



## D4.17

# Libraries for research health data sharing - V1

### ABSTRACT

This deliverable accompanies the demonstrator software components of the research data sharing libraries designed and developed in the InteropEHRate project. It provides an overview of the (first version of the) demonstrator implementation, including a list of functionalities demonstrated, building, installation, and configuration instructions, as well as a user manual.

Delivery Date	21 <sup>st</sup> April, 2021
Work Package	WP4
Task	T4.5
Dissemination Level	Public
Type of Deliverable	Demonstrator
Lead partner	UNITN



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.



This work by Parties of the InteropEHRate Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

DRAFT

## CONTRIBUTORS

	Name	Partner
Contributors	Gábor Bella	UNITN
Contributors	Simone Bocca	UNITN
Contributors	Niklas Haldorn	FRAU
Contributors	Sofianna Menesidou	UBITECH
Contributors	Martin Marot	A7

## LOGTABLE

Version	Date	Change	Author	Partner
0.1	17-03-2021	Document and first draft created	Gábor Bella	UNITN
0.2	02-04-2021	Partner contributions integrated	Martin Marot Sofianna Menesidou Niklas Haldorn	A7 UBITECH FRAU
0.3	09-04-2021	UNITN server-side contributions	Simone Bocca	UNITN
0.9	13-04-2021	Document prepared for submission	Gábor Bella	UNITN
1.0	19-04-2021	Quality check	Argyro Mavrogiorgou	UPRC
Vfinal	19-04-2021	Final check and submission	Laura Pucci	ENG

## ACRONYMS

Acronym	Term and definition
NPM	NodeJS Package Manager: a widely-used package manager application.
PI	Principal Investigator: the researcher in charge of a research study.
RDD	Research Definition Document: a formal, computer-processable description of a research study.
RDS	Research Data Sharing: secure IT communication protocol (and APIs) for publishing and retrieving machine processable descriptions of research studies and for sending citizen's consents and health data from S-EHR Apps to research centres (that are RDS nodes), without any cloud storage of health records. The RDS Protocol has not to be confused with a Research Protocol.
RRC	Reference Research Centre: a research centre participating in a given study is a RRC for the citizens who are officially attached to it (i.e. send data to it) for the duration of the study.
S-EHR App	The app running on Citizens' mobile devices that stores, displays, and shares smart health data with other healthcare services.

## TABLE OF CONTENT

1. INTRODUCTION .....	6
1.1. Scope of the document .....	6
1.2. Intended audience.....	6
1.3. Structure of the document.....	6
1.4. Updates with respect to previous version (if any) .....	6
2. OVERVIEW .....	7
3. SOFTWARE DESCRIPTION .....	8
3.1. Central Node Software Components.....	8
3.2. S-EHR App Software Components.....	10
4. INSTALLATION AND USER GUIDES.....	11
4.1. Installation and Configuration Guide .....	11
4.1.1. Central Node Demo Installation and Configuration .....	11
4.1.2. S-EHR App Demo Installation and Configuration .....	13
4.2. User Guide .....	14
4.2.1. Central Node Demo User Guide .....	14
4.2.2. S-EHR App Demo User Guide.....	17
4.3. Other Related Documentation .....	19

## LIST OF FIGURES

Figure 1 – Central Node login screen

Figure 2 – Central Node RDD upload interface

Figure 3 – Central Node RDD detail display interface

Figure 4 – Central Node RDD publishing interface

Figure 5 – S-EHR App interfaces, in order: (1...

## LIST OF TABLES

Table 1 - Central Node

Table 2 – RRC/CN RDS security library description

Table 3 – RDD Validator library description

Table 4 – S-EHR App executable description

Table 5 – S-EHR-App-side RDS security library description

DRAFT

# 1. INTRODUCTION

## 1.1. Scope of the document

This deliverable describes the software components for the Research Data Sharing Demonstrator. The goal of the demonstrator is to show how the Research Data Sharing Protocol [D4.8] and the corresponding libraries [D4.10] can be used to implement an actual working data sharing infrastructure. Research data sharing, as designed in the InteropEHRate project, consists of managing the participation of citizens in research studies, as well as automatically, securely, and anonymously transmitting personal health data from citizens' mobile devices to selected research centres. The software components described in this deliverable implement a first, in-progress version of the demonstrator that focuses on the handling of *Research Definition Documents*, both on the Central Node (server) of the Research Network and on the S-EHR App. The demonstrator executables are currently available on the internal InteropEHRate GitLab code repository. Once finished, they will be released to the public.

