



RESILOC			
<i>Resilient Europe and Societies by Innovating Local Communities</i>			
Grant Agreement No		833671	
Start date	01.06.2019	End date	30.11.2022

## D4.5 – Sensor based solutions

Due date of deliverable: 31/07/2022

Submission date: 01/08/2022

JSI, Slovenia

Revision	Organization & Person	Date
Written by	JSI	28/07/2022
Checked and approved by	NKUA, Vassilis Papataxiarhis	01/08/2022
Validated and released by	FhG, Karsten Uhing	01/08/2022



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833671

## I. Deliverable Information

<b>Deliverable Number</b>	D4.5
<b>Work Package</b>	WP4
<b>Date of Issue</b>	01/08/2022
<b>Version Number</b>	1.2
<b>Nature of Deliverable</b>	Other
<b>Dissemination Level (PU / RE / CO)</b>	PU

<b>Author(s)</b>	Andrej Hrovat (JSI) Miha Mohorčič (JSI) Mihael Mohorčič (JSI)
<b>Keywords</b>	statistical data, wireless gateways, BLE beacons, smartphone mobile APP, survey platform, WiFi signaling traffic monitoring, Android APP, iOS APP, database, raw data, detecting WiFi interfaces, clustering algorithm

### Abstract

As of type OTHER, the deliverable D4.5 consists of hardware and software components of the statistical data collection system, described more in detail and put in the broader context of the RESILOC platform & tools and their use in the deliverable D4.6. This brief report is only accompanying D4.5 hardware and software deliverables, providing a brief reference guide for their installation and deployment. After introducing the concept of the system, the individual hardware and software components are described. In particular, a subsystem of fixed deployed devices is briefly described, which includes BLE beacons for asset tracking and proximity detection, wireless gateways for capturing WiFi signaling traffic and detecting BLE beacons, the installation guidelines and add-on for WiFi signaling traffic collection. Next, the implementation and features of purposely developed mobile APPs for Android and iOS operating systems are described, followed by the description of the survey platform with the RESILOC-specific customizations. Finally, a brief introduction to the database for storing raw data collected by mobile APP, and wireless gateways is provided, followed by a more detailed description of the solution for identifying the number of individual active WiFi interfaces in the area at a given time. The document includes all the links to public repositories where the source codes of the implemented solutions are available, as well as the URL links to the software (survey platform deployment, App Store, Google Play, etc.).

### Disclosure Statement:

The text, figures and tables in this report can be reused under a provision of the Creative Commons Attribution 4.0 International License. Logos and other trademarks are not covered by this license. The content of the documents marked as restricted or confidential are not to be disclosed externally without prior written consent from the RESILOC Consortium, that can be requested via [resiloc-dpo@fraunhofer.de](mailto:resiloc-dpo@fraunhofer.de). The content of the publication herein is the sole

responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services. While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the RESILOC consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neither the RESILOC Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein. Without derogating from the generality of the foregoing neither the RESILOC Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

## II. Document History

Date	Version	Modified by (first name, name, organization)	Remarks
10/06/2022	0.1	Andrej Hrovat, JSI	Table of Contents
05/07/2022	0.2	Andrej Hrovat, JSI	Chapter 1, 2 ,3, 5
15/07/2022	0.3	Miha Mohorčič, JSI	Chapter 1,2,3, 4, 5
26/07/2022	1.0	Andrej Hrovat, JSI	Chapter 1, 2, 3, 4, 5
28/07/2022	1.1	Mihael Mohorčič, JSI	Finalization
01/08/2022	1.2	Andrej Hrovat and Mihael Mohorčič, JSI	Revised version as per QA comments

### III. Table of Contents

I.	Deliverable Information .....	i
II.	Document History .....	iii
III.	Table of Contents .....	iv
IV.	List of Figures .....	v
V.	List of Acronyms .....	vi
1	Introduction .....	1
1.1	Statistical data collection system .....	1
1.2	Components of the statistical data collection system .....	2
2	Fixed deployed devices .....	3
2.1	Wireless gateway .....	3
2.1.1	Wireless gateway setup .....	4
2.1.2	WiFi signaling traffic capturing .....	5
2.2	BLE beacons .....	6
3	Mobile application .....	8
3.1	Android application .....	8
3.2	iOS application .....	10
4	Survey platform .....	12
4.1	Deployment .....	12
4.2	RESILOC specific customization .....	12
5	Statistical analysis tool and database .....	14
5.1	Database for storing raw data .....	14
5.2	Detecting unique WiFi interfaces .....	14
5.2.1	Data collection .....	14
5.2.2	Data pre-processing .....	15
5.2.3	Algorithm for distinguishing individual WiFi capable devices .....	16
5.2.4	Algorithm implementation availability .....	19
6	Conclusions .....	20
7	References .....	21

