SafeCloud

# Final Deployment, Maxdata SafeCloud-based Healthcare Platform

## D5.6

Project reference no. 653884

August 2018

European Commission

Horizon 2020
European Union funding
for Research & Innovation

## Document information

| | |
|---|---|
| Scheduled delivery | 31.08.2018 |
| Actual delivery | 31.08.2018 |
| Version | 1.0 |
| Responsible Partner | Maxdata Software, S.A. |

## Dissemination level

Public

## Revision history

| Date | Editor | Status | Version | Changes |
|---|---|---|---|---|
| 16.08.2018 | N. Lages | Draft | 0.1 | Initial version. |
| 17.08.2018 | P. Sousa | Draft | 0.2 | Global review; GDPR chapter draft |
| 17.08.2018 | N. Lages | Draft | 0.3 | Global review. |
| 18.08.2018 | P. Sousa | Draft | 0.4 | GDPR chapter final |
| 20.08.2018 | N. Lages | Draft | 0.5 | Global review. |
| 21.08.2018 | J. Paulo | Draft | 0.6 | INESC TEC review. |
| 22.08.2018 | N. Lages | Draft | 0.7 | Global review. |
| 22.08.2018 | P. Sousa | Draft | 0.8 | Global review. |
| 24.08.2018 | V. Sokk | Draft | 0.9 | Cybernetica review. |
| 24.08.2018 | N. Lages | Final | 1.0 | Final release. |

## Contributors

Nuno Lages (Maxdata)
Paulo Sousa (Maxdata)

## Internal reviewers

João Paulo (INESC TEC)
Ville Sokk (Cybernetica)

## 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 |
|---|---|
| BI | Business Intelligence |
| BLOB | Binary Large Object |
| CPU | Central Processing Unit |
| DBMS | Database Management System |
| ETL | Extract Transform Load |
| EU | European Union |
| GCP | Google Cloud Platform |
| GDPR | General Data Protection Regulation |
| GUI | Graphical User Interface |
| GWT | Google Web Toolkit |
| HIPAA | Health Insurance Portability and Accountability Act |
| OLAP | Online Analytical Processing |
| ORM | Object-relational Mapping |
| SaaS | Software as a Service |
| SQ1 | SafeCloud Secure Queries solution 1 |
| SQ2 | SafeCloud Secure Queries solution 2 |
| SQ3 | SafeCloud Secure Queries solution 3 |
| SVG | Scalable Vector Graphics |
| WP5 | Work Package 5 |

## Table of contents

## Executive summary

This deliverable describes the final deployment of the Maxdata SafeCloud-based Healthcare Platform. The topics presented are:

- Overview of the CLINIdATA® eHealth solution, Maxdata's most important product, including its legal constraints and planned deployment strategies.
- Description of the final deployment of SafeCloud Healthcare Platform, covering the 3 scenarios presented in D5.2 – software as a service (SaaS), hybrid and analytics deployment) and including the challenges addressed during the integration of CLINIdATA® and Secure Queries solutions.
- Description of the testing of the integration of CLINIdATA® and Secure Queries Solutions whose results we describe in D4.5.
- Demonstrations of the use case described in Deliverable D5.2.
- Considerations on how potential customers (healthcare providers) should proceed when implementing the SafeCloud Healthcare Platform, to ensure General Data Protection Regulation (GDPR) compliance.

# 1 Introduction

The objective of work package 5 (WP5) was to deploy SafeCloud technologies in order to provide its novel privacy-preserving features to new and improved products and services. We demonstrate two use cases: a cloud storage platform and a healthcare platform.

In deliverable D5.2 we presented the design and requirements of the healthcare platform built by Maxdata on top of the SafeCloud framework.

In D5.4 we presented the first prototype of SafeCloud Healthcare Platform. This deliverable included a description of the design of the platform and the early challenges addressed during the integration of CLINIdATA® with SafeCloud Secure Queries solutions. It also included the first demonstrators of the integrated system.

This deliverable, D5.6, describes the final deployment of the SafeCloud-based healthcare platform. The structure of this document is as follows:

- Section 2 presents the design of the healthcare platform. This platform is based on CLINIdATA®, an eHealth web application owned by Maxdata.
- Section 3 describes the components of the platform and the challenges faced during their integration.
- Section 4 describes the demonstrator, which makes use of the SafeCloud Secure Queries layer.
- Section 5 presents considerations on how potential customers (healthcare providers) should proceed when implementing the SafeCloud-based healthcare platform, to ensure GDPR compliance.
- Section 6 provides a summary of the conclusions about the platform.

# 2 CLINIdATA® eHealth Solution

As described in deliverable D5.2, CLINIdATA®, an eHealth web application owned by Maxdata, is the base for the SafeCloud healthcare platform. CLINIdATA® is composed of several modules, including a laboratory information system and an epidemiological surveillance system dedicated to supporting hospital infection control committees.

## 2.1 Product context

CLINIdATA® supports several types of clinical and non-clinical organizations. Within the clinical domain, CLINIdATA® serves various healthcare organizations, including hospitals, clinics, laboratories and primary care units. In the non-clinical domain, CLINIdATA® serves laboratories in many areas, including water, food, and environmental analysis.

Maxdata's strategy for the deployment of CLINIdATA® has been to install it on client premises. This approach has allowed the company to conquer most of the Portuguese market – currently Maxdata is present in more than 80% of national public hospitals. Now Maxdata wants to sell CLINIdATA® abroad. One of the strategies to provide CLINIdATA® to international clients is to deploy it to the cloud and sell it as 'software as a service' (SaaS).

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 Edge). HTML and JavaScript code are generated from Google Web Toolkit (GWT)[1] Java code;
- **Business logic layer**: Set of services running on the server that implement business logic. These services are implemented in Java using Spring Framework[2];
- **Database access layer**: Set of methods running on the server used to access the relational database where data is persisted. These methods use the Hibernate[3] object-relational mapping framework to abstract the concrete database management system (DBMS) (e.g., Oracle[4], PostgreSQL[5]).

A variety of organizations are using CLINIdATA®, ranging from small laboratories with a few dozen professionals and hundreds of transactions per day to very large hospital clusters with thousands of professionals and tens of millions of transactions per day. Deliverable D5.2 details the typical workloads of large versus small organizations in terms of users, analysis instruments, clinical tests, storage, and database transactions.

## 2.2 Product capabilities/functions

CLINIdATA® is an eHealth application with the following characteristics:

- 100% web application - no plugins required on the client side, only a web browser is needed to run the application;

[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

- 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);
- Supports overall management and control in 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);
- Supports overall management and control in different types of non-clinical laboratories, e.g., water, air, food;
- Supports management of the entire workflow of analysis processing in clinical and non-clinical laboratories, including the 3 typical phases:
    - **Pre-analytical**: Prescription, specimen collection, specimen transport;
    - **Analytical**: Analysis execution, quality control, result validation or approval;
    - **Post-analytical**: Access to results by patients/customers or clinical/laboratory staff, billing.
- Epidemiological surveillance tool dedicated to supporting hospital infection control committees in preventing, identifying and monitoring infections;
- Provides a real-time interface that can present results collected from over 300 different automated analyzers;
- Integrates readily with dozens of other clinical and non-clinical information systems (e.g., intensive care unit, patient identification, billing, regional health portals);
- Rules engine that allows incorporation of intelligence and safety throughout the application;
- Business Intelligence (BI) platform for viewing and exploring statistics through dashboards and other visual elements.

## 2.3 Constraints

When deployed in healthcare organizations, CLINIdATA® manages personal data, so it must be compliant with regulations on personal data protection (e.g., in Europe, GDPR[6], in the USA, Health Insurance Portability and Accountability Act – HIPAA[7]).

## 2.4 Cloud Deployment Scenarios

CLINIdATA® may take advantage of the benefits of the SafeCloud framework by being deployed to the cloud in 3 different scenarios, depending on the type of customer and on the features the customer wants to use:

- **SaaS deployment**: for small and medium-sized healthcare organizations that want to reduce costs of infrastructure, Maxdata will offer CLINIdATA® through the SaaS model, deploying all components to cloud providers contracted by Maxdata (Figure 1).

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

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

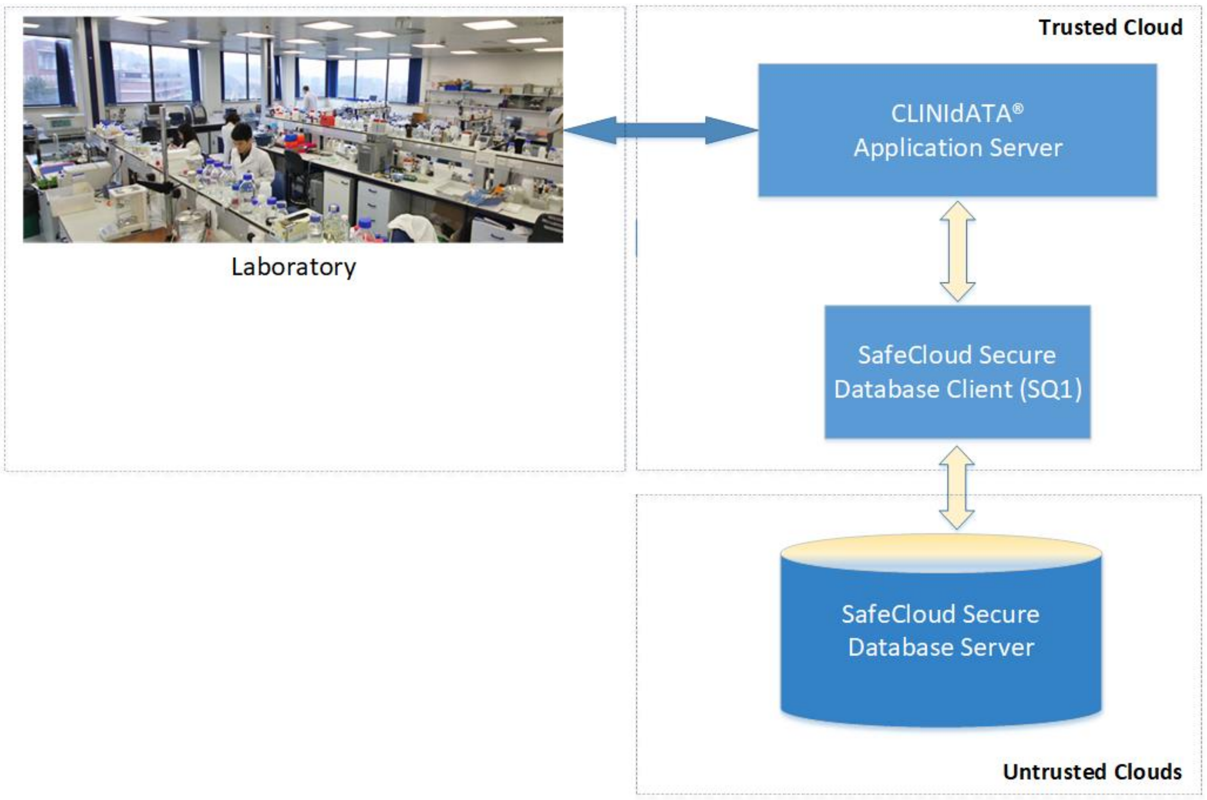


**Figure 1 - SafeCloud Healthcare platform: SaaS deployment**

- **Hybrid deployment**: for large healthcare organizations that want to reduce costs of infrastructure but do not trust any cloud provider, Maxdata will install CLINIdATA® on customer premises, making use of SafeCloud components to access and store data securely on untrusted cloud providers contracted by healthcare organizations (Figure 2).

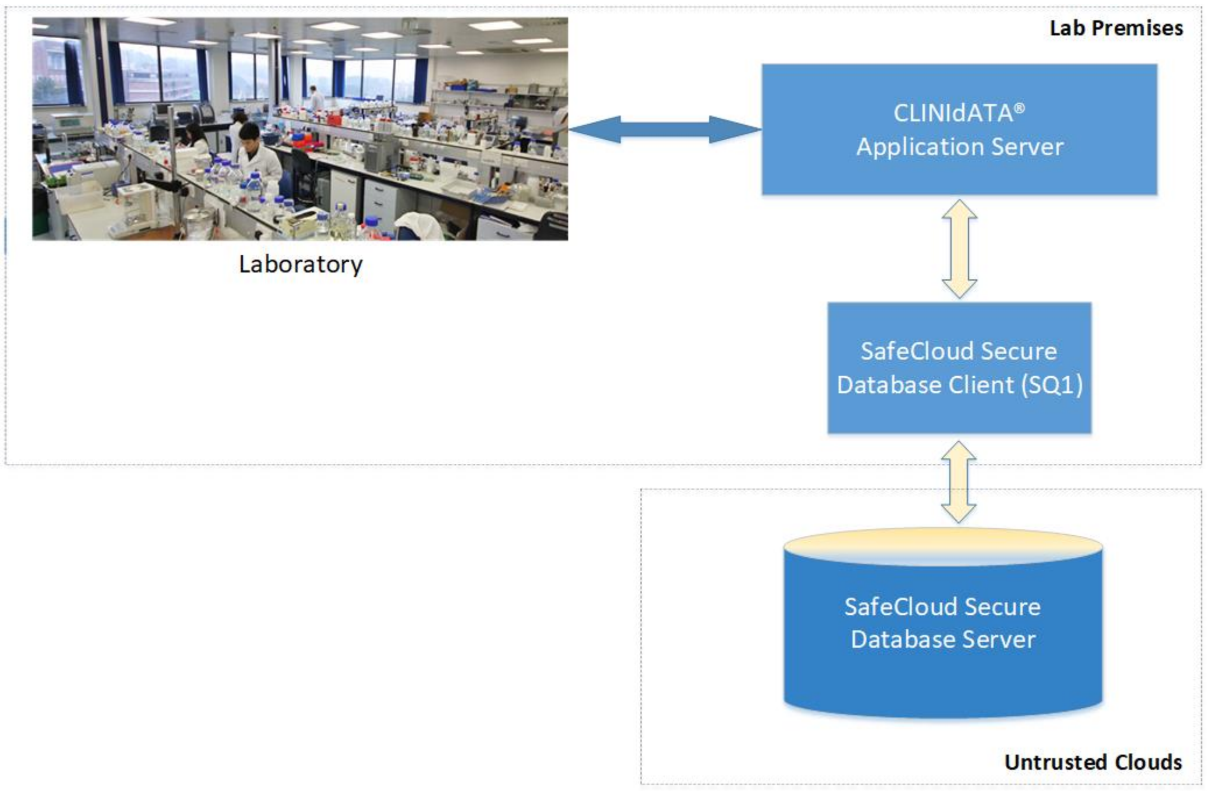


**Figure 2 – SafeCloud Healthcare platform: hybrid deployment**

- **Analytics deployment**: for groups of healthcare organizations that want to share data for the purpose of analytics without compromising data privacy, Maxdata will install CLINIdATA® on each of these organizations, making use of SafeCloud components to access data stored securely on untrusted cloud providers contracted by the organizations. Each healthcare organization will have a tool to import their data to a distributed SQ3 installation. The data of each organization stays at the organization's chosen cloud or server and is not visible to other organizations, because they cannot make direct queries to the data through SQ3. Organizations can only retrieve summarized data through aggregate queries. In SafeCloud Healthcare Platform, the distributed SQ3 installation receives queries from a third-party open-source BI solution, Knowage[8], with extensive statistical and data visualization capabilities (Figure 3)