This deliverable describes the demonstrator in its first, usable but so far incomplete version, as described in sections 2 and 4. The second deliverable version will present the demo in its entirety.

## 1.2. Intended audience

This deliverable is intended for:

- system developers who wish to study an example implementation of research data sharing;
- healthcare professionals and data scientists involved in medical research and the corresponding data preparation operations, who wish to see a demonstration on how research data sharing from mobile devices can be implemented.

## 1.3. Structure of the document

Section 2 provides an overview of the functionalities covered by the demonstrator. Section 3 provides concise descriptions of the demonstrator software components. Section 4 provides detailed installation, configuration, and usage guides.

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

Not applicable, this is the first version.

## 2. OVERVIEW

In its final form, the demonstrator will implement the interaction between three systems and three kinds of human actors:

- the **S-EHR App** operated by a **Citizen** sharing his/her data for research;
- the **Central Node** of the Research Network where **Researchers** (typically Principal Investigators) upload *Research Definition Documents* (RDD) and an **Administrator** formally accepts and publishes these, and from which the S-EHR App downloads these RDDs;
- the **Reference Research Centre** chosen by the Citizen to which the S-EHR App sends health data and consent information.

In the present first version of the demo, only the S-EHR App and the Central Node are included. The Reference Research Centre will be added in the second version. This first release implements the initial stages of the research data sharing process. With respect to the protocol steps described in [D4.8] and [D4.10], this demonstrator version covers:

### 1. PUBLISHING:

- a new research study is uploaded by the Principal Investigator of the study on the Research Platform of the Central Node of the Research Network;
- the study is validated both automatically and manually by the Central Node Administrator;
- the study is digitally signed and published on the Central Node.

### 2. OPT-IN:

- a Citizen decides to opt in for solicitations to participate in future research studies.

### 3. ENROLLMENT (only the initial part of this step so far):

- the S-EHR App of the Citizen automatically downloads new research studies from the Central Node, including the one just published, and checks their digital signatures;
- the eligibility of the Citizen is automatically verified with respect to the studies;
- the Citizen is solicited to participate in a research study, providing him/her information on the study;
- the Citizen chooses a Reference Research Centre to which he/she will be attached for that particular study;
- digital signature of consent by the Citizen: will be covered in v2;*
- sending of consent to the Reference Research Centre chosen: will be covered in v2;*
- the Reference Research Centre acknowledges and counter-signs the consent, and sends back contract signed by both parties to the Citizen: will be covered in v2;*
- the Citizen's mobile device receives the contract and stores it: will be covered in v2.*

### 4. DATA RETRIEVAL: will be covered in v2.

### 5. WITHDRAWAL: will be covered in v2.

**OPT-OUT:** the Citizen decides not to participate in future studies.



## 3. SOFTWARE DESCRIPTION

### 3.1. Central Node Software Components

The Central Node is a component of the InteropEHRate Research Network, defined to allow the publication of new studies, represented by the Research Definition Document (RDD), on the Research Network. Thanks to that, the studies can be retrieved by those citizens who want to participate, through the S-EHR Application installed in their own smartphone. The Central Node is implemented as a web portal that can be accessed by two kind of actors, namely:

- the Principal Investigator (PI) of a Study: the researcher who defined the study's RDD. The PI, having an account registered in the Central Node, can upload new RDD that will be later approved (if no problems occur) and finally published on the Research Network.
- the Central Node Administrator: the Central Node user who has special rights to check, publish and eventually delete each study uploaded on the Central Node. The administrator is in charge of checking and deciding if a study is valid, and so publishes the study itself on the Research Network, or rejects the study due to inconsistencies. The administrator can eventually withdraw a study from the Research Network (Unpublish operation) if necessary.