## IV. List of Figures

Figure 1: Statistical data collection system scheme .....	1
Figure 2: Deployment scheme of fixed devices.....	3
Figure 3: Wireless gateway in a waterproof enclosure .....	4
Figure 4: Wireless gateway WiFi setup screen .....	4
Figure 5: Wireless gateway LAN setup screen.....	5
Figure 6: System design and WiFi data flow .....	6
Figure 7: BLE tags and a user interface for tag reprogramming by a mobile application.....	7
Figure 8: Example of screenshots in different languages .....	8
Figure 9: Notification of the questionnaire on the smartphone home screen and a link in the mobile APP .....	9
Figure 10: Questionnaire interface .....	9
Figure 11: Mobile APP settings screen.....	10
Figure 12: iOS application interface.....	11
Figure 13: RESILOC survey home screen.....	12
Figure 14: RESILOC survey location notification plugin .....	13
Figure 15: LimeSurvey login screen .....	13
Figure 16 Example of data in DB gathered from a smartphone .....	14
Figure 17: Saving new probe request.....	16
Figure 18: Grouping probe requests to the groups of global and random devices.....	17
Figure 19: Reachability plot.....	18
Figure 20: Clustering based on the reachability plot.....	19

## V. List of Acronyms

Acronym	Meaning
AP	Access Point
API	Application Programming Interface
APP	Application
BLE	Bluetooth Low Energy
CID	Company ID
CPU	Central Processing Unit
GSM	Global System for Mobile Communications
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IE	Information Elements
JSI	Jozef Stefan Institute
JSON	JavaScript Object Notation
LAN	Local Area Network
LRT	Local Resilience Team
LTE	Long-Term Evolution
MAC	Media Access Control
MQTT	MQ Telemetry Transport
OPTICS	Ordering Points to Identify the Clustering Structure
OS	Operating System
PR	Probe Request
REST	REpresentational State Transfer
rPi	RaspberryPi
RSSI	Received Signal Strength Indicator
SD	Secure Digital
SSID	Service Set IDentifier
TOA	Time Of Arrival
USB	Universal Serial Bus
VPN	Virtual Private Network
WiFi	Wireless Fidelity

# 1 Introduction

Obtaining dynamic data in a given environment and time frame is important to understand the impact of specific events and to enable appropriate strategic planning of resilience measures because it provides an insight into past events and possible behavior of people. Therefore, we have developed an approach for obtaining statistical data that actively engages the population through purposely developed mobile application and filling the surveys. The entire solution is designed to be relatively unobtrusive, meaning that a large amount of data for further processing by experts is obtained without the active participation of the population. In addition to storing data from purposely deployed BLE (Bluetooth Low Energy) [1] beacons, the developed system also enables the unobtrusive and privacy preserving collection of data that is transmitted in any case via WiFi [2] interfaces built into smartphones and other WiFi enabled personal devices.

## 1.1 Statistical data collection system

The system for automatic collection of the statistical data is schematically depicted in Figure 1. It consists of wireless gateways (built on RaspberryPi (rPi) devices [3], hence rPi gateways), BLE beacons, smartphone mobile applications (mobile APP for Android and iOS operating systems), a survey platform integrated within the RESILOC platform and the database containing data collected by the field deployed devices and the smartphone applications. Data from the fixed deployed devices and mobile APPs are additionally pre-processed according to the use cases and made available to experts (LRTs) for further use. Results from the questionnaire are available to the expert in a survey add-on for further use in strategy planning.