[8] https://www.knowage-suite.com

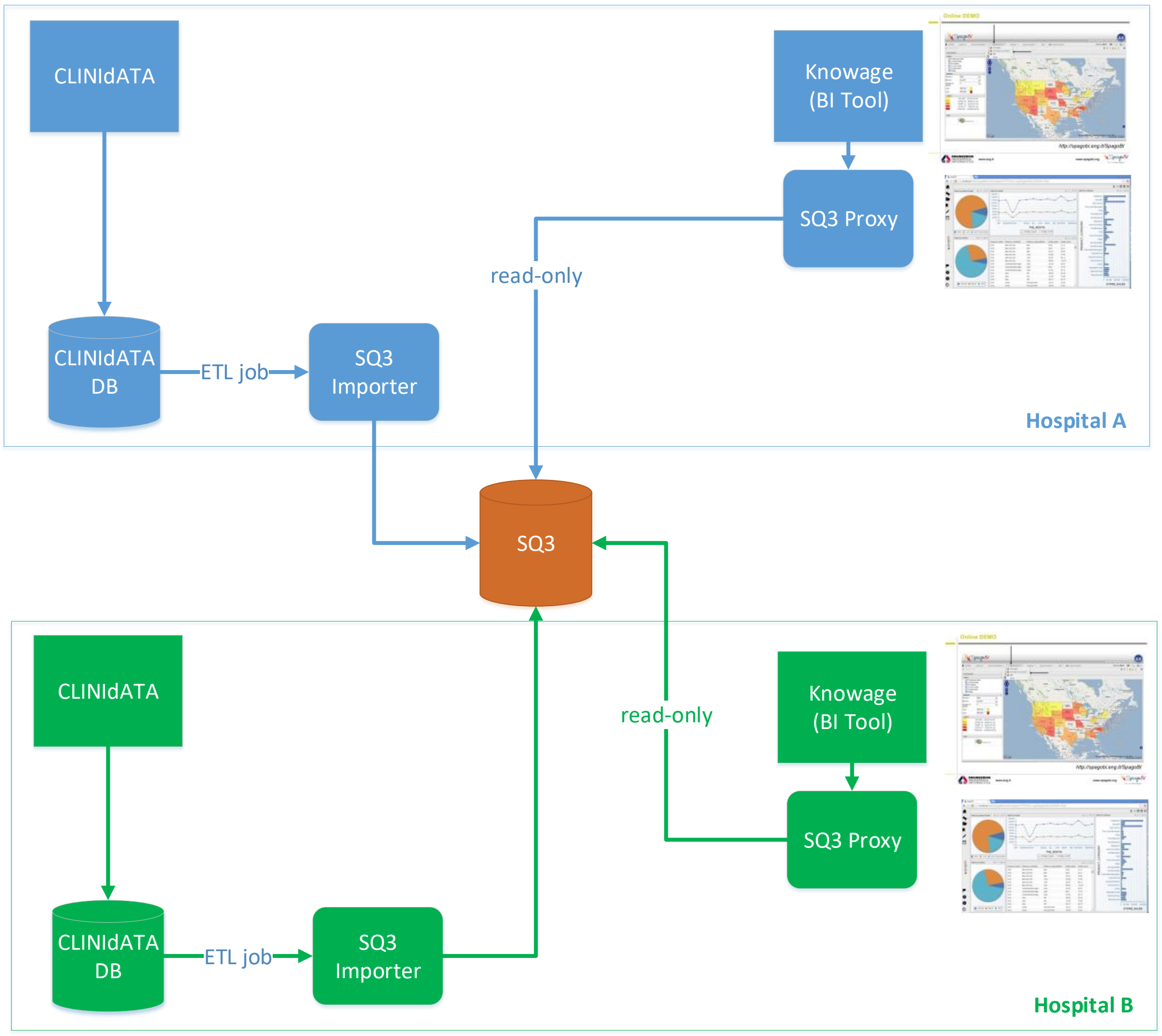


**Figure 3 – SafeCloud Healthcare platform: analytics deployment**

Deliverable D5.2 described each of these cloud deployment scenarios in more detail, including the way CLINIdATA® will take advantage of the SafeCloud framework to meet its requirements.

# 3 SafeCloud Healthcare Platform

## 3.1 Introduction

The healthcare platform covers the SaaS and hybrid deployment scenarios as well as the analytical solution. The three solutions are described in D5.2 and summarized in Section 2 of the present document.

Integrating CLINIdATA® and SafeCloud Secure Queries solution 1 (SQ1) took most of the effort to achieve the goals for the healthcare platform. Integration with SafeCloud Secure Queries solution 2 (SQ2) largely benefited from the integration work on SQ1, as both solutions share common components. Integration with SafeCloud Secure SQL Engine solution 3 (SQ3) provides the capability to analyze data from CLINIdATA with a business intelligence tool. The following sections describe the details of the integration of CLINIdATA® with SQ1, SQ2 and SQ3, and the tests performed to validate this work.

## 3.2 Integration of CLINIdATA® with SQ1 and SQ2

### 3.2.1 CLINIdATA® Integration Work

Integrating CLINIdATA® with SQ1 required some changes for full support of SQ1's SQL interface based on Apache Derby[9]. None of Maxdata's customers uses Apache Derby, therefore, although CLINIdATA® aims to be compatible with the major relational DBMSs, CLINIdATA®'s support of Derby SQL was a challenge addressed as part of the CLINIdATA®/SQ1 integration effort. Thanks to the use of the Hibernate ect-relational Mapping (ORM) Framework, such changes were relatively straightforward. The use of Hibernate allowed reusing almost all SQL queries already implemented in CLINIdATA®. Nevertheless, adjustments were necessary to support validation of CLINIdATA® user licenses (Figure 5), subtraction of timestamps (Figure 6 and Figure 7) and handling of strings (Figure 7).

We needed to load the database with sufficient data and ensure that it would be in a state such that CLINIdATA® would work correctly. To load data into the SQ1 database, we used Liquibase[10], a cross-database migration tool that uses specifications in xml documents. Liquibase xml supports definition of schemas and data for relational databases. After some experimentation, the optimal strategy to setup a functional database in SQ1 was to create CLINIdATA®'s schema with Liquibase[11] and then populate its tables with a script of insert statements (Figure 4). We migrated CLINIdATA®'s database schema using the Liquibase (Figure 8) and populated SQ1 with data from an internal PostgreSQL testing database at Maxdata. We used pg_dump, a PostgreSQL command line management tool, to export this data to a script of insert statements. This script still needed some manual processing to remove or replace data in formats not compatible with Apache Derby before it was loaded into SQ1 with Apache Derby's command line management tool, ij. The icons of CLINIdATA®'s graphical user interface (GUI) were loaded separately using Liquibase (as specified in the document partially shown in Figure 10). Loading the icons, which are scalable vector graphics (SVG) files, involved extra development effort on SQ1, to support storage of binary large objects (BLOBs) in the database. Finally, we wrote separate Java

[9] https://db.apache.org/derby/
[10] http://www.liquibase.org/
[11] https://www.liquibase.org/

scripts to update database indexes (Figure 11) and primary key-generating sequences (Figure 12).

```sql
INSERT INTO clinidata_new.tbw_clin_terminology_item VALUES (1, 'M8811/3', 'FIBROMIXOSSARCOMA', NULL, NULL, true, '0001-09-06 00:00:00', 2);
INSERT INTO clinidata_new.tbw_clin_terminology_item VALUES (1, 'M9540/0', 'NEUROFIBROMA SOE', NULL, NULL, true, '0001-09-06 00:00:00', 2);
INSERT INTO clinidata_new.tbw_clin_terminology_item VALUES (1, 'M9540/1', 'NEUROFIBROMATOSE SOE', NULL, NULL, true, '0001-09-06 00:00:00', 2);
INSERT INTO clinidata_new.tbw clin_terminology_item VALUES (1, 'M8811/0', 'FIBROMIXOMA', NULL, NULL, true, '0001-09-06 00:00:00', 2);
```

**Figure 4 – Example SQL insert statements used to populate SafeCloud Secure Database Server**

```java
@Override
public String getClientInformation() {
	String clientInfo = null;

	log.debug("getClientInformation: getting licenses");
	try {
		StringBuilder sqlQuery = new StringBuilder();
		String dialectName = ServerMethods.getDBRuntimeDialect().getClass().getSimpleName();
		if (dialectName.contains("Oracle")) {
			sqlQuery.append(" select object_id as OID from user_objects where object_name=UPPER('" +
TbLicense.table_sqlName + "') ");
		} else if (dialectName.contains("PostgreSQL")) {
			sqlQuery.append(" SELECT c.oid FROM pg_class c, pg_namespace a WHERE c.relname=LOWER('" +
TbLicense.table_sqlName + "') AND c.relnamespace = a.oid AND a.nspname=current_user");
		} else if (dialectName.contains("Derby")) {
			sqlQuery.append(" select t.TABLEID from SYS.SYSTABLES t JOIN sys.sysschemas sc ON t.schemaid = sc.schemaid
WHERE t.TABLENAME=UPPER('" + TbLicense.table_sqlName + "') and sc.schemaname = current schema");
		} else if (dialectName.contains("SQLServer")) {
			sqlQuery.append(" SELECT object_id  FROM sys.objects s inner join sys.schemas sh on   s.schema_id =
sh.schema_id where LOWER(s.name)=LOWER('" + TbLicense.table_sqlName + "')   and type_desc = 'USER_TABLE'  and LOWER(sh.name) =
LOWER(SCHEMA_NAME())");
		}

		Query query = getCurrentSession().createSQLQuery(sqlQuery.toString());
		clientInfo = query.uniqueResult().toString();

		log.debug("getClientInformation: get licenses successful");
	} catch (RuntimeException re) {
		log.error("getClientInformation: error getting licenses", re);
		throw re;
	}
	return clientInfo;
}
```

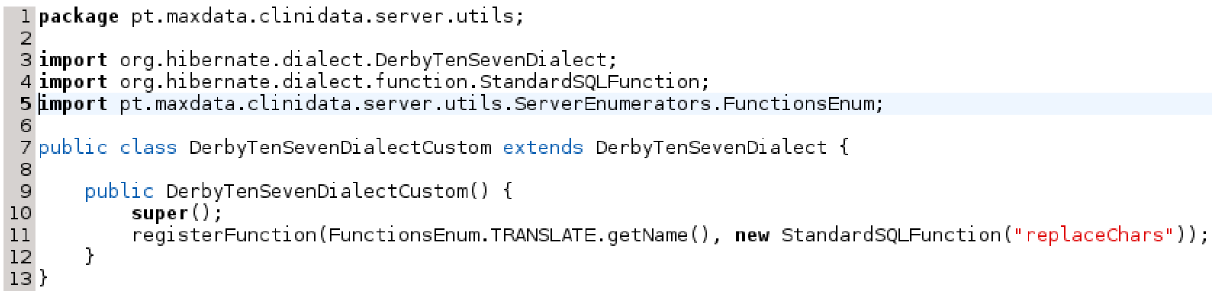
**Figure 5 – Java method that validates the client's license**

```java
package pt.uminho.haslab.SQlFunctions;

import java.sql.Timestamp;

public class TimestampsOperations {

    public static int subtractTimestampsInDays(Timestamp t1, Timestamp t2) {
        return (int) ((t2.getTime() - t1.getTime()) / (24 * 60 * 60 * 1000));
    }
}
```

**Figure 6 – Java class with a method that subtracts timestamps and returns the results in days**

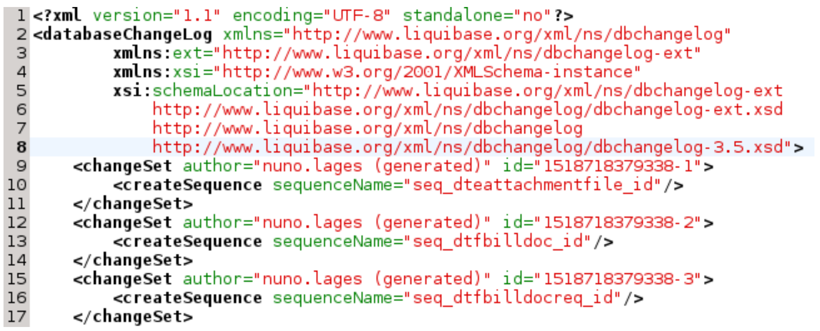
```sql
CALL SQLJ.INSTALL_JAR('/home/nuno.lages/Applications/commons-lang3-3.7.jar', 'APP.commons_lang', 0);
CALL SQLJ.INSTALL_JAR('/home/nuno.lages/Applications/SQlFunctions-1.0-SNAPSHOT.jar', 'APP.maxdata', 0);

CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.database.classpath', 'APP.commons_lang:APP.maxdata');

CREATE FUNCTION replaceChars (sourceString VARCHAR(8000), searchString VARCHAR(8000), replaceString VARCHAR(8000))
RETURNS VARCHAR(8000)
PARAMETER STYLE java NO SQL LANGUAGE java
EXTERNAL name 'org.apache.commons.lang3.StringUtils.replaceChars';

CREATE FUNCTION subtractTimestampsInDays(firstTs TIMESTAMP, secondTs TIMESTAMP)
RETURNS INT PARAMETER STYLE java NO SQL LANGUAGE java
EXTERNAL name 'pt.uminho.haslab.SQlFunctions.TimestampsOperations.subtractTimestampsInDays';
```

**Figure 7 – Apache Derby SQL script that defines a function to replace characters in strings and a function that calls the method shown in Figure 5**



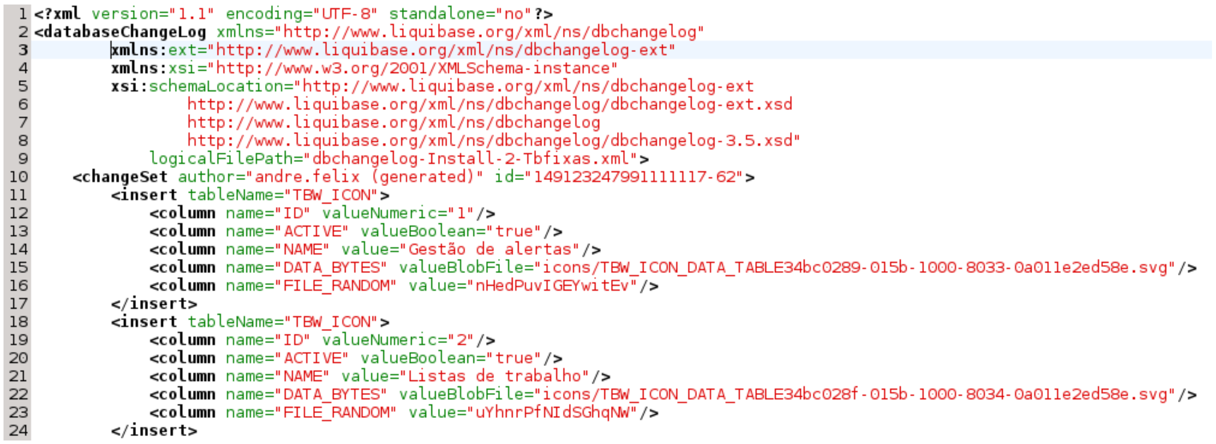
```java
package pt.maxdata.clinidata.server.utils;

import org.hibernate.dialect.DerbyTenSevenDialect;
import org.hibernate.dialect.function.StandardSQLFunction;
import pt.maxdata.clinidata.server.utils.ServerEnumerators.FunctionsEnum;

public class DerbyTenSevenDialectCustom extends DerbyTenSevenDialect {

    public DerbyTenSevenDialectCustom() {
        super();
        registerFunction(FunctionsEnum.TRANSLATE.getName(), new StandardSQLFunction("replaceChars"));
    }
}
```

**Figure 8 – Apache Derby dialect extension class in CLINIdATA®**



```xml
<?xml version="1.1" encoding="UTF-8" standalone="no"?>
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog-ext
            http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd
            http://www.liquibase.org/xml/ns/dbchangelog
            http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.5.xsd">
    <changeSet author="nuno.lages (generated)" id="1518718379338-1">
        <createSequence sequenceName="seq_dteattachmentfile_id"/>
    </changeSet>
    <changeSet author="nuno.lages (generated)" id="1518718379338-2">
        <createSequence sequenceName="seq_dtfbilldoc_id"/>
    </changeSet>
    <changeSet author="nuno.lages (generated)" id="1518718379338-3">
        <createSequence sequenceName="seq_dtfbilldocreq_id"/>
    </changeSet>
```