In order to support the operations performed by the two authors described above, the Central Node component provides, through dedicated UI, the functionalities for:

- login/logout to PI or administrator account;
- upload a new RDD. This functionalities include an automatic validation of the RDD to be uploaded;
- publish/unpublish a study, if the user is the Central Node administrator;
- delete a study from the Central Node (so also from the Research Network), if the user is the Central Node administrator;

Moreover, the Central Node component provides, in backend, a RESTful service which allows the S-EHR Application to download all the research studies published and currently open. The Table below shows the general information about the Central Node component.

The **Central Node Web Portal** provides the UI and back-end logic for the management of RDDs.

SW TITLE	
SW TITLE	Research Network Central Node Web Portal
SW VERSION	1.0
LICENCES AND PATENTS	Open source (TBD)
PROGRAMMING LANGUAGES	JavaScript, NodeJs
SUPPORTED PLATFORM(s)	Windows OS Unix/Linux OS

SOURCE CODE	<a href="http://iehrgitlab.ds.unipi.gr/interopehrate/research-services/irs-server-research-data-sharing/tree/master/Central%20Node">http://iehrgitlab.ds.unipi.gr/interopehrate/research-services/irs-server-research-data-sharing/tree/master/Central%20Node</a>
EXECUTABLE	N.A.

*Table 1 - Central Node*

The **Terminal D2D Security Management** library provides secure communication facilities (digital signature, encryption) to the server-side systems of the Central Node and the Research Centre Information System.

SW TITLE	Terminal D2D Security Management (T-RDS-SM)
SW VERSION	1.0
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Java SE 8.0
SUPPORTED PLATFORM(s)	Windows OS Unix/Linux OS
SOURCE CODE	N.A.
EXECUTABLE	<a href="http://213.249.46.206:8081/#browse/browse/components:maven-public:e5c50d09b73fd3c5fd9fb4a00791de8d">http://213.249.46.206:8081/#browse/browse/components:maven-public:e5c50d09b73fd3c5fd9fb4a00791de8d</a>

*Table 2 – RRC/CN RDS security library description*

The **RDD Validator** library provides syntactic validation functionalities for RDD files.

SW TITLE	RDD Validator Library
SW VERSION	0.1
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Java SE 8.0
SUPPORTED PLATFORM(s)	Windows OS Unix/Linux OS
SOURCE CODE	N.A.
EXECUTABLE	<a href="http://iehrgitlab.ds.unipi.gr/interopehrate/research-services/irs-server-research-data-sharing/tree/master/validator">http://iehrgitlab.ds.unipi.gr/interopehrate/research-services/irs-server-research-data-sharing/tree/master/validator</a>

*Table 3 – RDD Validator library description*

### 3.2. S-EHR App Software Components

The **S-EHR App** is the main executable app running on the Citizen's mobile device, aggregating all citizen-centred functionalities.

SW TITLE	S-EHR App
SW VERSION	0.3.0
LICENCES AND PATENTS	Proprietary
PROGRAMMING LANGUAGES	Java SE 8.0 & kotlin 1.3
SUPPORTED PLATFORM(s)	Android 4.1 - Current version
SOURCE CODE	<a href="http://iehrgitlab.ds.unipi.gr/interopehrate/s-ehr-mobile-app/s-ehr-app">http://iehrgitlab.ds.unipi.gr/interopehrate/s-ehr-mobile-app/s-ehr-app</a>
EXECUTABLE	
SW TITLE	S-EHR App

*Table 4 – S-EHR App executable description*

The **Mobile RDS Security Management** library provides secure communication facilities (digital signature, encryption) to the S-EHR App.

SW TITLE	Mobile RDS Security Management (M-RDS-SM)
SW VERSION	1.0
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Java SE 8.0
SUPPORTED PLATFORM(s)	Android 5.0
SOURCE CODE	N.A.
EXECUTABLE	<a href="http://213.249.46.206:8081/#browse/browse/components:maven-public:e5c50d09b73fd3c531e8c40e4a115835">http://213.249.46.206:8081/#browse/browse/components:maven-public:e5c50d09b73fd3c531e8c40e4a115835</a>

*Table 5 – S-EHR-App-side RDS security library description*

## 4. INSTALLATION AND USER GUIDES

### 4.1. Installation and Configuration Guide

#### 4.1.1. Central Node Demo Installation and Configuration

##### Central Node Web Portal

Before to start the Central Node server which allows the access to the web portal functionalities, a configuration phase is needed. Such a configuration includes the set up of the underlying database, used by the Central Node server to manage account information as well as details regarding the RDDs uploaded. The Central Node database, created using PostgreSQL in the demonstrator, maintains two tables, described as follows:

