
# D3.16

# Libraries for consent management and decentralized authorization mechanisms for HR Exchange - V1



ABSTRACT

This deliverable describes a demonstration of the initial version of the security libraries offered by the InteropEHRate Framework as a reference implementation of the consent management in mobile and web applications for health record exchange. It also outlines for these libraries their description regarding their current version and the used licences, as well as specific information regarding their development, followed by specific guidelines regarding the installation and the usage of these libraries, through a detailed guide. This software will be compliant to the specification released by the task on "*Cross-border consent management, decentralized authorization mechanisms and block chain*" [1] .

| | |
|---|---|
| **Delivery Date** | January 20th, 2020 |
| **Work Package** | WP3 |
| **Task** | T3.4 |
| **Dissemination Level** | Public |
| **Type of Deliverable** | Demonstrator |
| **Lead partner** | UBITECH LIMITED [UBIT] |

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

CONTRIBUTORS

| | Name | Partner |
|---|---|---|
| **Contributors** | Sofianna Menesidou<br>Dimitris Papamartzivanos | UBIT |

LOGTABLE

| Version | Date | Change | Author | Partner |
|---|---|---|---|---|
| 0.1 | 05-12-19 | First draft of ToC | D. Papamartzivanos<br>S. Menesidou | UBIT |
| 0.2 | 11-12-19 | Introduction | D. Papamartzivanos | UBIT |
| 0.3 | 14-12-19 | SW Description | S. Menesidou | UBIT |
| 0.4 | 16-12-19 | SW Description | S. Menesidou | UBIT |
| 0.5 | 19-12-19 | SW Description | S. Menesidou | UBIT |
| 0.6 | 29-12-19 | Overview | D. Papamartzivanos, S. Menesidou | UBIT |
| 0.7 | 17-01-20 | Overview | D. Papamartzivanos, S. Menesidou | UBIT |
| 0.8 | 17-01-20 | Structure | D. Papamartzivanos | UBIT |
| 0.9 | 19-01-20 | Quality Review | Argyro Mavrogiorgou | UPRC |
| vFinal | 20-01-20 | Final review and version for submission | Laura Pucci | ENG |

## ACRONYMS

| Acronym | Term and definition |
|---|---|
| D2D | Device-to-Device |
| R2D | Remote-to-Device |
| HCP | Healthcare Professional |
| HR | Health Record |
| S-EHR | Smart Electronic Health Record |
| SW | Software |

TABLE OF CONTENT



LIST OF TABLES

# 1. INTRODUCTION

## 1.1. Scope of the document

The main goal of this document is to deliver a demonstration of the initial version of the security libraries offered by the InteropEHRate Framework as a reference implementation for consent management. The decentralized authorization will be provided in the second version of this deliverable. In more detail, the current document outlines for these libraries their description regarding their current version and the used licences, as well as specific information regarding their development (e.g. programming languages, supported platforms, etc). Moreover, this document includes specific guidelines regarding the installation and the usage of these libraries, through a detailed guide.

It should be noted that this document is a software (SW) report, a pointer to the actual deliverable, regarding the design of the libraries for HR security and privacy services **[D3.9]**. In addition, the security libraries are closely related and dependent to the relying exchange protocol and for that reason, the same code base provided in **[D.12]** is extended to support the necessary security features for consent management.

## 1.2. Intended audience

The document is intended to security engineers, developers, architects, and all the InteropEHRate project participants and partners interested to have an overview of how InteropEHRate will support HR security and privacy services. Apart from that, the document is intended for researchers and developers as well, as they may be interested in installing and using the designed security libraries.

## 1.3. Structure of the document

The current document is organized in the following Sections:

- **Section 1** introduces the overall concept of the document, defining its scope, intended audience, and relation to the other project tasks and reports.
- **Section 2** outlines the description of the software regarding the offered security libraries, including details such as their programming languages, and their supported platforms.
- **Section 3** describes for each library an installation guide, as well as a user guide.