**Figure 9 – Beginning of the Liquibase file with information for the creation of CLINIdATA®'s schema**



```xml
<?xml version="1.1" encoding="UTF-8" standalone="no"?>
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog-ext
                http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd
                http://www.liquibase.org/xml/ns/dbchangelog
                http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.5.xsd"
            logicalFilePath="dbchangelog-Install-2-Tbfixas.xml">
    <changeSet author="andre.felix (generated)" id="149123247991111117-62">
        <insert tableName="TBW_ICON">
            <column name="ID" valueNumeric="1"/>
            <column name="ACTIVE" valueBoolean="true"/>
            <column name="NAME" value="Gestão de alertas"/>
            <column name="DATA_BYTES" valueBlobFile="icons/TBW_ICON_DATA_TABLE34bc0289-015b-1000-8033-0a011e2ed58e.svg"/>
            <column name="FILE_RANDOM" value="nHedPuvIGEYwitEv"/>
        </insert>
        <insert tableName="TBW_ICON">
            <column name="ID" valueNumeric="2"/>
            <column name="ACTIVE" valueBoolean="true"/>
            <column name="NAME" value="Listas de trabalho"/>
            <column name="DATA_BYTES" valueBlobFile="icons/TBW_ICON_DATA_TABLE34bc028f-015b-1000-8034-0a011e2ed58e.svg"/>
            <column name="FILE_RANDOM" value="uYhnrPfNIdSGhqNW"/>
        </insert>
```

**Figure 10 – Beginning of the Liquibase file with information for insertion of icons into CLINIdATA®'s database**

```java
import java.sql.*;


// CONNECT TO THE DATABASE
String url = "jdbc:derby://35.195.79.200:1527/maxdata";
Properties props = new Properties();
props.setProperty("user","**********");
props.setProperty("password","**********");

Connection conn = DriverManager.getConnection(url, props);

System.out.println("Connected to database.");

// LIST ALL TABLES IN THE DATABASE AND PREPARE CORRESPONDING UPDATE STATEMENTS
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery("SELECT TABLENAME FROM SYS.SYSTABLES");
List<String> updateQueries = new ArrayList<>();
String table = null;
while (rs.next()) {
    table = rs.getString("TABLENAME");
    if (!table.contains("SYS")) {
        // CALL SYSCS_UTIL.SYSCS_UPDATE_STATISTICS('CLINIDATA_NEW', 'DTE_ATTACHMENT_FILE', null);
        updateQueries.add("CALL SYSCS_UTIL.SYSCS_UPDATE_STATISTICS('CLINIDATA_NEW', '" + table + "', null)");
    }
}
rs.close();
st.close();

// EXECUTE UPDATE STATEMENTS
for (String s : updateQueries) {
    st = conn.createStatement();
    st.executeUpdate(s);
    st.close();
    System.out.println(s);
}
```

**Figure 11 – Java script that updates CLINIdATA®'s database indexes**

```java
import java.sql.*;

// CONNECT TO THE DATABASE
String url = "jdbc:derby://35.240.87.43:1527/maxdata";
Properties props = new Properties();
props.setProperty("user","**********");
props.setProperty("password","**********");

Connection conn = DriverManager.getConnection(url, props);

System.out.println("Connected to database.");

// LIST ALL SEQUENCES IN THE DATABASE
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM SYS.SYSSEQUENCES");
List<String> seqs = new ArrayList<>();
while (rs.next()) {
    seqs.add(rs.getString("SEQUENCENAME"));
}
rs.close();
st.close();

// SORT SEQ LIST ALPHABETICALLY
java.util.Collections.sort(seqs);
System.out.println("Got sequence names.");

List<Integer> failedIndexes = new ArrayList<>();
for (int i = 0; i < seqs.size(); i++) {
    Integer m = null, n = null;
    Statement st;
    ResultSet rs;

    // GET THE MAXIMUM VALUE OF TABLE KEY
    st = conn.createStatement();
    rs = st.executeQuery("SELECT MAX(id) FROM " + data_tables.get(i)); //data_tables was prepared upstream
    rs.next();
    m = rs.getInt(1);
    System.out.println(data_tables.get(i) + " " + m);
    m = m + 10;
    System.out.println("new m " + m);
    rs.close();
    st.close();

    // RECREATE CORRESPONDING SEQUENCE WITH THE CORRECT VALUE
    st = conn.createStatement();
    st.executeUpdate("DROP SEQUENCE " + seqs.get(i) + " RESTRICT");
    st.close();
    st = conn.createStatement();
    st.executeUpdate("CREATE SEQUENCE " + seqs.get(i) + " START WITH " + m + " MINVALUE " + m);
    st.close();

    // CHECK THAT SEQUENCE WAS CORRECTLY RECREATED
    st = conn.createStatement();
    String q = "SELECT CURRENTVALUE FROM SYS.SYSSEQUENCES WHERE SEQUENCENAME = '" + seqs.get(i) + "'";
    rs = st.executeQuery(q);
    rs.next();
    n = rs.getInt(1);
    if ( m != null && m.equals(n)) {
        System.out.println(seqs.get(i) + " updated correctly.");
    } else {
        System.out.println(seqs.get(i) + " update INCORRECT.");
    }
    rs.close();
    st.close();

}
```

**Figure 12 – Java script that updates CLINIdATA®'s database primary key-generating sequences**

The SQ1 Query Engine component is based on the open-source Apache Derby database, therefore we carried out tests with this database system for troubleshooting. Specific problems arose during integration of SQ1 with CLINIdATA® that were actually compatibility issues between CLINIdATA® and Apache Derby SQL.

Apache Derby can run as a database server accessible through a network connection or in embedded mode[12], in the same process (the same Java Virtual Machine instance) as CLINIdATA®. While Derby in embedded mode did not present additional challenges besides those already described, Derby Network Client [13] presented an additional limitation: CLINIdATA®'s data access layer relies on nested savepoints to avoid complete rollback of some transactions, but Derby Network Client does not support nested savepoints – as reported in Apache's bug tracking system[14]. However, Derby supports nested savepoints when running in embedded mode, therefore we used embedded Derby for testing purposes. The SQ1 and SQ2 solutions resort to the Derby network mode and, currently, do not support the functionalities of embedded mode. Hence, we provisionally deactivated nested savepoints in SQ1 (and SQ2) to circumvent this limitation.

We needed to define database procedures required by CLINIdATA® but not implemented in Apache Derby. These procedures calculate the number of days between two dates and replace specific characters in a string (Figure 7). The latter uses the replaceChars function from the open-source Apache Commons Lang[15] library. We had to write an extension to Hibernate's Apache Derby dialect to support this function.
Finally, we disabled Hibernate's auditing system[16]. Hibernate maintains a set of auditing tables that store information about the operations performed on the database. Creating and maintaining auditing tables requires queries that SQ1 does not yet support (for example 'ALTER TABLE COLUMN…') giving rise to exceptions. Disabling the auditing system prevented these exceptions.

The outcome of this work was a functional version of CLINIdATA® running on SQ1, which we tested extensively for a typical clinical laboratory use case. The use case comprised:

1) Logging into CLINIdATA®;
2) Placing an order for multiple clinical tests;
3) Inserting the results of the clinical tests, as a lab technician would do;
4) Validating the results of the clinical tests, as a pathologist would do.

We tested CLINIdATA®-SQ1 for this use case in several ways:

1. With the automated testing utility presented in D5.4, which allowed the user to execute it sequentially multiple times; for information about these tests, refer to D5.4.
2. In the performance tests in Section 3.2.3; the execution of this use case was recorded and then executed multiple times in parallel.
3. In the pilot (Section 3.2.4), where real healthcare users executed the scenario and reported on their experience.

### 3.2.2 Secure SQL Engine Integration Work

In D5.4 we identified two features required by CLINIdATA® that SafeCloud Secure Database Server didn't support: nested savepoints in transactions and storage of BLOBs, which was necessary to store CLINIdATA®'s user interface icons. The former is still

[12] Embedded Derby, https://db.apache.org/derby/papers/DerbyTut/embedded_intro.html
[13] Derby Network Client, https://db.apache.org/derby/papers/DerbyClientSpec.html
[14] *Make client driver allow nested savepoints*, https://issues.apache.org/jira/browse/DERBY-3687
[15] Apache Commons Lang, https://commons.apache.org/proper/commons-lang/
[16] Hibernate Auditing System, https://docs.jboss.org/envers/docs/#preface

missing: transactional operations still bypass savepoints, but this does not prevent CLINIdATA® from working correctly, and it is future work that it is outside the SafeClouds project's scope On the other hand, we implemented support for BLOBs and CLINIdATA® can now run on SQ1 with its full GUI.

We designed SafeCloud Secure Database Server to be application-agnostic and support most SQL query workloads. However, data encryption and decryption inevitably adds overhead to the operations, hence, it is desirable to identify and encrypt only the data that is sufficient to meet the required privacy-preserving security level. This way, it is possible to instantiate the database with a set of privacy-preserving techniques that offer the ideal compromise between privacy and performance. For this end, we identified the database table columns that contain sensitive data or store information without which it is not possible to determine patient identity – Table 1.

| TABLE | COLUMN | DESCRIPTION |
|---|---|---|
| **DTW_PATIENT** | **NAME** | Name of the patient. |
| **DTW_PATIENT** | **CALCULATED_BIRTHDATE_STAMP** | Calculated birthdate for the patient in particular cases where the patient does not know the exact birthdate. |
| **DTW_PATIENT** | **BIRTHDATE_STAMP** | Patient's birthdate. |
| **DTW_PATIENT_ID_BY_PATIENT** | **SUBJECT_ID** | Set of IDs that identifies a patient, e.g., social security number, citizen id. |
| **DTW_TEST_RESULT** | **NORMAL_VALUE** | Value of a specific result for an clinical test. |

**Table 1 – Columns with encrypted data in SafeCloud SQ1.**

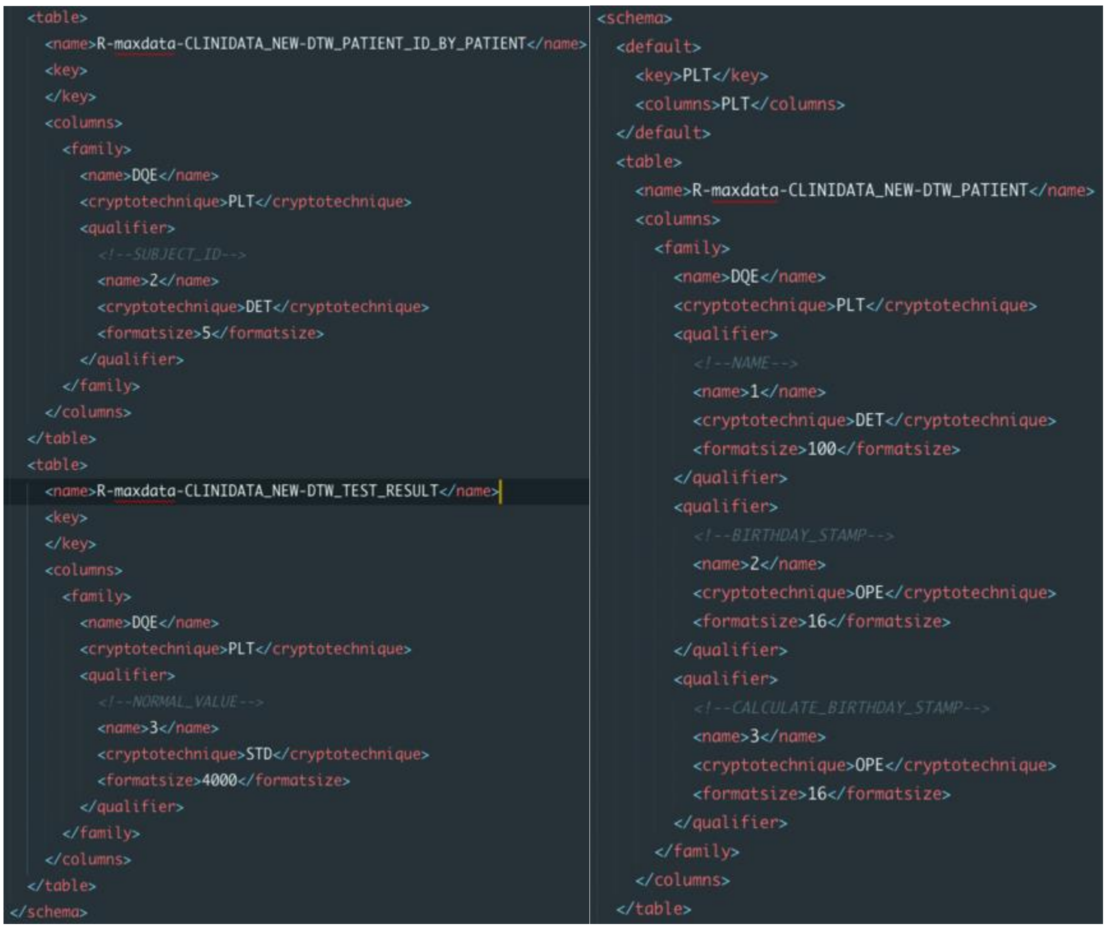
With the information from Table 1, we produced a configuration file for SQ1 to encrypt all data in these columns (Figure 13). The configuration file does not contain any explicit reference to the tables and columns in the relational schema: SQ1 and SQ2 map the relational SQL schema, including the name of the tables (Table 2) and columns (Table 3), to NoSQL structures and translate SQL queries to secure NoSQL operations that execute on an untrusted third-party cloud infrastructure. In this configuration file, the user can specify cryptographic techniques for each column or table. The user may also choose a cryptographic technique as default to avoid having to set cryptographic techniques for each column (the data may be organized in many columns), allowing them to focus on the configuration and the data that is most sensitive and important to protect.

| SQL Table | NoSQL Table |
|---|---|
| DTW_PATIENT | R-maxdata-CLINIDATA_NEW-DTW_PATIENT |
| DTW_PATIENT_ID_BY_PATIENT | R-maxdata-CLINIDATA_NEW-DTW_PATIENT_ID_BY_PATIENT |
| DTW_TEST_RESULT | R-maxdata-CLINIDATA_NEW-DTW_TEST_RESULT |

**Table 2 - SQL to NoSQL table translation.**

| SQL Columns | NoSQL Columns (Column Family – Qualifier) |
|---|---|
| NAME | DQE-1 |
| BIRTHDAY_STAMP | DQE-2 |
| CALCULATE_BIRTHDAY_STAMP | DQE-3 |
| SUBJECT_ID | DQE-2 |
| NORMAL_VALUE | DQE-3 |

**Table 3 - SQL to NoSQL Columns translation.**



```
<schema>
  <default>
    <key>PLT</key>
    <columns>PLT</columns>
  </default>
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_PATIENT</name>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--NAME-->
          <name>1</name>
          <cryptotechnique>DET</cryptotechnique>
          <formatsize>100</formatsize>
        </qualifier>
        <qualifier>
          <!--BIRTHDAY_STAMP-->
          <name>2</name>
          <cryptotechnique>OPE</cryptotechnique>
          <formatsize>16</formatsize>
        </qualifier>
        <qualifier>
          <!--CALCULATE_BIRTHDAY_STAMP-->
          <name>3</name>
          <cryptotechnique>OPE</cryptotechnique>
          <formatsize>16</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
```

```
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_PATIENT_ID_BY_PATIENT</name>
    <key>
    </key>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--SUBJECT_ID-->
          <name>2</name>
          <cryptotechnique>DET</cryptotechnique>
          <formatsize>5</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_TEST_RESULT</name>
    <key>
    </key>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--NORMAL_VALUE-->
          <name>3</name>
          <cryptotechnique>STD</cryptotechnique>
          <formatsize>4000</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
</schema>
```

**Figure 13 – SafeCloud Query Engine Configuration File**

### 3.2.3 Performance Tests

We assessed the performance overhead of the data security features introduced in SQ1 and SQ2 with respect to equivalent non-secure technology. Briefly, we resorted to a baseline database deployment that leverages the Query Engine and NoSQL components used in SQ1 and SQ2 but that does not provide any privacy-preserving guarantees. With such approach, it was possible to observe the direct impact of the novel privacy-preserving mechanisms proposed in the SafeCloud project.

