of it would be infeasible for practical applicability.

3 Maxdata Healthcare Platform

This chapter describes the healthcare platform that will be built in SafeCloud. This platform will be based on CLINIdATA®, an eHealth web application owned by Maxdata.

CLINIdATA® is composed of several modules, including a healthcare laboratory information system and an epidemiological surveillance system dedicated to supporting Hospital Infection Control Committees. The chapter is divided in 2 main parts:

- Sections 3.1 and 3.2: Description of CLINIdATA®, including its main use cases.
- Section 3.3: Identification of the major deployment scenarios of CLINIdATA®, and how these scenarios will be tackled using the components of the SafeCloud framework presented in Section 2.

3.1 CLINIdATA® eHealth Solution

3.1.1 Product context

CLINIdATA® addresses different types of clinical and non-clinical organizations. In the clinical domain, CLINIdATA® is used by various healthcare organizations, including hospitals, clinics, laboratories, and primary care units. In the non-clinical domain, CLINIdATA® is used in laboratories of many areas, including water, food, and environmental health.

Maxdata current approach is to install its software on customer premises. This approach has allowed Maxdata to conquer most of the Portuguese market - currently present in more than 80% of national public hospitals -, but now the company wants to sell its products abroad and one of the ways is to deploy them on the cloud and sell them as a service (SaaS).

Architecture-wise, CLINIdATA® is a web application composed of 3 layers:

- **Presentation layer**: Includes the presentation logic and runs on any common browser (e.g., Google Chrome, Mozilla Firefox, Microsoft Internet Explorer). HTML and JavaScript code is generated automatically using the Google Web Toolkit[1];
- **Business logic layer**: Set of services, running on the server side, that implement the business logic. These services are implemented in Java using the Spring Framework[2];
- **Database access layer**: Set of methods, running on the server side, used to access the relational database where data is persisted. These methods use the Hibernate ORM Framework[3] to abstract the concrete DBMS (e.g., Oracle[4], PostgreSQL[5]) being used.

CLINIdATA® is used by different types of organizations, ranging from small laboratories with a few dozens of professionals and hundreds of transactions per day, to very large hospital clusters with thousands of professionals and tens of millions of transactions per day. Table 1 shows the typical workloads of large versus small organizations in terms of users, analyzers, exams, storage, and database transactions.

[1] http://www.gwtproject.org

[2] https://projects.spring.io/spring-framework

[3] http://hibernate.org/orm/

[4] https://www.oracle.com/database/index.html

[5] http://www.postgresql.org

	Large organizations	Small organizations
Users	1,000 – 8,000	20 – 200
Analyzers	70 -100	10 – 20
Exams per day	20,000 – 30,000	500 – 1,500
Storage size (5 years period)	1 TB	50 – 100 GB
DB transactions (normal periods)	7 – 14 Tx per sec	0.2 – 0.8 Tx per sec
DB transactions (peak periods)	40 – 70 Tx per sec	1 – 4 Tx per sec

Table 1: Comparison of CLINIdATA® workloads on large versus small organizations.

3.1.2 Product capabilities/functions

CLINIdATA® is an eHealth web application that includes the following main features:

- 100% web application - no plugins required on the client side, only a web browser is need to run the application;
- Cross-platform application where server components may run on any common operating system (e.g., Linux, Mac OS X, Solaris, Windows) and relational database (e.g., MySQL, PostgreSQL, Oracle, SQL Server);
- Overall management and control of clinical laboratories, including technical and financial aspects of all areas of Clinical Pathology (e.g., clinical chemistry, hematopathology, microbiology, immunology) and Anatomic Pathology (e.g., histopathology, cytopathology, immunohistochemistry);
- Overall management and control of different types of non-clinical laboratories, e.g., water, air, food;
- Management of the entire exam process in clinical and non-clinical laboratories, including the 3 typical phases:
  - **Pre-analytical**: Prescription, specimen collection, specimen transport;
  - **Analytical**: Exam realization, quality control, results validation/approval;
  - **Post-analytical**: Access to exam results by patients & clinicians, billing.
- Epidemiological surveillance tool dedicated to supporting Hospital Infection Control Committees in the prevention, identification and monitoring of infections;
- Real-time interface that supports more than 300 different automated analyzers;
- Integration with dozens of other clinical and non-clinical information systems (e.g., intensive care unit, patient identification, billing, regional health portals);
- Rules engine, allowing the incorporation of intelligence and safety throughout the whole application;
- Business Intelligence (BI) platform that allows to view and explore statistics through dashboards and other visual elements.

3.1.3 User characteristics

In clinical and non-clinical organizations, CLINIdATA® is used by two main types of users:

- Organization professionals, e.g., healthcare professionals, that access most of the features described in Section 3.2;
- Organization customers, e.g., patients, that access exam results.

3.1.4 Constraints

When deployed in healthcare organizations, CLINIdATA® manages personal data, so it should be compliant with regulations on personal data protection (e.g., in Europe - General Data Protection Regulation[6], in the USA - Health Insurance Portability and Accountability Act[7]).