## 1.4. Updates with respect to previous version (if any)

Since this document is the initial version of the demonstration of the security libraries for the consent management and decentralized authorization, no update with regards to previous related documents can be reported.

# 2. SW DESCRIPTION

## 2.1. S-EHR side Security Library

| | |
|---|---|
| **SW TITLE** | Mobile D2D Security Management (M-D2D-SM) |
| **SW VERSION** | 1.0 |
| **LICENCES AND PATENTS** | Apache License |
| **PROGRAMMING LANGUAGES** | Java SE 8.0 |
| **SUPPORTED PLATFORM(s)** | Android (5.0 - Current Version) |
| **SOURCE CODE** | http://iehrgitlab.ds.unipi.gr/interopehrate/s-ehr-mobile-app/d2d-hr-exchange/tree/security |
| **EXECUTABLE** | N.A. |

*Table 1 – S-EHR security library description*

## 2.2. HCP side Security Library

| | |
|---|---|
| **SW TITLE** | Terminal D2D Security Management (T-D2D-SM) |
| **SW VERSION** | 1.0 |
| **LICENCES AND PATENTS** | Apache License |
| **PROGRAMMING LANGUAGES** | Java SE 8.0 |
| **SUPPORTED PLATFORM(s)** | Windows OS |
| **SOURCE CODE** | http://iehrgitlab.ds.unipi.gr/interopehrate/reference-hcp-app/terminal-d2d-hr-exchange/tree/security |
| **EXECUTABLE** | N.A. |

*Table 2 – HCP side security library description*

# 3. OVERVIEW

## 3.1. S-HER side Security Library

The current release of the mobile security library (i.e. M-D2D-SM), contains all the operations needed from the side of the S-EHR application developer to interact with the D2D library and the reference S-EHR application. This library contains different operations that have to be invoked for implementing the security functions of the D2D protocol in the context of the mobile application. This library is a Java-based component that can be nested in any Android application. It offers a set of Java operations for consent management. More details regarding these operations can be found in deliverables **[D3.7]** and **[D3.9]**.

### 3.1.1. Installation guide

The installation guide of the security D2D library inside the S-EHR app contains the necessary information for the process that has to be followed in order to add the library to the S-EHR app Android project, and make sure that every component of the S-EHR app is able to use the security library. As already mentioned, the security library extends the same code base of the D2D protocol due to some necessary dependencies. Hence, the installation guide introduced in **[D4.12]** is the same for the security library. For completeness purposes we provided it also in this deliverable. This guide is used to install the version of the library released in December 2019, and may be no longer valid for more recent versions. The only requirement for this process is for the developer to have an Android project application running with the min-SDK version properties upper or equal to 15.

**Installation steps of the S-EHR side security D2D library:**

1. Insert into the build.gradle file of the project the following lines of code inside the repository section to retrieve libraries:

```
repositories {
    google()
    jcenter()
    maven {
   url 'http://213.249.46.206:8081/repository/maven-releases/'
   content {
     includeGroup 'eu.interoperhate'
     }
   }
}
```

2. Insert into the build.gradle file of the module the following dependency:

```
implementation(group:'eu.interoperhate',name:'md2de',version: '0.0.1')
```

3. Refresh the gradle project.

### 3.1.2. User guide

This user guide refers to the version 1.0 (V1) of the consent management functionalities of the S-EHR side as an extension of the D2D library. At this stage we assume both parties had already exchanged the certificates during the identification steps. The main flow of the functionality of this library is the following:

The functions `onConsentRequested ()`, `verifySignature ()`, and `signPayload ()` are called automatically upon receiving the consent from HCP, in order to validate the signature of the HCP, to double-sing the approved Consent and forward it back.

```
public void onConsentRequested(Consent consent, String signedConsent);

public boolean verifySignature(RSAPublicKey publicKey, byte[] scannedAddress,
byte[] scannedSignature) throws UnsupportedEncodingException,
NoSuchAlgorithmException, InvalidKeyException, SignatureException;

public String signPayload(String payload, PrivateKey privateKey) throws
IOException, SignatureException, InvalidKeyException, NoSuchAlgorithmException,
InvalidKeySpecException;
```

## 3.2. HCP side Security Library

The current release of the library contains all the operations that are needed from the side of the HCP application developer to interact with the D2D library and the reference HCP application. This library contains different operations that have to be invoked in a specific sequence for implementing the security functions of the D2D protocol. This library is a Java-based component that can be embedded in any Java application. It offers a set of Java methods for consent management. More details regarding these operations can be found in deliverables **[D3.7]** and **[D3.9]**.