We deployed CLINIdATA® + SQ1 and CLINIdATA® + SQ2 to Ubuntu Linux 16.04 LTS virtual machines running on Google Cloud Platform (GCP)[17]. We tuned the processing power and the maximum allowed memory for all processes not to limit their performance: we used 8-core Intel Sandy Bridge virtual processors for SQ1 and 16-core Intel Sandy Bridge virtual processors for SQ2. In both cases, the virtual machines were allocated 32GB of RAM.

We simulated CLINIdATA usage with the tool Apache JMeter[18]. We ran JMeter from a separate virtual machine, to avoid sharing of resources between JMeter and the CLINIdATA® + SQ1/SQ2 deployment, but in the same region and zone, to minimize any effect of network latency on the measurements. We configured JMeter to simulate the complete demonstration procedure described in Section 4 a specified number of consecutive times by a specified number of users. We chose the configuration of the tests performed with JMeter to approach the typical workloads for small and large healthcare organizations described in D5.2, Section 3.1.1. The deployments and configurations used in the tests, as well as, the results are further detailed in D4.5, Section 6.2.1.1.

### 3.2.4 Pilot

We exposed SafeCloud Healthcare Platform, namely the CLINIdATA® + SQ1 solution, to use by actual healthcare professionals, in a realistic pilot SaaS scenario deployment.

For this pilot we deployed CLINIdATA® and SQ1 to an 8-core Intel Sandy Bridge virtual Central Processing Units (CPUs), 24GB virtual machine running Ubuntu Linux 16.04 LTS on GCP. We ran CLINIdATA® on a complete installation of SQ1, with a single Apache HBase[19] node, Secure Query Engine and Apache Omid[20] transaction support. We configured CLINIdATA® for use in the scenario demonstrated in Section 4.

We prepared a guide with screenshots to guide the users through the scenario and made it available as a Google Presentation and as a PDF document. We also prepared a Google Form to collect user feedback.

D4.5, Section 6.2.2 describes in detail the documents, procedures and results of the pilot.

## 3.3 Integration of Knowage / CLINIdATA® BI with SafeCloud SQ3

For clients who wish to analyse their data, Maxdata's usual approach is to deploy a third-party open-source BI solution, Knowage, to the client's premises. Knowage is a software suite that, among other features, can query relational databases, generate reports and display results in dashboards. The data used in the reports and dashboards may come from queries to CLINIdATA®'s database or from queries to a custom Online Analytical Processing (OLAP) cube periodically populated from CLINIdATA®'s database by an ETL process.

In the use case considered for SafeCloud Healthcare Platform, several healthcare providers wish to perform data analysis on their combined data without sharing that data,

[17] https://cloud.google.com/
[18] https://jmeter.apache.org/
[19] https://hbase.apache.org/
[20] https://omid.incubator.apache.org/

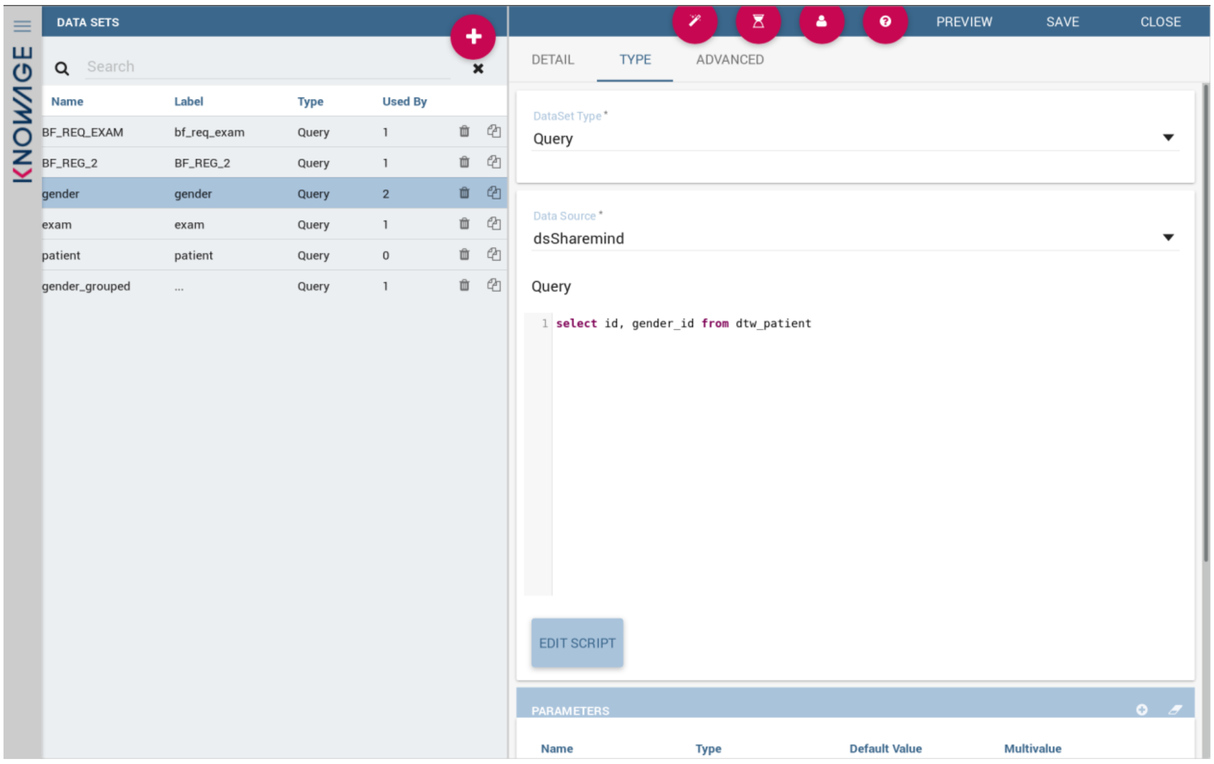
to avoid any chance of compromising confidentiality. SQ3 provides a solution for this use case with the following architecture:

- At least 3 of the organizations sharing analytics host SQ3 servers.
- Each participating healthcare provider has its own installation of SQ3 Data Importer in a server of their choice, in addition to the regular deployment of CLINIdATA® (with CLINIdATA® using a regular third-party relational database – e.g. Oracle or PostgreSQL).
- The relevant data to be analysed is loaded periodically from CLINIdATA®'s database into SQ3 through SQ3 Data Importer.
- Each healthcare provider also has its own deployment of Knowage and SQ3 SQL Proxy on a server of their choice, with their own custom reports and dashboards. Knowage queries the distributed SQ3 installation for aggregate data through the SQL Proxy. SQ3 processes the query securely in the background, returning aggregate results from all participant providers. Knowage presents the results to its users in reports and dashboards.

Knowage's developers made Knowage a generic BI tool only with basic security concerns. For efficiency, Knowage keeps a cache with data prefetched from the database. The most significant configuration change for this integration was disabling Knowage's caching mechanism, since SQ3 would not allow prefetching of non-aggregate non-encrypted data. Otherwise, dashboard and report configuration was as straightforward as for any other database.

Figure 14 to Figure 17 show an example query running on SQ3. In this example, a user wants to assess the gender distribution of a combined population of patients. For this end, the user defines an appropriate SQL query in Knowage to retrieve the count of patients grouped by gender. In the example, in Figure 14, the SQL query retrieves the count of patients for each gender from a table (dtw_patient) that contains sensitive data that can uniquely identify patients. Note that the query does not contain the aggregate function COUNT or the GROUP BY clause because in Knowage aggregations are defined elsewhere, outside the SQL text.

SQ3 receives and processes the query securely. Figure 15 shows part of the output of SQ3 while processing the query. When the execution of the query completes, SQ3 returns the aggregate result to Knowage.



| Name | Label | Type | Used By |
|---|---|---|---|
| BF_REQ_EXAM | bf_req_exam | Query | 1 |
| BF_REG_2 | BF_REG_2 | Query | 1 |
| gender | gender | Query | 2 |
| exam | exam | Query | 1 |
| patient | patient | Query | 0 |
| gender_grouped | ... | Query | 1 |

DataSet Type *
Query

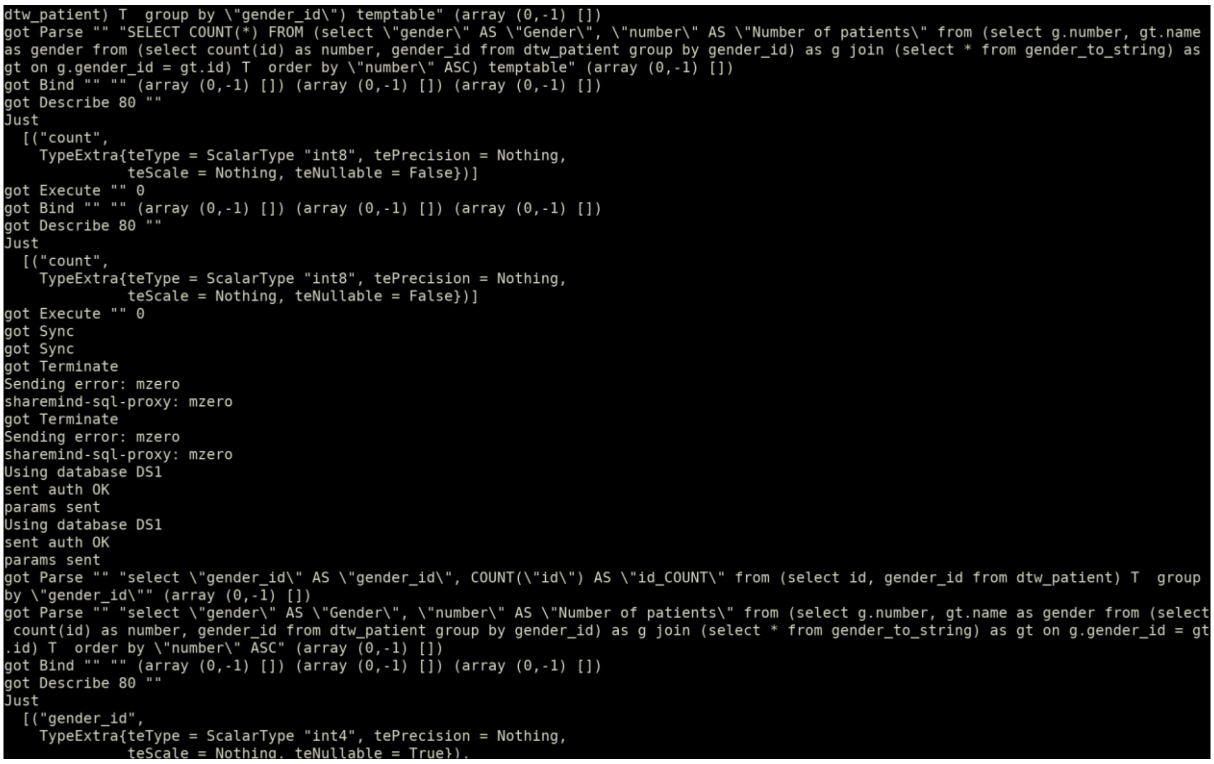
Data Source *
dsSharemind

Query

```
select id, gender_id from dtw_patient
```

| Name | Type | Default Value | Multivalue |
|---|---|---|---|

Figure 14 – SQL query edition page in Knowage



```
dtw_patient) T  group by \"gender_id\") temptable" (array (0,-1) [])
got Parse "" "SELECT COUNT(*) FROM (select \"gender\" AS \"Gender\", \"number\" AS \"Number of patients\" from (select g.number, gt.name
as gender from (select count(id) as number, gender_id from dtw_patient group by gender_id) as g join (select * from gender_to_string) as
gt on g.gender_id = gt.id) T  order by \"number\" ASC) temptable" (array (0,-1) [])
got Bind "" "" (array (0,-1) []) (array (0,-1) []) (array (0,-1) [])
got Describe 80 ""
Just
  [("count",
    TypeExtra{teType = ScalarType "int8", tePrecision = Nothing,
              teScale = Nothing, teNullable = False})]
got Execute "" 0
got Bind "" "" (array (0,-1) []) (array (0,-1) []) (array (0,-1) [])
got Describe 80 ""
Just
  [("count",
    TypeExtra{teType = ScalarType "int8", tePrecision = Nothing,
              teScale = Nothing, teNullable = False})]
got Execute "" 0
got Sync
got Sync
got Terminate
Sending error: mzero
sharemind-sql-proxy: mzero
got Terminate
Sending error: mzero
sharemind-sql-proxy: mzero
Using database DS1
sent auth OK
params sent
Using database DS1
sent auth OK
params sent
got Parse "" "select \"gender_id\" AS \"gender_id\", COUNT(\"id\") AS \"id_COUNT\" from (select id, gender_id from dtw_patient) T  group
by \"gender_id\"" (array (0,-1) [])
got Parse "" "select \"gender\" AS \"Gender\", \"number\" AS \"Number of patients\" from (select g.number, gt.name as gender from (select
 count(id) as number, gender_id from dtw_patient group by gender_id) as g join (select * from gender_to_string) as gt on g.gender_id = gt
.id) T  order by \"number\" ASC" (array (0,-1) [])
got Bind "" "" (array (0,-1) []) (array (0,-1) []) (array (0,-1) [])
got Describe 80 ""
Just
  [("gender_id",
    TypeExtra{teType = ScalarType "int4", tePrecision = Nothing,
              teScale = Nothing, teNullable = True}),
```

Figure 15 – Terminal output of SQ3 processing a query for the number of patients of each gender

The user can define a variety of reports and dashboard widgets in Knowage to inspect the results – Figure 16 shows the chart configuration page in Knowage. In our example, the user chose to visualize the gender distribution on a table and a pie chart (Figure 17).