- *account*: used to store information related to the Central Node users. The information maintained for each element stored in the table are:
  - *user\_id*: unique identifier of the user within the the table;
  - *username*: username specified by the user when the account has been created;
  - *password*: password specified by the user when the account has been created;
- *rdd*: used to store information related to the RDDs uploaded. The information maintained for each element stored in the table are:
  - *rdd\_id*: unique identifier of the user within the the table;
  - *pi*: name of the principal investigator of the study;
  - *path*: local path specifying where the study's RDD is stored;
  - *title*: study's title;
  - *created\_at*: upload date of the study in the Central Node;
  - *start\_date*: start date of the study;
  - *end\_date*: end date of the study;
  - *published*: boolean value indicating whether the study is published on the research network or not.

Moreover, some configurations are required on the Central Node server component. To configure the Central Node component the *config.js* file (located in the Central Node installation directory) has to be edited, specifying in the indicated fields (JSON structure):

- the information regarding the PostgreSQL database connection: these mandatory information,, are used by the Central Node component to interact with the underlying database (described above) containing the information about the Central Node users accounts, as well as about all the RDD uploaded. The database connection information are respectively, the host name (*client.host*), the database's host port number (*client.port*), the database name (*client.database*), the database user (*client.user*) and the database password (*client.password*).
- The information about the Central Node web portal endpoint: this data indicates the web portal host (*endpoint.host*) and port number (*endpoint.port*) used to access the Central Node web portal.

The Central Node component is developed in JavaScript Node.js. Due to that, the installation is done using the NodeJs Package Manager, *npm*, running the following command, in the Central Node installation directory, where the *package.json* file is present (used to specify the NodeJs running endpoints):

```
#npm install
```

After the execution of the command above, all the requested dependencies will be correctly installed in the running environment. In order to run the Central Node component, and so allowing the usage of the web portal functionalities as well as the backend RESTful service, it's necessary to run (always in the Central Node installation directory) the following command:

```
#node index.js
```

The command above will run the Central Node server that will allow access to the web portal through a generic browser specifying the host IP address and the port defined in the configuration file. Once the server is running also the RESTful service to download the published open studies will be available and waiting for requests.

### RDD Validator Library

The *IGInstaller* library contains an archived version of the *npm* package for the Implementation Guide, that can be used for validation. The *IGInstaller* class can be used to install this package and overwrites existing versions and the following table provides a description of the exposed operation.

The *installRDSIG* operation installs the *npm* package of the library in the package cache. This overwrites the existing version of the package and should not be used if the currently installed version is more recent than the library version.

### Server-Side Security Library

In order to install the server-side security library (T-RDS-SM), for the convenience of deployment, it can be uploaded onto Nexus as follows.

1. For the first time, in Eclipse, go to `${user.home}/.m2/settings.xml` and add the following lines of code:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository />
  <interactiveMode />
  <usePluginRegistry />
  <offline />
  <pluginGroups />
  <servers>
    <server>
      <id>nexus</id>
      <username>USERNAME</username>
      <password>PASS</password>
    </server>
  </servers>
  <mirrors />
  <proxies />
  <profiles />
  <activeProfiles />
</settings>
```

2. Go to `pom.xml` and change the version of the library.
3. For the first time right click on `pom.xml` Run us -> Run configurations. Right click on Maven build ->

New Configuration. Select the base directory and in field 'Goals' write 'package' and press Apply.

4. Go again to Maven build -> New Configuration. Select the base directory and in field 'Goals' write 'clean deploy' and press Apply.
5. Choose the two configurations and press Run.

Then, the Installation steps for the security library are as follows.

1. Access the library's Java Project repository that is located on GitLab EHR private repository and clone the corresponding repository.
2. Change the branch of the project to the desired one that corresponds to the latest version.
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 it is located in the local Maven repository.
5. Include the necessary dependencies for the RDS security library jar file inside the project.

Note that further detailed instructions with respect to the installation of security libraries can be found in the deliverable [\[D3.14\]](#).

#### 4.1.2.S-EHR App Demo Installation and Configuration

In order to run the demo on the S-EHR App side, first the S-EHR App (such as the Andaman7 app) itself needs to be installed, in the same way as in other InteropEHRate demos. Please follow the instructions described in the [\[D6.10\]](#) "S-EHR mobile app - V2" in section 3.1 "Installation guide". Likewise, in order to install the S-EHR-App-side security library (M-RDS-SM), please follow the instructions described in the [\[D3.14\]](#).

##### Uploading the S-EHR-App-side security library on Nexus

1. For the first time, in Android Studio, go to File -> Settings -> Tools -> Terminal and change the 'Shell path' to the directory of installed 'git bash'.
2. Go to build.gradle (Module: M-RDS-SM) file and change the version of the library.
3. Change the Build Variants from 'debug' to 'release'.
4. In the toolbar select Build -> Make module 'M-RDS-SM'.
5. Open the terminal of Android Studio and execute the command './gradlew:M-RDS-SM:publish'.
6. Return the Build Variants from 'release' to 'debug'.

##### S-EHR-App-side security 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: 'M-RDS-SM', version: '0.0.2')
```