### 3.2.1. Installation guide

The installation guide of the security D2D library inside the HCP contains necessary information for the process that has to be followed in order to add the library to the HCP Java project and make sure that every component of the HCP is able to use the security library. As already mentioned, the security library extends the same code base of the D2D protocol due to some necessary dependencies. Hence the installation guide introduced in **[D4.12]** is the same for the security library. For the sake of completeness, we provided it also in this deliverable. This guide is used to install the version of the library released in December 2019 and may be no longer valid for more recent versions. The only requirement for this process is for the developer to have Git and a Java 8 JDK installed. This guide is used to install the version of the library released in December 2019, and may be no longer valid for more recent versions.

**Installation steps of the HCP side security D2D library:**

1. Access the library's Java Project repository that is located on GitLab EHR private repository and clone the Terminal D2D HR Exchange.
2. Change to the security branch of the HCP app.
   http://iehrgitlab.ds.unipi.gr/interopehrate/reference-hcp-app/terminal-d2d-hr-exchange/tree/security
3. Open the cloned project using the Maven opener menu of the desired IDE.
4. Build the project by generating the .jar file and ensure that is located in the Maven repository of libraries of the computer that the HCP App project will be located.
5. Include the necessary dependencies for the D2D library jar file inside the HCP App project.

### 3.2.2. User guide

This user guide refers to the version 1.0 (V1) of the consent management functionalities of the HCP side as an extension of the D2D library. At this stage we assume both parties had already exchanged the certificates during the identification steps. The main flow of the functionality of this library is the following:

- The `createConsent()` function will be called to create the Consent which will be sent to the citizen in order to get accepted and signed.

```
public Consent createConsent(ConsentState theState, Patient thePatient,
Practitioner practitioner, Date date, List<Reference> organazation, Narrative
purpose, List<CodeableConcept> cagegory);
```

- The `signPayload()` function will be called to sign the Consent using the private key of the healthcare professional.

```
public String signPayload(String payload, PrivateKey privateKey)
        throws IOException, SignatureException, InvalidKeyException,
NoSuchAlgorithmException, InvalidKeySpecException
```

- The `getSigneConsent()` function will be called to send the Consent and request the approval of the S-HER user with her digital signature. The function sends the following Strings separated by "#" encoded in Base64 format:
  `"ConsentDetailsDocument#"+encoded+"#"+signature`

```
public void getSignedConsent(Patient patient) throws Exception;
```

- The `onConsentAnswerReceived()` function will be called immediately after the signed Consent by the S-EHR App user. The function `verifySignature()` is then used for signature verification and storing the consent.

```
public void onConsentAnswerReceived(String consentAnswer);

public boolean verifySignature(RSAPublicKey publicKey, byte[] payload, byte[]
signature)throws UnsupportedEncodingException, NoSuchAlgorithmException,
InvalidKeyException, SignatureException;
```

# REFERENCES


[1] **[D3.7]** Specification of consent management and decentralized authorization mechanisms for HR Exchange - V1, 2019. https://www.interopehrate.eu/resources/#dels

[2] **[D3.9]** InteropEHRate Consortium, Design of libraries for HR security and privacy services - V1, 2019. https://www.interopehrate.eu/resources/#dels

[3] **[D4.12]** InteropEHRate Consortium, Libraries for remote and D2D HR exchange - V1, 2019. https://www.interopehrate.eu/resources/#dels