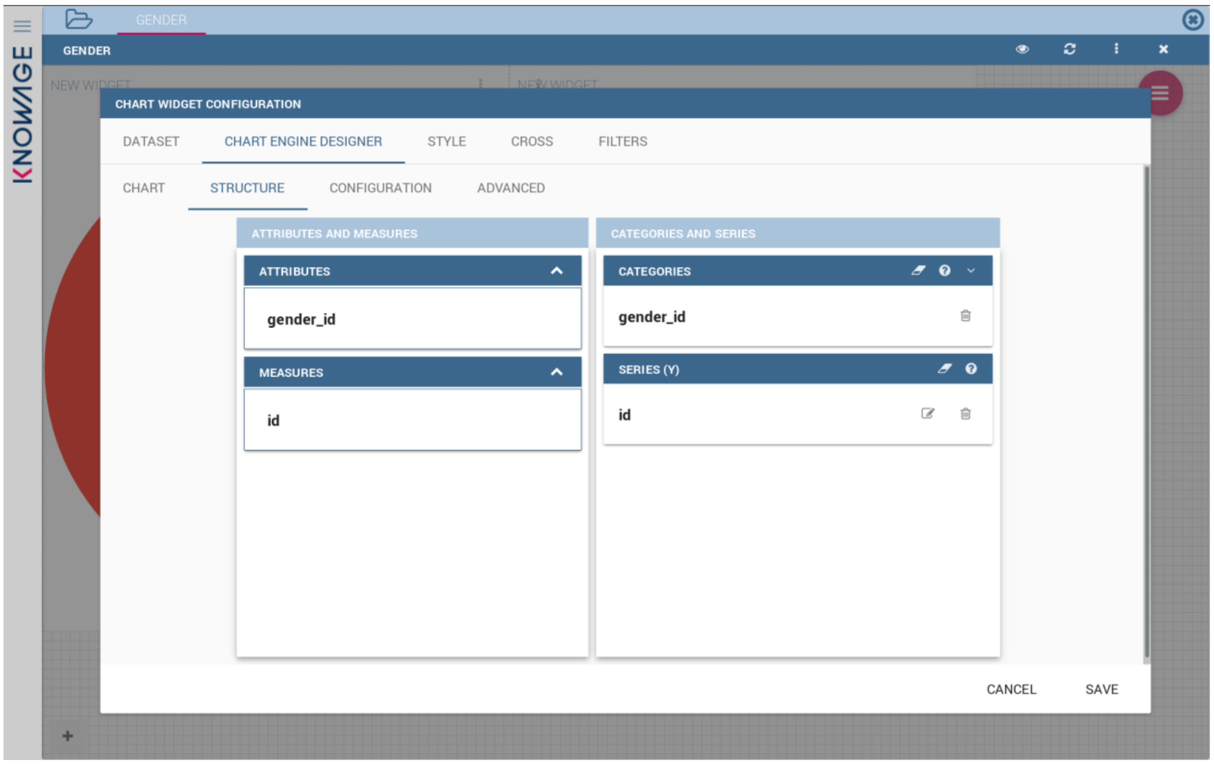
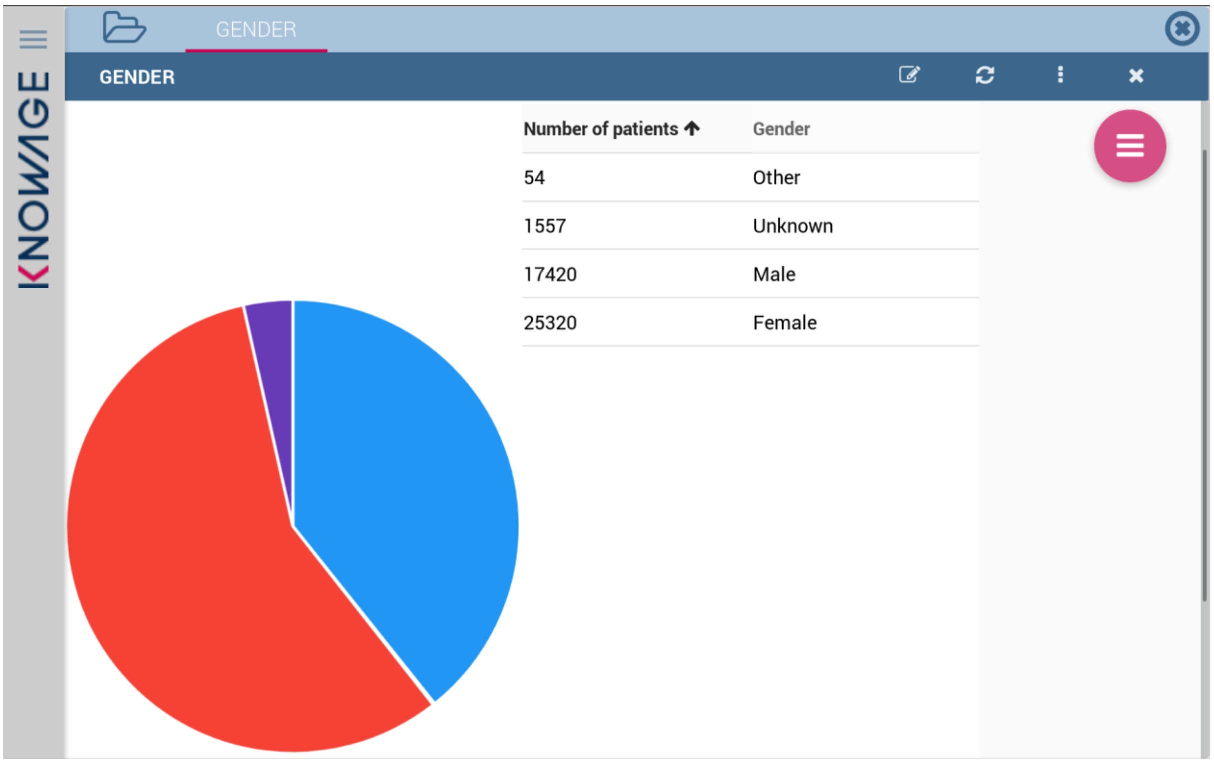


Figure 16 – Chart configuration page in Knowage



| Number of patients | Gender |
|---|---|
| 54 | Other |
| 1557 | Unknown |
| 17420 | Male |
| 25320 | Female |

Figure 17 – Knowage dashboard showing a table and a pie chart for our example

# 4 Demonstration Procedure

This section presents how a CLINIdATA® user executes the tasks from the use case described in Section 3.2.1, used for demonstration, load tests and the pilot of the SafeCloud Healthcare Platform.

## 4.1 Placing an order for clinical tests

This demonstration procedure concerns use case 3.2.3.3 described in Deliverable D5.2. After insertion of user credentials and successful login to the application (Figure 18), CLINIdATA®'s main screen appears to the user. As we can see in Figure 19, CLINIdATA®'s main screen looks like a classic desktop environment, with a start menu, shortcuts, a taskbar, clock, message system and help system icons.

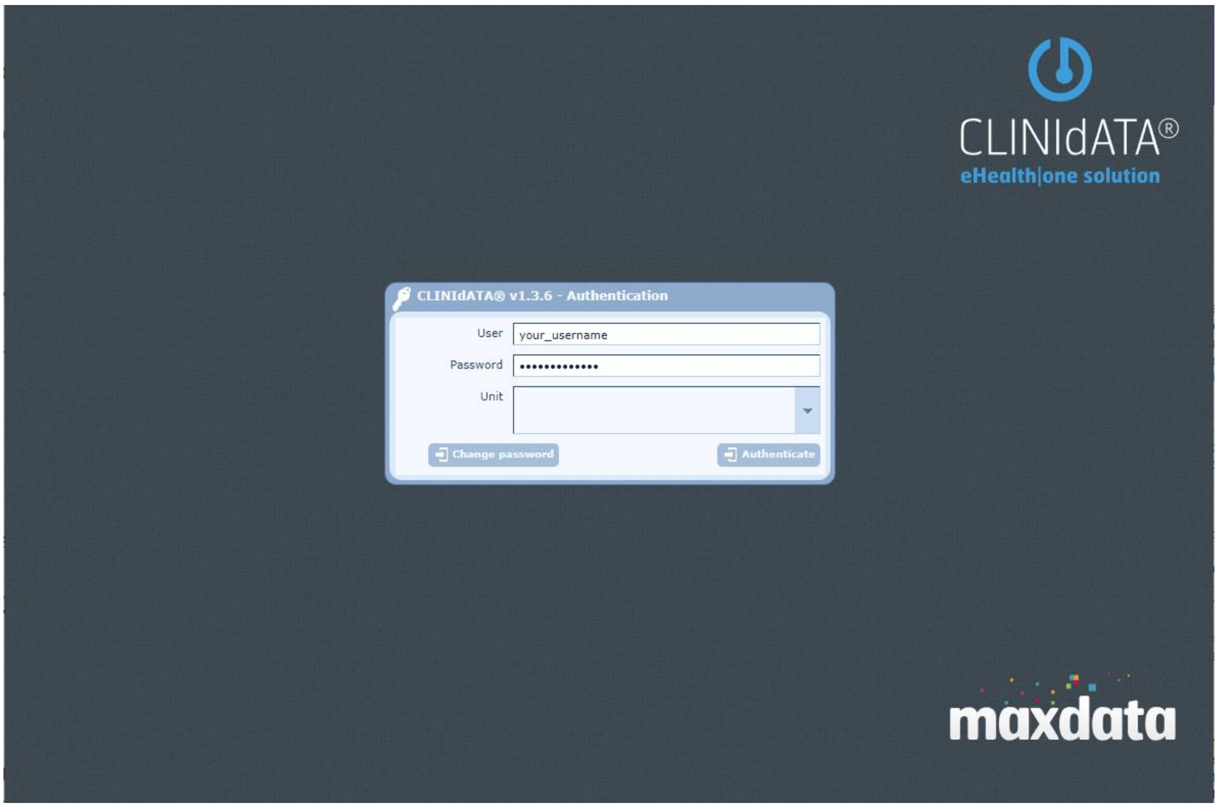


Figure 18 – CLINIdATA®'s Login Screen

A typical clinical laboratory scenario starts with placing an order for clinical tests, usually by an administrative at the reception desk. When a patient arrives, the administrative searches for them in the system and places the appropriate order.

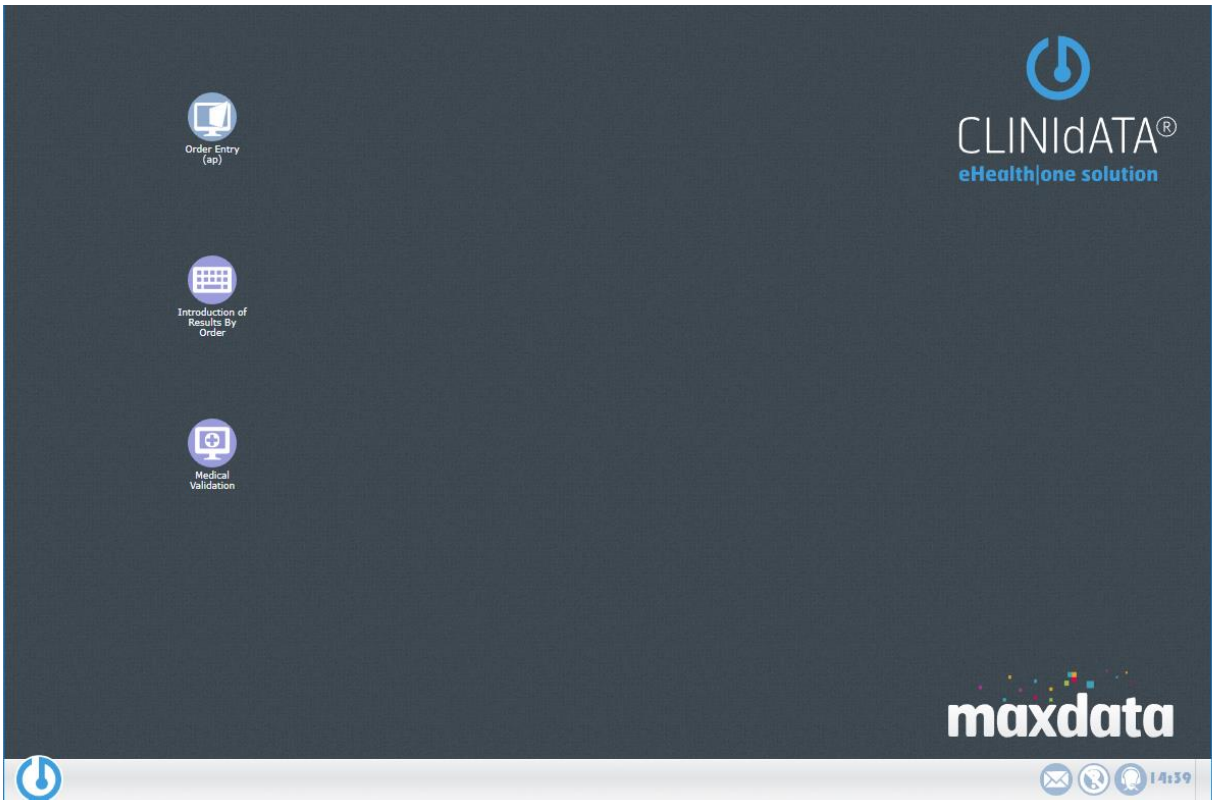
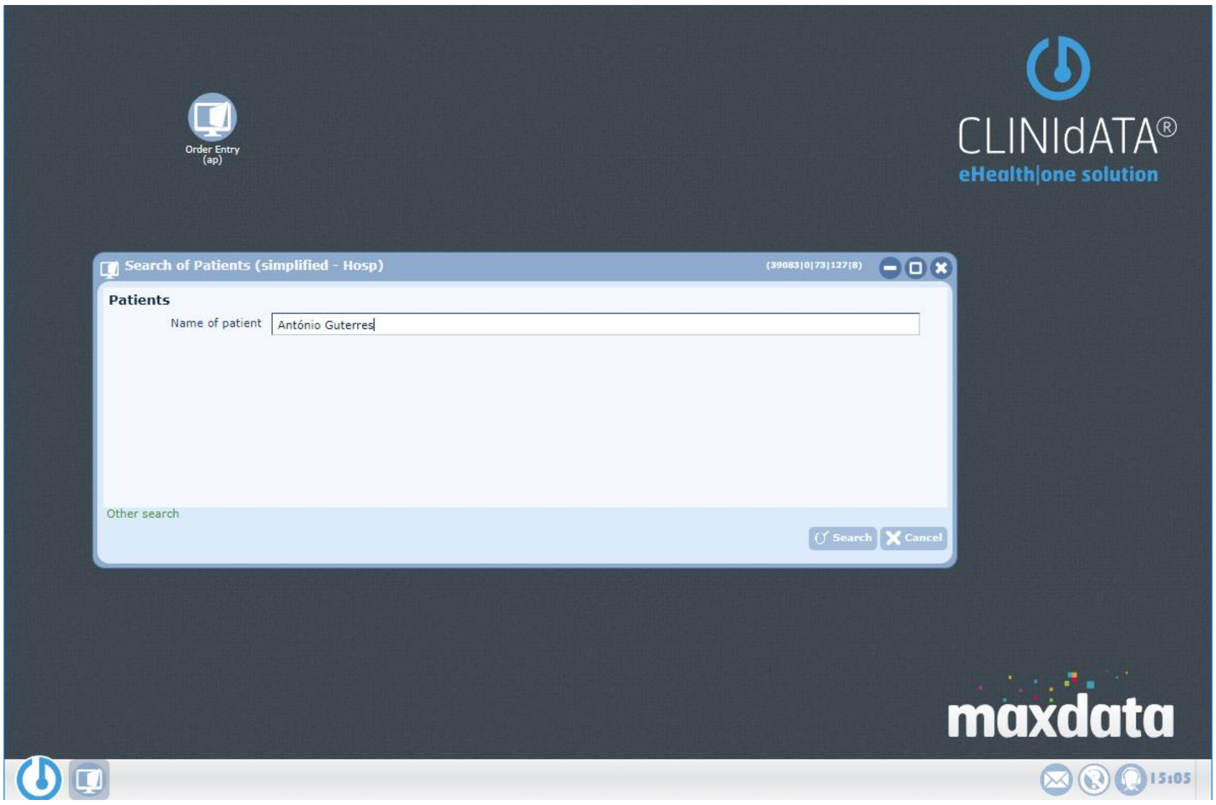