3. Refresh the gradle project.

## 4.2. User Guide

Beyond the deployment of software components, a full demonstration also needs one or more RDD files to be uploaded onto the Central Node and to be subsequently downloaded and interpreted by the S-EHR App.

The demonstrator should be executed in the same order as it is described in the following sections. In the first version, the following steps are demonstrated.

1. Central Node demo, consisting of:
  - a. login by a Researcher (PI of a study);
  - b. uploading and validation one or more RDDs;
  - c. logout and login by the Central Node Administrator;
  - d. publishing of one or more RDDs.
2. S-EHR App demo, consisting of:
  - a. opt-in by the Citizen;
  - b. silent download of RDDs from the Central Node and silent checking of enrollment criteria;
  - c. solicitation of the Citizen for participation in a study;
  - d. formal acceptance of participation and selection of a Reference Research Centre;
  - e. silent sharing of data with the Reference Research Centre.

### 4.2.1. Central Node Demo User Guide

This section provides a list of all the functionalities provided by the Central Node component. It includes both the functionalities accessed by the web portal UI, and those provided as RESTful services.

Accessing, through a generic browser, to the Central Node web portal (using host address and port number specified during the configuration), the user can perform the following actions:

- Login as PI or Central Node administrator. The following figure shows the UI offered to the user to perform such an operation.

# Research Network Central Node

Please log in as:

- Principal Investigator in order to submit a new study
- Central Node Administrator in order to manage studies

Username:

Password:

Log In

*Figure 1 – Central Node login screen*

- Once the user is logged in, he/she can logout from her/his account using the button on the top right of the page, as shown in the figure below.
- The PI can upload a new study, selecting the relative RDD from his/her local filesystem, and clicking the *Upload* button. The validation check will be automatically performed on the RDD selected, verifying if the document can be correctly uploaded or it contains structural errors.

Hi, Simone Bocca

Log Out

## Upload a new Research Study

Sfoglia...

Nessun file selezionato.

Upload

*Figure 2 – Central Node RDD upload interface*

- Once a new RDD has been uploaded, it is shown in the center of the current page. The general information about each study uploaded, reported in this page, specify the title of the study, the start and end date of the study and a red or green square indicating the publication status of the relative study. Clicking on the title of the study, the PI can explore the details of the study, accessing the following page.



Hi, Simone Bocca Log Out

**Profile Information**

**Username :** Simone Bocca

**General information**

<b>Title:</b>	COVID zero day (3)
<b>Principal Investigator:</b>	Dr. Stefano Dalmiani
<b>Description:</b>	Body temperature of people in different environments and seasons
<b>Start date:</b>	January 1st, 2020
<b>End date:</b>	December 31st, 2021
<b>Enrollment start date:</b>	January 1st, 2020
<b>Enrollment end date:</b>	July 1st, 2021

Data requirements

Enrollment criteria

Research centres

Figure 3 – Central Node RDD detail display interface

- If the user logged in is the Central Node Administrator, then the first page will be the one shown by the following figure. Here the administrator has the possibility to publish, unpublish or completely delete each study previously uploaded by any PI, registered on the Central Node. Of course, in order to check and analyze the study, also the administrator can explore the study's details by clicking on the title of the study.