3.2 CLINIdATA® Use Cases

3.2.1 Overview

This section presents an overview of CLINIdATA® use cases and their actors taking into account not only the already existing CLINIdATA® features, but also the 3 scenarios presented in Section 3.3 that will be developed in the SafeCloud project. The use cases are summarized in Figure 2.

[6] http://ec.europa.eu/justice/data-protection/

[7] http://www.dhcs.ca.gov/formsandpubs/laws/hipaa/Pages/1.00WhatisHIPAA.aspx

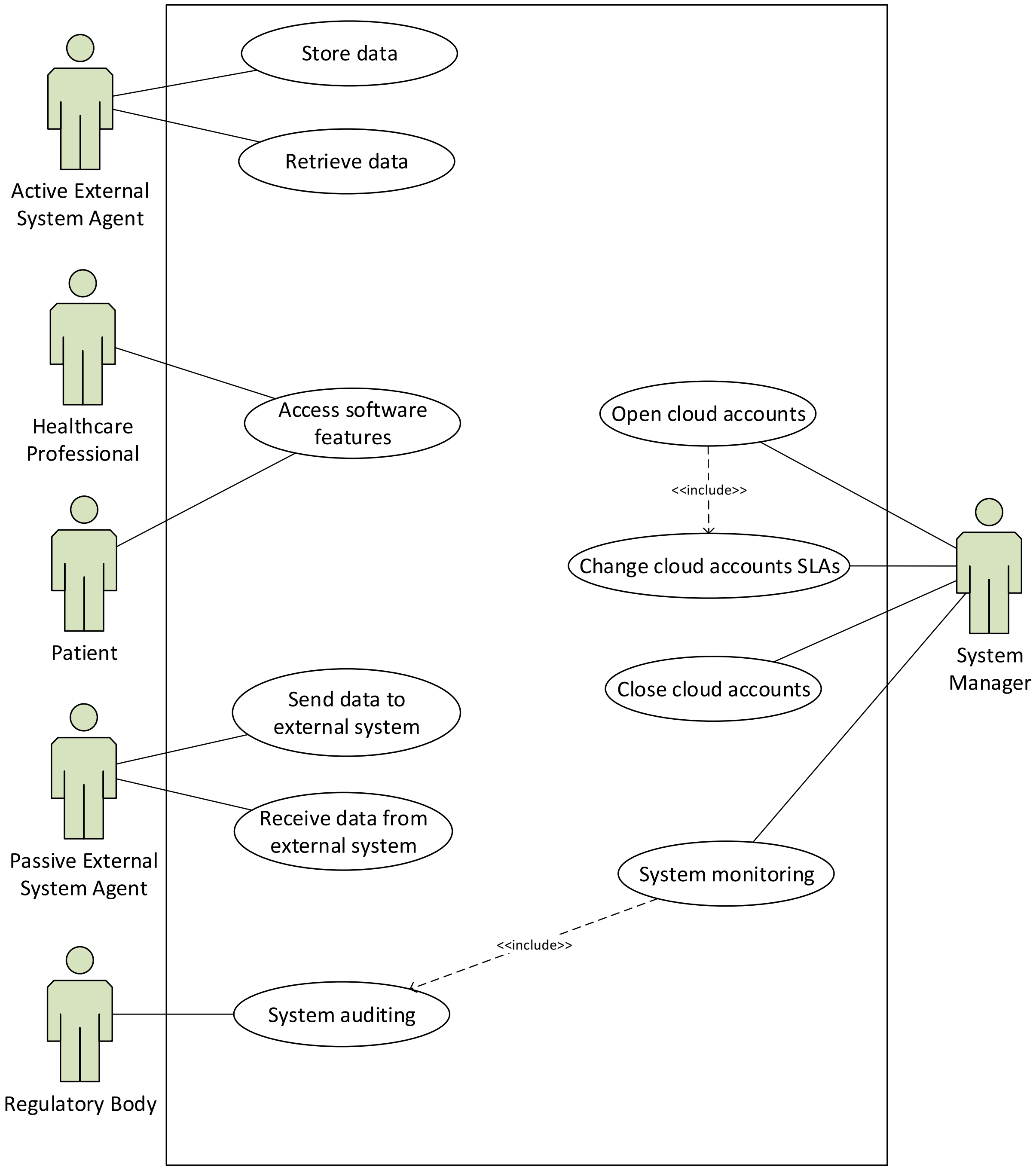


Figure 2: CLINIdATA® use cases diagram.

3.2.2 Actors

The actors involved in Maxdata use cases are:

- **Active External System Agent**: An active external system that communicates with CLINIdATA®. It is designated as active because the communication is initiated by the external system.
- **Passive External System Agent**: A passive external system that communicates with CLINIdATA®. It is designated as passive because the communication is initiated by CLINIdATA®.

- **Healthcare Professional**: A person working in a healthcare organization.
- **Patient**: A person that performs one or more exams in a healthcare organization.
- **System Manager**: A person responsible for maintenance and configuration operations, including setting up the cloud accounts necessary for the deployment of CLINIdATA® on the cloud.
- **Regulatory Body:** A public or government agency verifying system compliance with regulations.