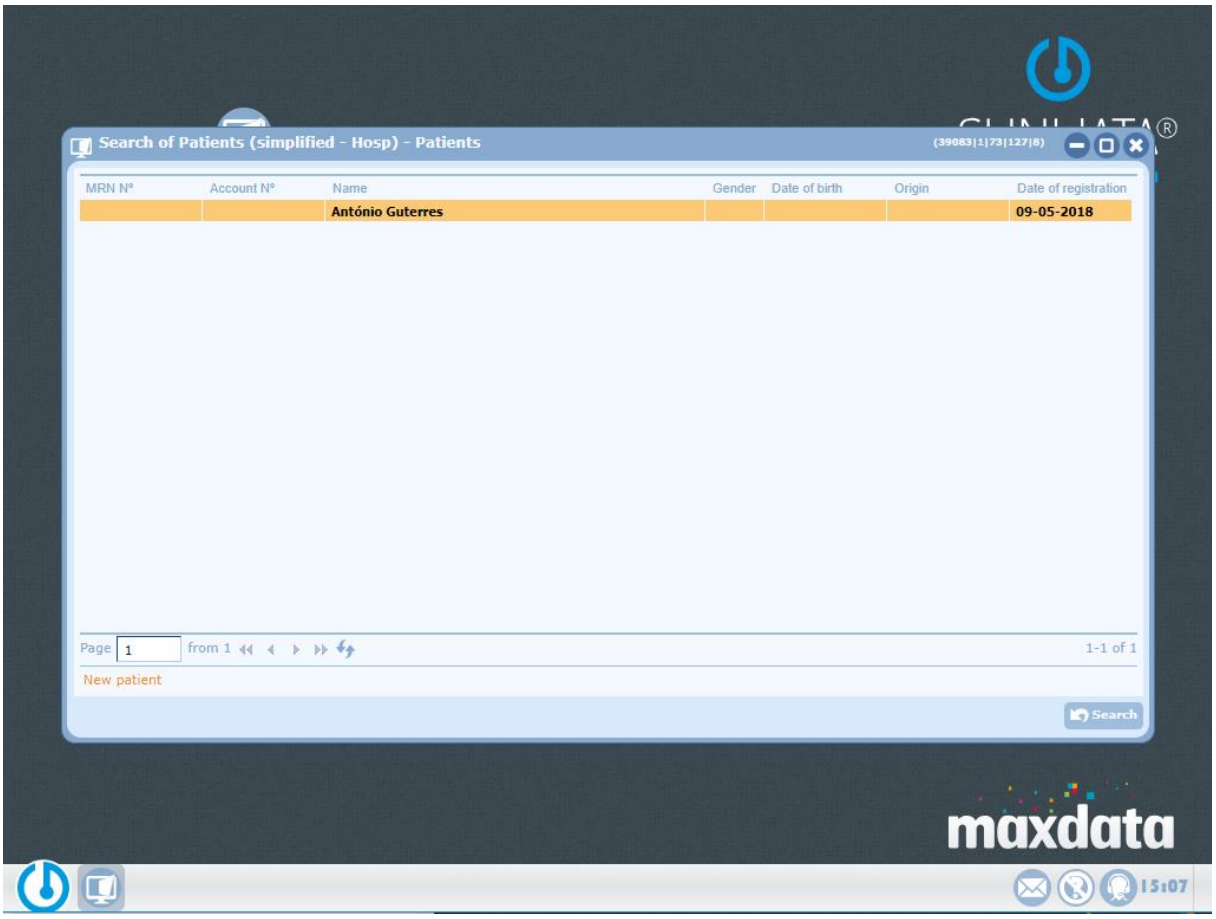
**Figure 19 – CLINIdATA®'s main screen**

The first step is searching for the patient in the system, for instance 'António Guterres' (Figure 20). If a patient with such name exists in the system, CLINIdATA® opens a table in a dedicated window (Figure 21), where the user can select the intended patient. Selection of the patient triggers the opening of the ordering window, where the user defines the order for clinical tests. If the patient does not exist in the system, CLINIdATA® immediately opens the window to define the order.



Name of patient: António Guterres

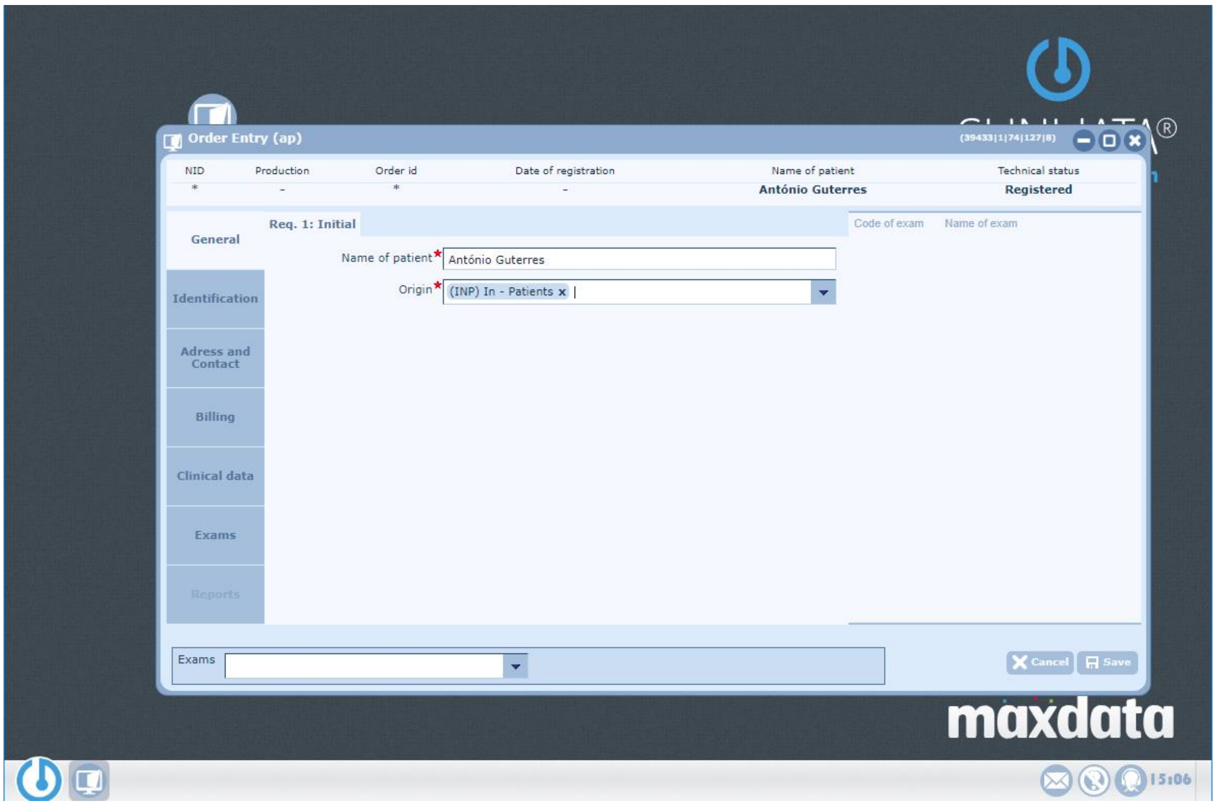
**Figure 20 – Window used to search for a patient**



| MRN Nº | Account Nº | Name | Gender | Date of birth | Origin | Date of registration |
|---|---|---|---|---|---|---|
| | | António Guterres | | | | 09-05-2018 |

**Figure 21 – Results for a search by patient name**

The user may use checkboxes or a dropdown menu to select the intended clinical tests. This window also holds fields for a variety of patient data, which may be prefilled if there is information for the patient the system (Figure 22 and Figure 23). Selected tests appear on a grid on the right side of the screen (Figure 24). Finally, the user places the order by clicking the Save button. Unless an error occurs, the system shows a message informing the user that the order was placed successfully (Figure 25). In the example in the figures, the order was given id 148. This id is important to identify the order later, in the laboratory.



| NID | Production | Order id | Date of registration | Name of patient | Technical status |
|---|---|---|---|---|---|
| * | - | * | - | António Guterres | Registered |

Req. 1: Initial

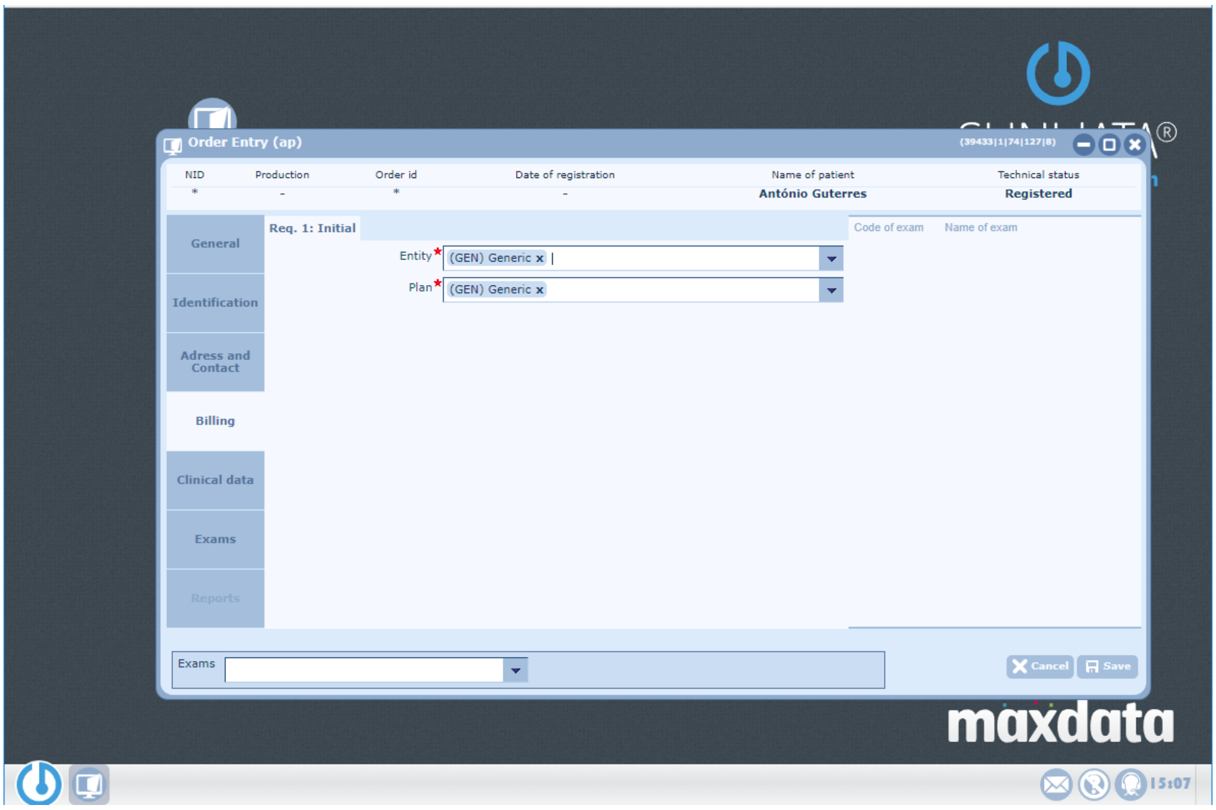
Name of patient* António Guterres

Origin* (INP) In - Patients

| Code of exam | Name of exam |
|---|---|

Exams

Figure 22 – Order entry window, general information tab

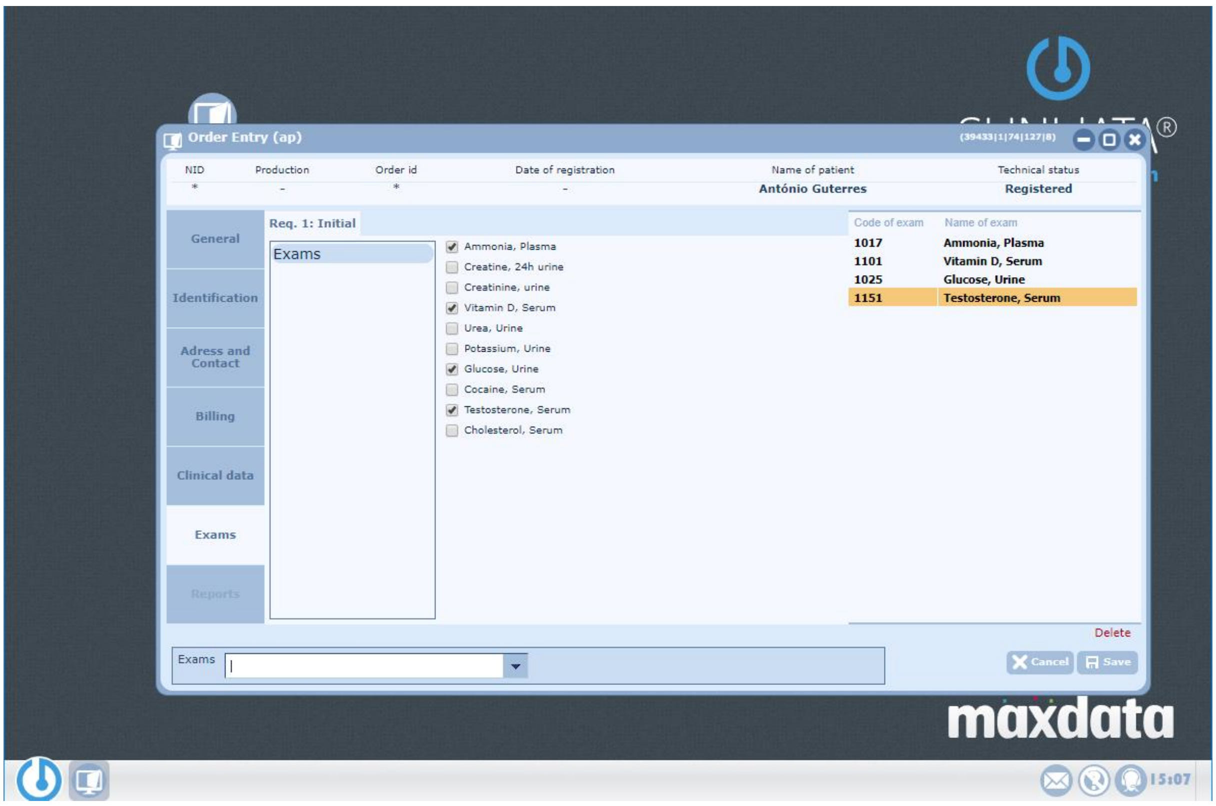


| NID | Production | Order id | Date of registration | Name of patient | Technical status |
|---|---|---|---|---|---|
| * | - | * | - | António Guterres | Registered |

Entity* (GEN) Generic x

Plan* (GEN) Generic x

Figure 23 – Order entry window, billing information tab

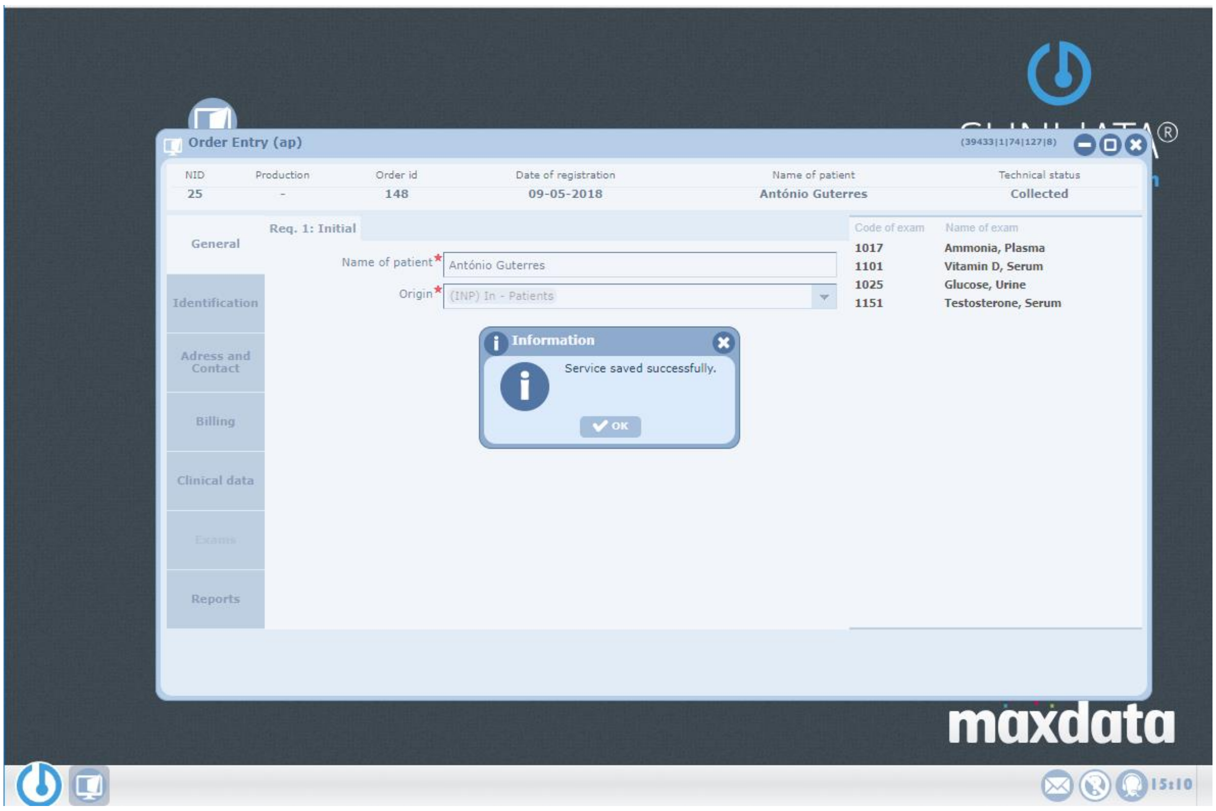


| NID | Production | Order id | Date of registration | Name of patient | Technical status |
|---|---|---|---|---|---|
| * | - | * | - | António Guterres | Registered |