Hi, admin Log Out

**Administrator Profile Information**

**Username :** admin

**List of studies**




	P.I. OF THE STUDY	TITLE	BEGINNING OF THE STUDY	END OF THE STUDY	PUBLISHED
publish / unpublish delete	Simone Bocca	COVID zero day (2)	February 1st, 2020	November 30th, 2020	
publish / unpublish delete	Simone Bocca	COVID zero day (1)	January 1st, 2020	December 31st, 2021	
publish / unpublish delete	Simone Bocca	COVID zero day (3)	January 1st, 2020	December 31st, 2021	

Figure 4 – Central Node RDD publishing interface

The Central Node RESTful service, called *open-studies*, allows the S-EHR Application to download all the open studies that have been published by the Central Node administrator. This service can be called by the mobile device that has the S-EHR App correctly installed and configured, through the following REST call:

```
http://<host>:<port>/open-studies -H "Content-Type: application/json"
```

where, <host> is the Central Node host, specified in the Central Node configuration file, and <port> is the host's port number which exposes the Central Node web service (also specified in the configuration file).

#### 4.2.2.S-EHR App Demo User Guide

On the mobile side, in the first version of the demo, the Citizen will have access to different functionalities related to the research.

1. Opt in into the InteropEHRate open research network. Being part of this network means that the citizen allows the S-EHR App silently to retrieve available studies from the Central Node, and to check if the citizen's data match the enrolment criteria of the study.
2. Opt out from the InteropEHRate open research network.
3. Consult the list of received invitations to research studies (only those that match the enrolment criteria).
4. Consult the details of a study.

The screenshots below correspond to the UI actions described above.