3.2.3 Use Cases Description

This section details out all the relevant CLINIdATA® use cases by providing the following information:

- **Title**: A result-oriented title that reflects the task a user wants to accomplish.
- **Description**: High-level description of actions and the outcome of the use case.
- **Actors**: A list of external entities interacting with the product to accomplish a task.
- **Preconditions**: A list of conditions that should be true, before the use case can be started.
- **Postconditions**: A list of conditions that should be true, after the Normal flow has been executed.
- **Normal flow**: A detailed description of steps the actor and product go through to accomplish the goal of the use case.
- **Alternative flows**: A description of the alternative/less common scenarios of the use case (next to the Normal flow).
- **Exceptions**: A description of any anticipated error condition that could occur during the execution of the use case.
- **Includes**: A list of all the use cases that are used by the described use case.

3.2.3.1 Use Case – Store data

Description:	External systems (e.g., analysers, electronic requisition systems) need to store data (e.g., test results, external test requisitions) that will then be processed by CLINIdATA®.
Actors:	Active External System Agent
Preconditions:	External system agent is authorized to communicate with CLINIdATA®.
Postconditions:	The requested data is stored in CLINIdATA®.
Normal flow:	Steps: 1. Active External System Agent authenticates itself in CLINIdATA®. 2. Active External System Agent sends the data that should be stored in CLINIdATA®. 3. CLINIdATA® checks if the Active External System Agent is authorized to store the given data. 4. Data sent by the Active External System Agent is stored in CLINIdATA®.
Alternative flows:	Alternative flow 1: • In step 1. Active External System Agent is not able to authenticate due to lack of permission or wrong credentials. • Active External System Agent reports an error and stops use case execution. Alternative flow 2: • In step 3. CLINIdATA® verifies that the Active External System Agent is not authorized to store the data that was sent. • CLINIdATA® reports an error and stops use case execution. Alternative flow 3: • In step 4. CLINIdATA® cannot store the data that was sent due to an error. • CLINIdATA® reports an error and stops use case execution.
Exceptions:	• Network fails repeatedly during transfer → retry the transfer • Data loss during transfer → retry the transfer • Data errors during transfer → retry the transfer • Not enough space on the cloud storage → ask for more space
Includes:	-

3.2.3.2 Use Case – Retrieve data

Description:	External systems (e.g., analysers, ICU systems) need to retrieve data (e.g., patient information, test results) produced by CLINIdATA®.
Actors:	Active External System Agent
Preconditions:	External system agent is authorized to communicate with CLINIdATA®.
Postconditions:	The requested data is retrieved from CLINIdATA®.
Normal flow:	Steps: 1. Active External System Agent authenticates itself in CLINIdATA®. 2. Active External System Agent specifies the data that it wants to retrieve. 3. CLINIdATA® checks if the Active External System Agent is authorized to retrieve the specified data. 4. Active External System Agent retrieves data from CLINIdATA®.
Alternative flows:	Alternative flow 1: • In step 1. Active External System Agent is not able to authenticate due to lack of permission or wrong credentials. • Active External System Agent reports an error and stops use case execution. Alternative flow 2: • In step 3. CLINIdATA® verifies that the Active External System Agent is not authorized to retrieve the requested data. • CLINIdATA® reports an error and stops use case execution. Alternative flow 3: • In step 4. Active External System Agent cannot retrieve data from CLINIdATA® due to an error or data is not available. • Active External System Agent reports an error and stops use case execution.
Exceptions:	• Network fails repeatedly during transfer → retry the transfer • Data loss during transfer → retry the transfer • Data errors during transfer → retry the transfer
Includes:	-

3.2.3.3 Use Case – Access software features

Description:	Healthcare professionals and patients need to access CLINIdATA® features.
Actors:	Healthcare Professional, Patient
Preconditions:	Healthcare professional/Patient is authorized to use CLINIdATA®.
Postconditions:	Healthcare professional/Patient accesses CLINIdATA® features.
Normal flow:	Steps: 1. Healthcare professional/patient authenticates itself in CLINIdATA®. 2. Healthcare professional/patient accesses CLINIdATA® features.
Alternative flows:	Alternative flow 1: • In step 1. Healthcare professional/Patient is not able to authenticate due to lack of permission or wrong credentials. • Healthcare professional/Patient reports an error and stops use case execution. Alternative flow 2: • In step 2. Healthcare professional/Patient cannot access CLINIdATA® features due to an error. • Healthcare professional/Patient reports an error and stops use case execution.
Exceptions:	• Errors during access to CLINIdATA® features → retry the access to CLINIdATA® features
Includes:	-

3.2.3.4 Use Case – Send data to external system

Description:	CLINIdATA® needs to send data (e.g., test results) to external systems (e.g., hospital information system).
Actors:	Passive External System Agent
Preconditions:	Passive External System Agent is authorized to receive data from CLINIdATA®.
Postconditions:	The requested data is sent from CLINIdATA® to the Passive External System Agent.
Normal flow:	Steps: 1. CLINIdATA® checks if Passive External System Agent is authorized to receive the data ready to be sent. 2. CLINIdATA® sends data to the Passive External System Agent.
Alternative flows:	Alternative flow 1: • In step 1. Passive External System Agent is not authorized to receive the data to be sent due to lack of permission. • CLINIdATA® reports an error and stops use case execution. Alternative flow 2: • In step 2. CLINIdATA® cannot send data due to an error or data is not available. • CLINIdATA® reports an error and stops use case execution.
Exceptions:	• Network fails repeatedly during transfer → retry the transfer • Data loss during transfer → retry the transfer • Data errors during transfer → retry the transfer
Includes:	-