- [✓] Ammonia, Plasma
- [ ] Creatine, 24h urine
- [ ] Creatinine, urine
- [✓] Vitamin D, Serum
- [ ] Urea, Urine
- [ ] Potassium, Urine
- [✓] Glucose, Urine
- [ ] Cocaine, Serum
- [✓] Testosterone, Serum
- [ ] Cholesterol, Serum

| Code of exam | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

Figure 24 – Clinical test (exam) selection


Order Entry (ap)

(39433|1|74|127|8)

| NID | Production | Order id | Date of registration | Name of patient | Technical status |
|---|---|---|---|---|---|
| 25 | - | 148 | 09-05-2018 | António Guterres | Collected |

Req. 1: Initial

Name of patient* António Guterres

Origin* (INP) In - Patients

| Code of exam | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

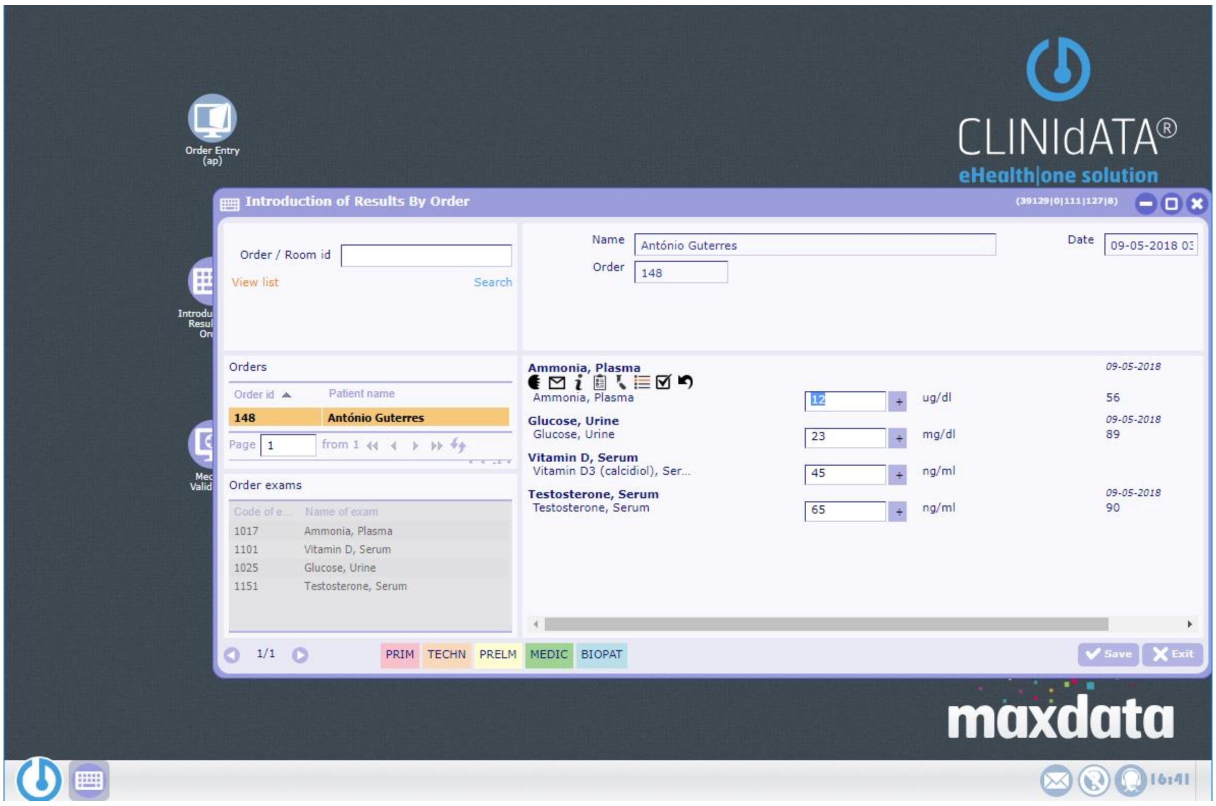
Information

Service saved successfully.

Figure 25 – Order successfully saved

## 4.2 Introducing Clinical Test Results

This demonstration procedure concerns use-case 3.2.3.3, described in Deliverable D5.2. After successful placement of an order, the laboratory carries out the ordered tests and a lab technician inserts the results in CLINIdATA®. For this purpose, the technician may use the Introduce Results by Order screen (Figure 26). In this screen the user searches for an order by its id number. If the search yields results, the screen shows information about the patient and opens input fields for the results (Figure 27). In our example, the user inputs results for ammonia concentration in the plasma, glucose concentration in the urine, vitamin D in the blood serum and testosterone in the blood serum. The user must click the Save button to record the results. Upon saving, the system highlights the titles of the tests in orange, indicating that the results are validated by a technician, as encoded in the colour legend at the bottom of the screen.



Order / Room id

Name: António Guterres
Order: 148
Date: 09-05-2018 03

Orders

| Order id | Patient name |
|---|---|
| 148 | António Guterres |

Page 1 from 1

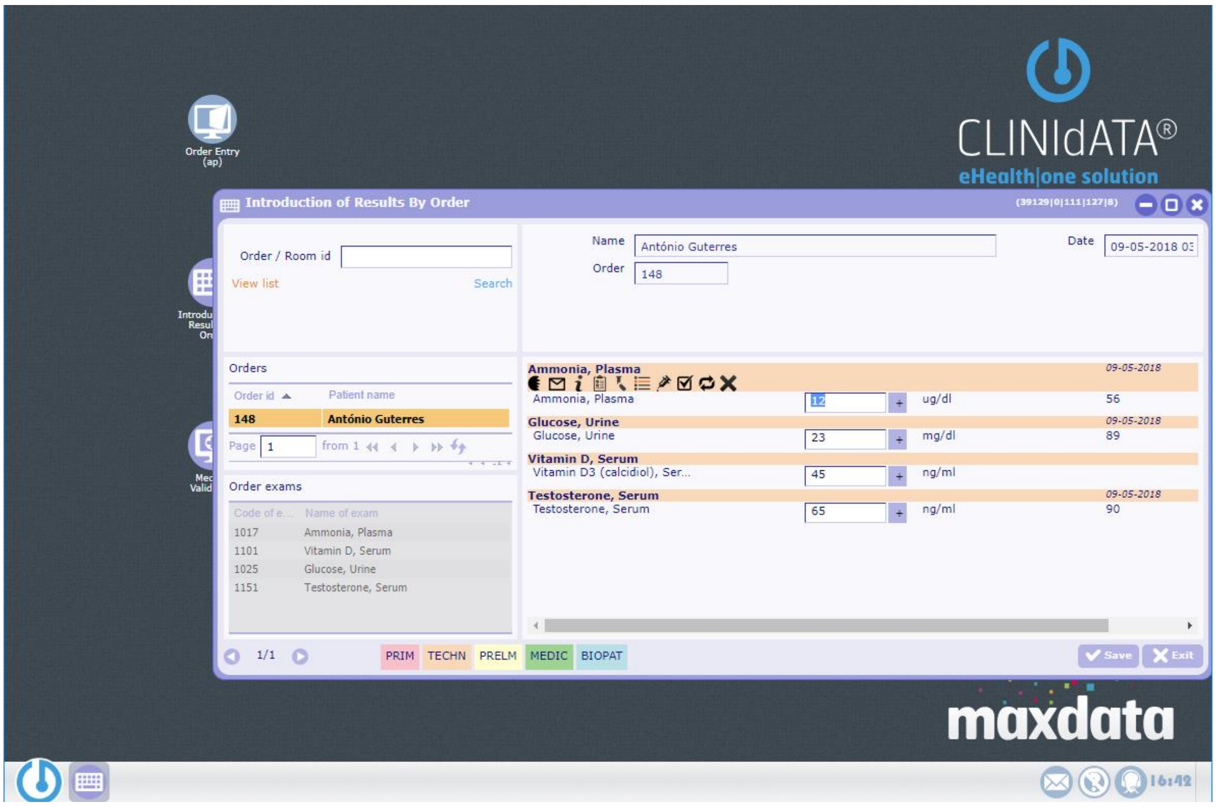
Order exams

| Code of e... | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

| Exam | Result | Unit | Previous date | Previous result |
|---|---|---|---|---|
| Ammonia, Plasma<br>Ammonia, Plasma | 12 | ug/dl | 09-05-2018 | 56 |
| Glucose, Urine<br>Glucose, Urine | 23 | mg/dl | 09-05-2018 | 89 |
| Vitamin D, Serum<br>Vitamin D3 (calcidiol), Ser... | 45 | ng/ml | | |
| Testosterone, Serum<br>Testosterone, Serum | 65 | ng/ml | 09-05-2018 | 90 |

1/1

**Figure 26 – Window for insertion of clinical test results**



Order / Room id

Name: António Guterres
Order: 148
Date: 09-05-2018 03

Orders

| Order id | Patient name |
|---|---|
| 148 | António Guterres |

Page 1 from 1

Order exams

| Code of e... | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

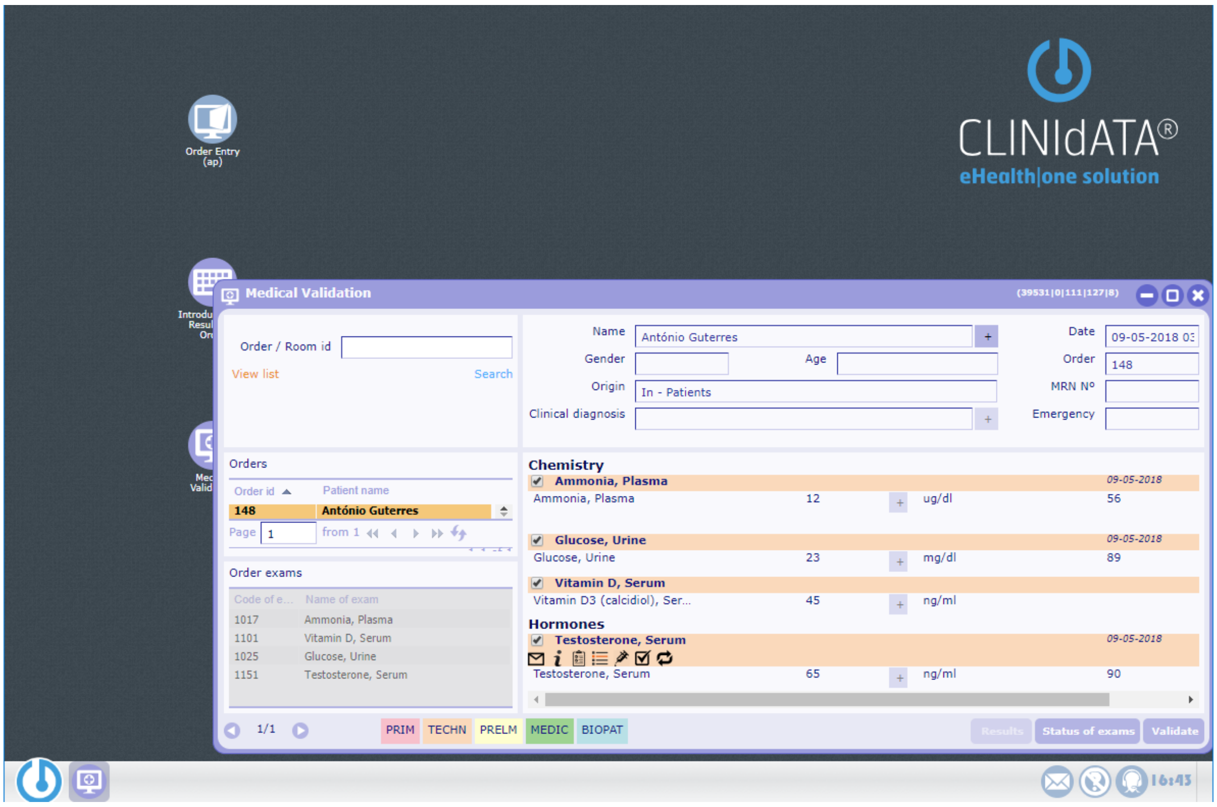
| Exam | Result | Unit | Previous date | Previous result |
|---|---|---|---|---|
| Ammonia, Plasma<br>Ammonia, Plasma | 12 | ug/dl | 09-05-2018 | 56 |
| Glucose, Urine<br>Glucose, Urine | 23 | mg/dl | 09-05-2018 | 89 |
| Vitamin D, Serum<br>Vitamin D3 (calcidiol), Ser... | 45 | ng/ml | | |
| Testosterone, Serum<br>Testosterone, Serum | 65 | ng/ml | 09-05-2018 | 90 |

1/1

**Figure 27 – Window for insertion of clinical test results after saving/technical validation**

## 4.3 Validation of Clinical Test Results

This demonstration procedure concerns use case 3.2.3.3 described in Deliverable D5.2. The results of the ordered clinical tests are now in the system and need to be approved (or not) by a medical doctor. In CLINIdATA® there are two levels of validation by medical doctors: 'medical' and 'biopathological'. To validate results, the doctor opens the Medical Validation Window (Figure 28), selects individual results by checking the corresponding checkboxes and clicks the Validate button. Upon successful validation, the result entries become highlighted in blue or green (Figure 29), depending on whether the validation was 'medical' or 'biopathological'. In the example of Figure 29, the validation was 'medical'.



| | | | |
|---|---|---|---|
| Order / Room id | | | |
| Name | António Guterres | Date | 09-05-2018 03 |
| Gender | | Age | |
| Order | 148 | | |
| Origin | In - Patients | MRN Nº | |
| Clinical diagnosis | | Emergency | |