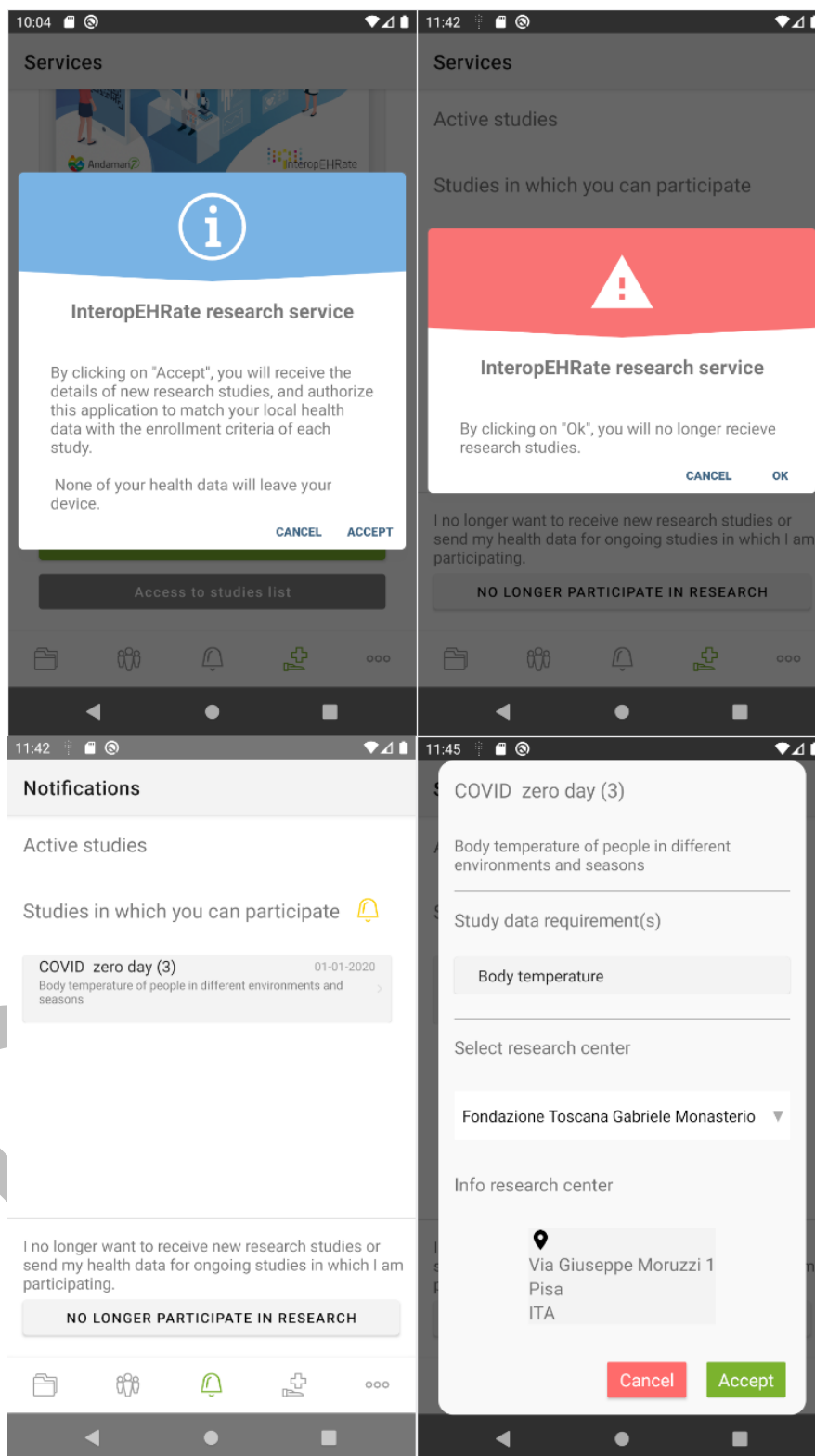


Figure 5 – S-EHR App interfaces, in order: (1) opt-in, (2) opt-out, (3) visualization of studies available to the Citizen, (4) display and selection of study details as well as the interface for selecting the Citizen's Reference Research Centre

Further description of these features can be found in the deliverable [D6.10] "S-EHR mobile app - V2" in section 3.2.5 "Scenario 3 : Research".

### 4.3. Other Related Documentation

For the Research Data Sharing demonstrator, four examples of RDDs have been produced in order to test the protocol features such as RDD publishing and match of enrollment criteria to verify the citizen involvement in the study. The four RDD examples, defined as JSON files stored within the InteropEHRate [GitLab repository](#), are defined as different versions of the same research study. The study considers the temperature measurements on citizens, during the COVID-19 pandemic period. The four versions of the studies are described below:

- *ErrorRDDBundle.json*: this example of RDD is wrong defined, in particular a mandatory attribute (Following the RDD FHIR InteropEHRate profile) is missing. Due to that this file will cause error when it is uploaded on the Central Node. This file is used to test the RDD validation library used in the Central Node server.
- *ExpiredRDDBundle.json*: this example represents an RDD that is not valid anymore. More in details the end date (as well as the enrollment end date) is antecedent respect to the demonstrator presentation date.
- *MatchRDDBundle.json*: this example represents a valid RDD (not expired), and is defined in order to be downloaded by a S-EHR app maintaining citizen's health data that match the enrollment criteria specified by within the RDD.
- *NoMatchRDDBundle.json*: in a complementary way, this example is a valid RDD defined to test the case in which the citizen's data do not match the enrollment criteria specified by within the RDD.

## REFERENCES

- [1] **[D3.1]** InteropEHRate Consortium, D3.1-Specification of S-EHR mobile privacy and security conformance levels - V1, 2020. <http://www.interopehrate.eu/resources/#dels>
- [2] **[D3.14]** InteropEHRate Consortium, D3.14-Libraries for encrypted storage and communication in mobile and web applications - V1, 2021. <http://www.interopehrate.eu/resources/#dels>
- [3] **[D4.8]** InteropEHRate Consortium, D4.8-Specification of protocol and APIs for research health data sharing - V1, 2021. <http://www.interopehrate.eu/resources/#dels>
- [4] **[D4.10]** InteropEHRate Consortium, D4.10-Design of library for health data sharing for research - V1, 2021. <http://www.interopehrate.eu/resources/#dels>
- [5] **[D6.10]** InteropEHRate Consortium, D6.10-S-EHR mobile app - V2, 2020. <http://www.interopehrate.eu/resources/#dels>

DRAFT