3.2.3.5 Use Case – Receive data from external system

Description:	CLINIdATA® needs to receive data (e.g., test requisitions, patient information) from external systems (e.g., hospital information system).
Actors:	Passive External System Agent
Preconditions:	CLINIdATA® is authorized to receive data from the Passive External System Agent.
Postconditions:	The received data is stored in CLINIdATA®.
Normal flow:	Steps: 1. CLINIdATA® checks if it is authorized to receive data from the Passive External System Agent. 2. CLINIdATA® receives data from the Passive External System Agent.
Alternative flows:	Alternative flow 1: • In step 1. CLINIdATA® is not authorized to receive data from the Passive External System Agent due to lack of permission. • CLINIdATA® reports an error and stops use case execution. Alternative flow 2: • In step 2. CLINIdATA® cannot receive data due to an error. • CLINIdATA® reports an error and stops use case execution.
Exceptions:	• Network fails repeatedly during transfer → retry the transfer • Data loss during transfer → retry the transfer • Data errors during transfer → retry the transfer • Not enough space on the cloud storage → ask for more space
Includes:	-

3.2.3.6 Use Case – System auditing

Description:	The healthcare organization stores patient data in CLINIdATA® deployed on the cloud. However, to comply with the regulation it is necessary to audit (log) all the sensitive operations performed in CLINIdATA®. The audit logs are secured to ensure their integrity.
Actors:	Regulatory Body
Preconditions:	CLINIdATA® is able to produce audit logs capturing sensitive operation of healthcare professionals accessing the patient data. These logs include who, when and what data was accessed/edited in CLINIdATA®.
Postconditions:	The regulatory body accesses CLINIdATA® audit logs.
Normal flow:	Steps: 1. Regulatory body logs-in the CLINIdATA® audit system. Regulatory body authenticates itself with multi-factor authentication. 2. Regulatory body uses the CLINIdATA® audit system to view and generate the audit reports. 3. Regulatory body logs-out from the CLINIdATA® audit system. 4. Regulatory body analyses the reports and informs the healthcare organization in case of regulatory problems.
Alternative flows:	Alternative flow 1: • In step 2. the CLINIdATA® audit system fails to generate the audit reports. • Regulatory body reports the issue to the healthcare organization.
Exceptions:	• The CLINIdATA® audit system crashes → retry report generation
Includes:	-

3.2.3.7 Use Case – Open cloud accounts

Description:	System manager opens a set of cloud accounts so that a healthcare organization can access CLINIdATA® deployed on a set of cloud providers by making use of SafeCloud components (see Section 3.3). Initial SLAs are configured according to the requirements of the healthcare organization.
Actors:	System Manager
Preconditions:	System manager can access each cloud provider management system.
Postconditions:	Cloud accounts ready so that a healthcare organization can access CLINIdATA® deployed on a set of cloud providers by making use of SafeCloud components.
Normal flow:	Steps: 1. System manager logs-in each cloud provider management system (e.g. Amazon, Google). 2. System manager enters all the required information. 3. The cloud provider management system creates the account with the requested data. 4. System manager deploys CLINIdATA®. 5. System manager logs-out from each cloud provider management system.
Alternative flows:	Alternative flow 1: • In step 3. the cloud provider management system fails to create the cloud account. • System manager reports the issue to the cloud provider.
Exceptions:	• The cloud provider management system crashes → restart from step 1 for the cloud provider that crashed
Includes:	See Section 3.2.3.8: "Use case - Change cloud accounts SLAs"

3.2.3.8 Use Case – Change cloud accounts SLAs

Description:	System manager needs to set the SLAs associated to the cloud accounts, when the accounts are created and when there are changes in the requirements of the healthcare organization.
Actors:	System Manager
Preconditions:	System manager can access each cloud provider management system.
Postconditions:	SLAs associated to the cloud accounts are changed.
Normal flow:	Steps: 1. System manager logs-in each cloud provider management system (e.g., Amazon, Google). 2. System manager enters the new SLAs. 3. The cloud provider management system applies the new SLAs. 4. System manager redeploys CLINIdATA® if needed. 5. System manager logs-out from each cloud provider management system.
Alternative flows:	Alternative flow 1: • In step 3. the cloud provider management system fails to set the new SLAs. • System manager reports the issue to the cloud provider.
Exceptions:	• The cloud provider management system crashes → restart from step 1 for the cloud provider that crashed
Includes:	-

3.2.3.9 Use Case – Close cloud accounts

Description:	System manager closes existing cloud accounts used by a healthcare organization when they are no longer required.
Actors:	System Manager
Preconditions:	System manager can access each cloud provider management system.
Postconditions:	Cloud accounts are deleted.
Normal flow:	Steps: 1. System manager logs-in each cloud provider management system (e.g., Amazon, Google). 2. System manager enters the required information. 3. The cloud provider management system closes the account. 4. System manager undeploys CLINIdATA®. 5. System manager logs-out from each cloud provider management system.
Alternative flows:	Alternative flow 1: • In step 3. the cloud provider management system fails to close the cloud account. • System manager reports the issue to the cloud provider.
Exceptions:	• The cloud provider management system crashes → restart from step 1 for the cloud provider that crashed
Includes:	-