Orders

| Order id | Patient name |
|---|---|
| 148 | António Guterres |

Page 1 from 1

Order exams

| Code of e... | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

**Chemistry**

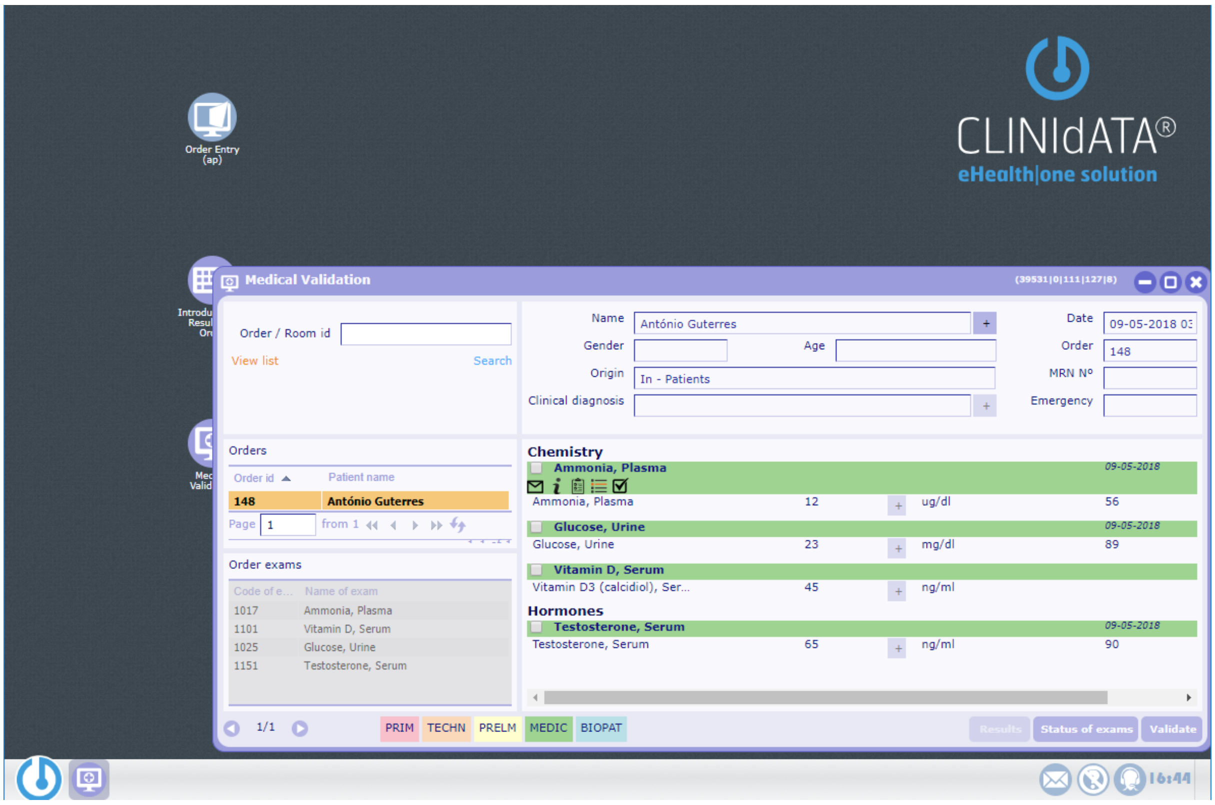
| | | | |
|---|---|---|---|
| ✓ **Ammonia, Plasma** | | | 09-05-2018 |
| Ammonia, Plasma | 12 | ug/dl | 56 |
| ✓ **Glucose, Urine** | | | 09-05-2018 |
| Glucose, Urine | 23 | mg/dl | 89 |
| ✓ **Vitamin D, Serum** | | | |
| Vitamin D3 (calcidiol), Ser... | 45 | ng/ml | |

**Hormones**

| | | | |
|---|---|---|---|
| ✓ **Testosterone, Serum** | | | 09-05-2018 |
| Testosterone, Serum | 65 | ng/ml | 90 |

1/1

Figure 28 – Window for validation of clinical test results (before validation)



Order / Room id

| | | | |
|---|---|---|---|
| Name | António Guterres | Date | 09-05-2018 03 |
| Gender | | Order | 148 |
| Age | | MRN Nº | |
| Origin | In - Patients | | |
| Clinical diagnosis | | Emergency | |

Orders

| Order id | Patient name |
|---|---|
| 148 | António Guterres |

Page 1 from 1

Order exams

| Code of e... | Name of exam |
|---|---|
| 1017 | Ammonia, Plasma |
| 1101 | Vitamin D, Serum |
| 1025 | Glucose, Urine |
| 1151 | Testosterone, Serum |

**Chemistry**

| | | | |
|---|---|---|---|
| **Ammonia, Plasma** | | | 09-05-2018 |
| Ammonia, Plasma | 12 | ug/dl | 56 |
| **Glucose, Urine** | | | 09-05-2018 |
| Glucose, Urine | 23 | mg/dl | 89 |
| **Vitamin D, Serum** | | | |
| Vitamin D3 (calcidiol), Ser... | 45 | ng/ml | |

**Hormones**

| | | | |
|---|---|---|---|
| **Testosterone, Serum** | | | 09-05-2018 |
| Testosterone, Serum | 65 | ng/ml | 90 |

1/1

**Figure 29 – Window for validation of clinical test results (after medical validation)**

# 5 GDPR compliance

This section describes how healthcare providers should proceed when implementing the SafeCloud-based healthcare platform, to ensure GDPR compliance. The section is divided in two parts:

- Section 5.1 describes how GDPR recommendations on data life cycle, described in D2.6, can be implemented in the context of the SafeCloud-based healthcare platform;
- Section 5.2 describes how to comply with specific articles of GDPR, following recommendation 5 of SafeCloud project's second year review.

## 5.1 Data Life Cycle

As explained in D2.6, data protection principles apply to all stages of data processing. In other words, duties imposed by GDPR on the data controller must cover the whole life cycle of the data, and data protection principles must be enforced at all times.

This section examines each step of the data life cycle and what measures should be taken in order to implement a GDPR-compliant SafeCloud-based healthcare platform.

### 5.1.1 Collection

Collection is the first processing activity performed on personal data. This step of the data life cycle is particularly important, as the controller is subject to numerous obligations at this stage. Indeed, the controller's obligation to provide information appears either at the precise moment of the collection, or during a short time frame after (Articles 13 and 14 of the GDPR). Collection is the first interaction with the data subject; therefore, it is also the moment to obtain the data subject's consent, which is one of the permitted legal justifications for processing.

#### *5.1.1.1 Data collected from the data subject (Article 13)*

The controller shall, when collecting the data directly from the data subject, provide the required information at the time of collection. In the specific case of the SafeCloud healthcare platform, Article 13 mandates that the data controller (the healthcare organization using the platform) provides the following information at this stage:

- Identity and contact details of the controller;
- Contact details of the controller's data protection officer;
- Purpose of the processing and the legitimating ground that allows such treatment;
- Recipients of the personal data collected;
- Any intention to transfer the data abroad;
- The period for which the personal data is stored, or the criteria used to determine such period;
- The existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability;
- The information that the data subject has the right to withdraw consent when consent is the legitimating ground for processing – in the context of healthcare, the provision of personal data by the subject is a contractual requirement, therefore

withdrawing consent would make the contract void, so the patient should be informed of this consequence;

- The right to lodge a complaint with the supervisory authority;
- The information that the provision of personal data is a contractual requirement.

While using the SafeCloud-based healthcare platform, a healthcare organization (the controller) may provide this information in a consent form written in clear language and preconfigured in the CLINIdATA® application, which is printed when the patient arrives to the organization. Depending on the clinical tests that the patient will perform, the consent form may include different types of information. For instance, if the clinical test is to be performed by a different organization (e.g., external laboratory), the consent form should explicitly state the external laboratory that will receive the patient's data. Additionally, if any kind of analytic processing will be done on the patient's data (e.g., using SQ3), such processing should be described in the consent form, explaining which type of security measures are in place to protect the data, and which third-party organizations will have access to aggregate data.

After being read and signed by the patient (or by the holder of parental responsibility if the patient is a child, according to Article 8), the consent form is scanned and uploaded to CLINIdATA® documents manager and stored in SafeCloud Secure Database. By obtaining such an explicit consent, the healthcare organization also complies with Article 9 that specifies stronger protection for special categories of data such as health data.

#### *5.1.1.2 Data not obtained from the data subject (Article 14)*

When the controller obtains data indirectly, the controller must provide the same information that it would if it collected the data directly from the data subject. The controller must provide this information no later than one month after obtaining the data. In addition, the controller must also inform the data subject about the source of the data.

The controller is not obligated to inform the data subject in four situations described in D2.6, Section 3.1.1.2. This includes when the data subject is already informed, which is exactly the case of healthcare organizations using the SafeCloud-based healthcare platform. As explained in Section 5.1.1.1, if the clinical test will be performed in a different organization (e.g. external laboratory), the consent form must explicitly state the external laboratory that will receive the patient's data. Therefore, if an organization B receives an clinical test request from an organization A regarding a certain patient X, organization B should obtain from organization A the consent form signed by patient X and confirm that it mentions explicitly that personal data will be transferred to organization B. This consent form should be uploaded to CLINIdATA® document manager and stored in SafeCloud Secure Database, because according to Article 5(2) *"The controller shall be responsible for, and be able to demonstrate compliance (…)"*. If it is not possible to obtain a consent form from organization A, then organization B should not proceed with the clinical test without obtaining a signed consent form as explained in Section 5.1.1.1.

### 5.1.2 Introduction and access to personal data in the system

As explained in D2.6, Article 5(1)d states that data shall be *"accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that is inaccurate, having regard to the purposes for which they are processed, is erased or rectified without delay."* Moreover, recital 39 states that the data controller shall take every reasonable step in order to ensure that inaccurate data is rectified or deleted. Although this obligation covers the whole data life cycle, for example, the data subject's right to rectification, introduction of the data into the system is a crucial step because it can lead to errors or further violation of the regulation.

In the context of healthcare, the controller (healthcare organization) must define, within its own organization, who is authorized to introduce which type of data. This element is fundamental, as it offers two advantages. First, it provides a way to monitor the introduction of data in its system, thus improving the controller's ability to find the entry with incomplete or false data. Second, it helps to make sure that only authorized trained people, who know the data protection requirements, introduce data in the system. Indeed, it is necessary to train the authorized staff so that they know what data protection requirements are in place.

When using the SafeCloud-based healthcare platform, the definition of authorized users is based on the OrBAC access control model[21]. This approach allows to limit the circle of healthcare professionals authorized to introduce personal data in the system to what is necessary. For example, the system only allows users with medical competencies to introduce medical data. More details on the OrBAC access control model can be found in D2.6.

Additionally, the system manager or any kind of supervisory authority is able to verify that only authorized users have been able to insert/change/delete/access personal data. This is made possible through the audit trail feature of CLINIdATA®. For each operation on personal data, CLINIdATA® records the following information:

- Which user processed the data?
- What type of processing was it?
- When was the processing done?

### 5.1.3 End of data life

According to Article 5(1)e of the GDPR, personal data shall be *"kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) subject to implementation of the appropriate technical and organizational measures required by this Regulation in order to safeguard the rights and freedoms of the data subject"*. In other words, personal data shall be stored for a period that is strictly limited to the minimum. In order to ensure that the storage period is limited, the controller shall establish time limits for erasure, or for a periodic review (Recital 39 of GDPR).

When using SafeCloud Healthcare Platform, healthcare organizations must configure a retention policy for the different types of personal data. CLINIdATA® allows to define different kinds of alerts so that the healthcare organization is able to eliminate personal data according to what was promised to the patient in the consent form.

## 5.2 GDPR Articles 7, 8, 13 and 30

This section discusses how the SafeCloud-based healthcare platform complies with specific articles of GDPR, following recommendation 5 of SafeCloud project's 2nd year review.

[21] http://orbac.org/

### 5.2.1 GDPR Article 7

Article 7 of the GDPR details four conditions for consent. We discuss each condition below.

*"1. Where processing is based on consent, the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data."*

When using the SafeCloud-based healthcare platform, healthcare organizations (controllers) are able to demonstrate that patients (data subjects) have consented to processing of their personal data, by implementing the procedures described in Section 5.1.1.

*"2. If the data subject's consent is given in the context of a written declaration which also concerns other matters, the request for consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language. Any part of such a declaration which constitutes an infringement of this Regulation shall not be binding."*

The procedures described in Section 5.1.1 make it clear that the consent form is printed from the CLINIdATA® application and it does not concern other matters.

*"3. The data subject shall have the right to withdraw his or her consent at any time. The withdrawal of consent shall not affect the lawfulness of processing based on consent before its withdrawal. Prior to giving consent, the data subject shall be informed thereof. It shall be as easy to withdraw as to give consent."*

As explained in Section 5.1.1.1, in the context of healthcare, the provision of personal data is a contractual requirement, therefore withdrawing consent would make the contract void. Such information must be included in the consent form.

*"4. When assessing whether consent is freely given, utmost account shall be taken of whether, inter alia, the performance of a contract, including the provision of a service, is conditional on consent to the processing of personal data that is not necessary for the performance of that contract."*

As explained in Section 5.1.1.1, the consent form must include the purpose of processing and the legitimating ground that allows such treatment. Only personal data necessary for the stated purpose can be collected from the patient.

### 5.2.2 GDPR Article 8

Article 8 of the GDPR describes the conditions applicable to a child's consent in relation to information society services. The procedures described in Section 5.1.1 make it clear that the consent form must be signed by the patient or by the holder of parental responsibility if the patient is a child according to Article 8.

### 5.2.3 GDPR Article 13

Discussion on the Article 13 of the GDPR is presented in Section 5.1.1.1.

### 5.2.4 GDPR Article 30

Article 30 of the GDPR describes the requirements regarding how processing activities should be recorded. According to Article 30(5) *"The obligations referred to in paragraphs*

*1 and 2 shall not apply to an enterprise or an organisation employing fewer than 250 persons unless the processing it carries out is likely to result in a risk to the rights and freedoms of data subjects, the processing is not occasional, or the processing includes special categories of data as referred to in Article 9(1) or personal data relating to criminal convictions and offences referred to in Article 10.”*, paragraphs 1 and 2 are applicable to healthcare organizations independently of their size because they process health data, which is one of the special categories of data included in Article 9(1).

The recording of processing activities is out of the scope of the SafeCloud-based healthcare platform, but healthcare organizations may attach any kind of document using CLINIdATA® document manager, including for instance an information security policy that details the recording activities. Independently of the technology used, healthcare organizations should record, in writing, all processing activities, including those implemented through the SafeCloud-based healthcare platform (e.g., processing of personal data to perform blood tests).

For each processing activity, healthcare organizations, when acting as **controllers**, should record the following information (Article 30(1)):

- the name and contact details of the controller and, where applicable, the joint controller, the controller's representative and the data protection officer;
- the purposes of the processing;
- a description of the categories of data subjects and of the categories of personal data;
- the categories of recipients to whom the personal data have been or will be disclosed including recipients in third countries or international organizations;
- where applicable, transfers of personal data to a third country or an international organization, including the identification of that third country or international organization and, in the case of transfers referred to in the second subparagraph of Article 49(1), the documentation of suitable safeguards;
- where possible, the envisaged time limits for erasure of the different categories of data;
- where possible, a general description of the technical and organisational security measures referred to in Article 32(1).

Regarding the last bullet above, D2.6 Section 4 describes the technical security measures of the SafeCloud-based healthcare platform to protect integrity and confidentiality.

For each processing activity, healthcare organizations, when acting as **processors**, must record the following information (Article 30(2)):

- the name and contact details of the processor or processors and of each controller on behalf of which the processor is acting, and, where applicable, of the controller's or the processor's representative, and the data protection officer;
- the categories of processing carried out on behalf of each controller;
- where applicable, transfers of personal data to a third country or an international organization, including the identification of that third country or international organization and, in the case of transfers referred to in the second subparagraph of Article 49(1), the documentation of suitable safeguards;

- where possible, a general description of the technical and organisational security measures referred to in Article 32(1).

Again, regarding the last bullet above, D2.6 Section 4 describes the technical security measures of the SafeCloud-based healthcare platform to protect integrity and confidentiality.

Finally, according to Article 30(4), *"The controller or the processor and, where applicable, the controller's or the processor's representative, shall make the record available to the supervisory authority on request."* This means that processing activities should be recorded in a way that they can be quickly obtained.

# 6 Conclusion

This document described the final deployment of SafeCloud Healthcare Platform. The deliverable presented:

- An overview of the CLINIdATA® eHealth Solution.
- A description of the final deployment and integration of CLINIdATA® with the Secure Queries solutions. This deployment covers the 3 scenarios presented in D5.2: SaaS, hybrid and analytics deployments.
- Performance tests used to evaluate SafeCloud Healthcare Platform.
- Pilot of SafeCloud Healthcare Platform, which exposed the CLINIdATA® + SQ1 integration to critique by healthcare professionals.
- A set of demonstration procedures that show the implementation of the use case described in Deliverable D5.2 in practice.
- A discussion of the compliance of SafeCloud Healthcare Platform with European Union's (EU) GDPR.

# 7 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