3.2.3.10 Use Case – System monitoring

Description:	System manager needs the ability to monitor resources used by each cloud account in order to prove compliance with the SLA agreements. This includes CLINIdATA® audit logs.
Actors:	System Manager
Preconditions:	Each cloud provider has infrastructure able to capture SLA-related core parameters of the running cloud accounts.
Postconditions:	A monitoring dashboard and report showing the current status of each cloud account.
Normal flow:	Steps: 1. System manager logs-in each cloud provider management system (e.g., Amazon, Google). 2. System manager uses the cloud provider management system to access a dashboard showing the current status of each cloud account. From the dashboard, it is possible to generate a report of system health per cloud account. 3. System manager logs-out from each cloud provider management system. 4. System manager uses the report for analysis of SLA compliance (i.e., failure to meet some SLA aspects by the cloud provider).
Alternative flows:	Alternative flow 1: • In step 2. the cloud provider management system fails to show dashboard or to generate report. • System manager reports the issue to the cloud provider.
Exceptions:	• The cloud provider management system crashes → restart from step 1 for the cloud provider that crashed
Includes:	See Section 3.2.3.6: "Use case - System auditing"

3.3 Integration of CLINIdATA® with the SafeCloud framework

3.3.1 Overview

Taking advantage of the benefits of the SafeCloud framework, CLINIdATA® may be deployed on the cloud in 3 different scenarios depending on the customer type and on the features that the customer wants to access:

- **SaaS deployment**: for small and medium-scale healthcare organizations that want to reduce costs on infrastructure, CLINIdATA® will be offered using the software-as-a-service (SaaS) model where all components are deployed on cloud providers contracted by Maxdata.
- **Hybrid deployment**: for large healthcare organizations that want to reduce costs on infrastructure but do not trust any cloud provider, CLINIdATA® computation/processing will be installed on customer's premises making use of SafeCloud components to access data securely stored on untrusted cloud providers contracted by healthcare organizations.
- **Analytics deployment**: for groups of healthcare organizations that want to share analytics on their combined data without revealing the private data of each organization, CLINIdATA® computation/processing will be installed on each of the involved customer's premises making use of SafeCloud components to access data securely stored on untrusted cloud providers – contracted by the healthcare organizations - in a way that each healthcare organization is only allowed to put in its private data but cannot make direct queries for data – i.e. only aggregated queries are possible.

The remaining sections describe each of these cloud deployment scenarios in more detail, including the way CLINIdATA® will take advantage of the SafeCloud framework to achieve its goals.

3.3.2 SaaS Deployment

For small and medium healthcare organizations that want to reduce costs on infrastructure, CLINIdATA® will be offered using the model software-as-a-service (SaaS) where all components are deployed on cloud providers contracted by Maxdata. Healthcare organizations subscribe access to the software and pay per use. Different kinds of cloud providers are used for the different layers of CLINIdATA®:

- CLINIdATA® stateless application server – i.e., CLINIdATA® processing component - is deployed on a trusted private cloud owned by a cloud provider, which is trusted for processing but not for long-term storage. Different replicas of the CLINIdATA® application server may be deployed for load-balancing and crash fault-tolerance.
- CLINIdATA® data is stored in a set of untrusted public cloud providers. Availability and integrity are guaranteed by the mechanisms already existing in most of the public cloud providers (e.g., Amazon, Google). Confidentiality of private data, including personal data, is guaranteed by the SafeCloud Secure SQL Engine (cf. Section 2.4) and the SafeCloud Secure File System (cf. Section 2.3).
- Secure channels are used for the communication between the trusted and the untrusted clouds.

Figure 3 depicts the way CLINIdATA® is integrated with the SafeCloud framework in a SaaS deployment. Before the initial deployment, the healthcare organization defines the location (e.g., country, state) where data may be processed and stored. Cloud providers are selected accordingly.

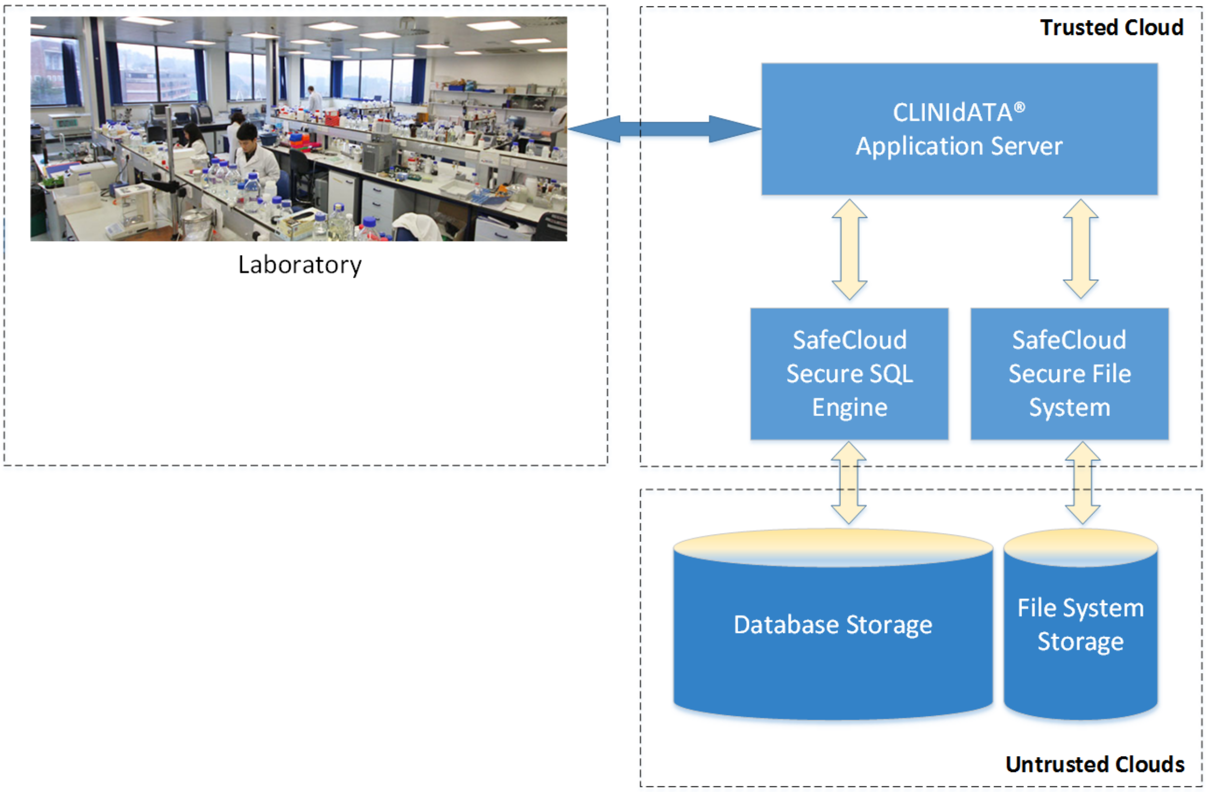


Figure 3: SaaS deployment: integration between CLINIdATA® and the SafeCloud framework.

In this scenario, all the use cases presented in Section 3.2 apply.

3.3.3 Hybrid Deployment

For large healthcare organizations that want to reduce costs on infrastructure but do not trust on any kind of cloud provider, CLINIdATA® computation/processing will be installed on premises making use of SafeCloud components to access data securely stored on untrusted cloud providers, which are less expensive and more reliable than local storage. In summary:

- CLINIdATA® stateless application server – i.e., CLINIdATA® processing component - is deployed on premises. Different replicas of the CLINIdATA® application server may be deployed for load-balancing and crash fault-tolerance.
- CLINIdATA® data is stored in a set of untrusted public cloud providers. Availability and integrity are guaranteed by the mechanisms already existing in most of the public cloud providers (e.g., Amazon, Google). Confidentiality of private data, including personal data, is guaranteed by the SafeCloud Secure SQL Engine (cf. Section 2.4) and the SafeCloud Secure File System (cf. Section 2.3). Untrusted cloud providers are able to store the data but also to do some processing on top of it without disclosing any sensitive information. This allows reducing the amount of computation done on organizations premises.
- Secure channels are used for the communication between the laboratory premises and the untrusted clouds.

Figure 4 depicts the way CLINIdATA® is integrated with the SafeCloud framework in a hybrid deployment. Before the initial deployment, the healthcare organization defines the location (e.g., country, state) where data may be stored. Cloud providers are selected accordingly.

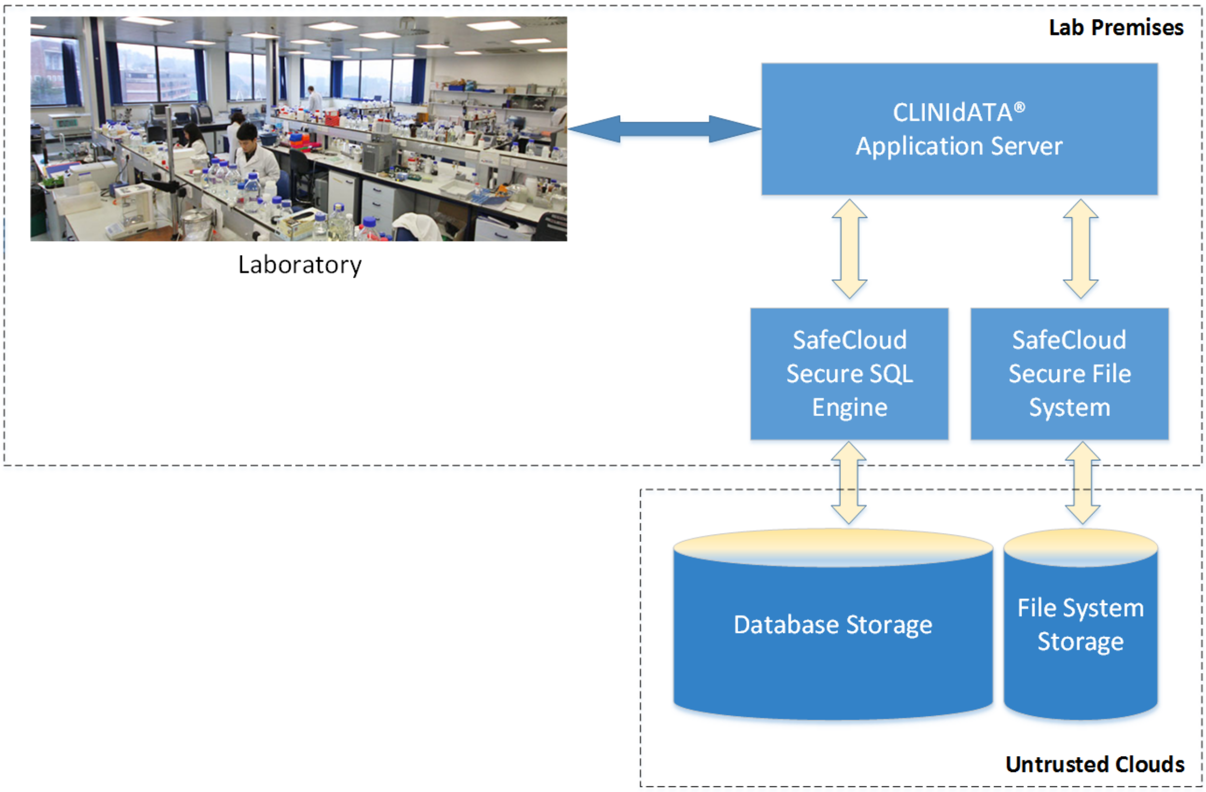


Figure 4: Hybrid deployment: integration between CLINIdATA® and the SafeCloud framework.

In this scenario, all the use cases presented in Section 3.2 apply.

3.3.4 Analytics Deployment

For groups of healthcare organizations that want to share analytics on their combined data without revealing the private data of each organization, CLINIdATA® computation/processing will be installed on premises making use of SafeCloud components to access data securely stored on untrusted cloud providers in a way that each healthcare organization is only allowed to put in its private data but cannot make direct queries for data – only aggregated queries are possible. In summary:

- CLINIdATA® stateless application server – i.e., CLINIdATA® processing component - is deployed on premises.
- CLINIdATA® data is stored in the following way:
    - The private data of each organization is stored on its premises. Each organization Secure SQL Engine participates in a secure multi-party communication protocol to produce the results of aggregated queries.
    - To tolerate cases where some of the organizations Secure SQL Engines are offline, private data is secret-shared and shares are stored in a set of untrusted public cloud providers. Availability and integrity are guaranteed by the mechanisms already existing in most of the public cloud providers (e.g., Amazon, Google). Confidentiality of private data,

including personal data, is guaranteed by the SafeCloud Secure SQL Engine (cf. Section 2.4).

    - In this scenario, the SafeCloud Secure SQL Engine is also responsible for ensuring that each healthcare organization can only make aggregated queries for data.

- Secure channels are used for the communication between the hospital premises and the untrusted clouds.

Figure 5 depicts an example of how CLINIdATA® is integrated with the SafeCloud framework in an analytics deployment. In this example, two Portuguese hospitals that share analytics on epidemiological surveillance using SafeCloud components to protect their own private data, while sharing aggregated information. Before the initial deployment, the healthcare organizations define the location (e.g., country, state) where data may be stored. Cloud providers are selected accordingly.

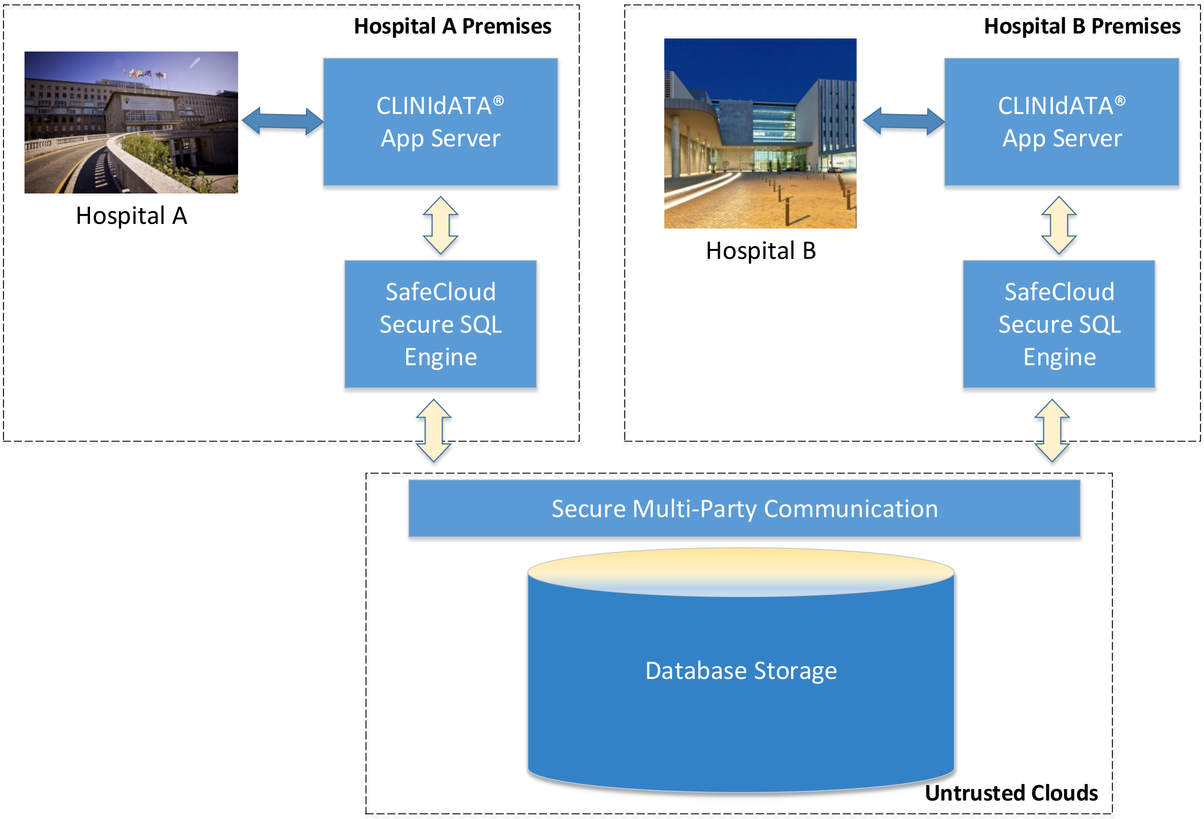


Figure 5: Analytics deployment: integration between CLINIdATA® and the SafeCloud framework.

In this scenario, all the use cases presented in Section 3.2 apply. However, the use cases that involve data access – store data, retrieve data, access software features, send data to external system, receive data from external system – will have limited functionality in practice given that only aggregated queries are possible.

4 Requirements

This chapter presents the relevant requirements to be addressed by the SafeCloud framework from the perspective of a healthcare software vendor that wants to deploy its software on the cloud ensuring the confidentiality, integrity and availability of personal data. These requirements are derived from the use cases and scenarios presented in Section 3. Requirements marked as mandatory should be satisfied by the end of the SafeCloud project.

4.1 Functional requirements

Data-Location-Definition (Mandatory): SafeCloud framework shall give the possibility to constrain the location of the resources to country boundaries if the underlying cloud provider(s) support this feature.

Time-Bounds-Definition: SafeCloud framework shall give the possibility to specify time bounds regarding the response times of the computation / storage / network resources.

4.2 Non-functional requirements

4.2.1 Security

- **GDPR-Compliance (Mandatory)**: SafeCloud framework shall be compliant with the EU General Data Protection Regulation [GDPR16], namely when the SafeCloud framework is used to process/store/communicate personal data.
- **Secure-Resources-Access (Mandatory)**:
    - SafeCloud framework shall give access to its resources only for the authorized user.
    - SafeCloud framework shall allow definition of access to the system from defined locations, e.g., only from within the healthcare organization.
- **Secure-Data-Handling (Mandatory)**:
    - SafeCloud framework shall guarantee that data cannot be interpretable in rest.
    - SafeCloud framework shall guarantee that data cannot be interpretable in transit.
    - SafeCloud framework shall guarantee data integrity and tamper resistance.
    - SafeCloud framework shall ensure that all privacy-sensitive information (e.g., patient name, date of birth) is not exposed (e.g., via logging).

4.2.2 Reliability

- **Data-Location-Guarantees (Mandatory)**:
    - SafeCloud framework shall guarantee to store data in the prescribed legal boundaries if the underlying cloud provider(s) support this feature.
    - SafeCloud framework shall guarantee to process data in the prescribed legal boundaries if the underlying cloud provider(s) support this feature.
- **Availability-Storage-Guarantees**: SafeCloud framework shall guarantee storage availability up to 99.999%.

- **Availability-Network-Guarantees**: SafeCloud framework shall guarantee network availability up to 99.999%.
- **Availability-Computation-Guarantees**: SafeCloud framework shall guarantee computation availability up to 99.999%.

4.2.3 Real-Time

- **Time-Bounds-Guarantees**: SafeCloud framework shall guarantee predictable and bounded computation, communication and storage access times.

4.3 Interface requirements

- **File-Storage**: SafeCloud framework shall provide storage based on files.
- **SQL-Database-Storage**: SafeCloud framework shall provide storage based on a relational SQL database.

5 Conclusion

This document presents the design and requirements of the healthcare platform that will be built by Maxdata on top of the SafeCloud framework.

The deliverable presents:

- A summary of the SafeCloud framework.
- The **design** of the healthcare platform, including its main features, use cases, and how it will be integrated with the SafeCloud framework.
- The relevant **requirements** to be addressed by the SafeCloud framework taking into consideration the healthcare platform that will be built.

The next step will be to implement and demonstrate the described healthcare platform on top of the SafeCloud framework. A first prototype will be described in deliverable D5.4 (month 24) and the final version will be described in D5.6 (month 36).

6 References

[GDPR16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) http://data.europa.eu/eli/reg/2016/679/oj