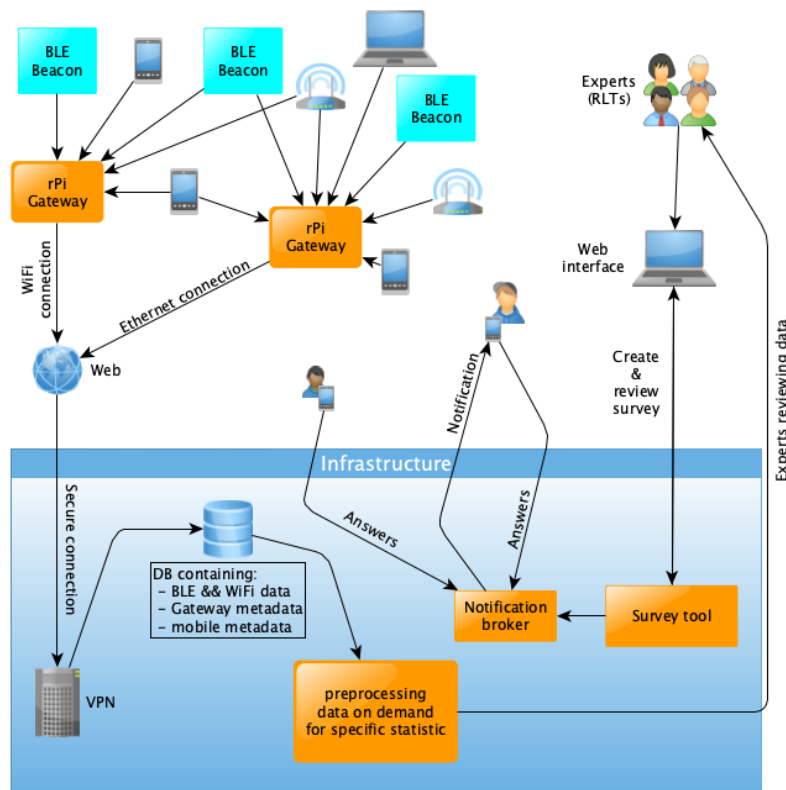


Figure 1: Statistical data collection system scheme



## 1.2 Components of the statistical data collection system

Components of the statistical data collection system that constitute deliverable D4.5, briefly described in the following sections, include:

- Wireless gateway
  - o Raspberry Pi  
<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
  - o USB WiFi dongle  
[https://www.edimax.com/edimax/merchandise/merchandise\\_detail/data/edimax/au/wireless\\_adapters\\_ac600\\_dual-band/ew-7811utc](https://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/au/wireless_adapters_ac600_dual-band/ew-7811utc)
  - o wireless gateway setup software  
[https://gitlab.com/e62Lab/resiloc\\_project/public/gateway](https://gitlab.com/e62Lab/resiloc_project/public/gateway)
  - o WiFi signaling traffic capturing software:  
[https://gitlab.com/e62Lab/resiloc\\_project/public/gateway](https://gitlab.com/e62Lab/resiloc_project/public/gateway)
- BLE beacon
  - o Minew i9 Coin Tag  
<https://www.minewstore.com/product/i9-coin-tag/>
  - o BLE tag reprogramming software:  
<https://play.google.com/store/apps/details?id=com.minnw.beaconset>
- Mobile application
  - o source code: [https://gitlab.com/e62Lab/resiloc\\_project/public/mobile](https://gitlab.com/e62Lab/resiloc_project/public/mobile)
  - o Android download:  
<https://play.google.com/store/apps/details?id=si.ijs.e6.resilocApp>
  - o iOS download:  
<https://apps.apple.com/us/app/resiloc/id1636471068>
- Survey platform
  - o source code: [https://gitlab.com/e62Lab/resiloc\\_project/public/survey](https://gitlab.com/e62Lab/resiloc_project/public/survey)
  - o RESILOC deployment: <https://resiloc10.di.uoa.gr/>
- Algorithm for detecting unique WiFi interfaces
  - o source code:  
[https://gitlab.com/e62Lab/resiloc\\_project/public/wireless-data-analysis](https://gitlab.com/e62Lab/resiloc_project/public/wireless-data-analysis)

## 2 Fixed deployed devices

Leveraging existing technologies, a low-cost system consisting of a wireless gateway (rPi) and BLE beacons was developed to collect additional data on population behavior in terms of occupancy, density, presence, proximity, track mobile/portable dangerous assets, tag the assets, etc. in predefined areas of interest.

System deployment scheme is shown in Figure 2. The wireless gateway, installed at a predefined location, passively monitors and collects WiFi signaling traffic (no interaction with population or their devices) and detects active BLE beacons within the range (purposely deployed beacons, activated BLE beacons in smartphones). The wireless gateway filters and parses the data, performs some data aggregation and sends it to the database in JSON [4] format over a broadband connection (LAN/WiFi/4G) using the MQTT [5] protocol.

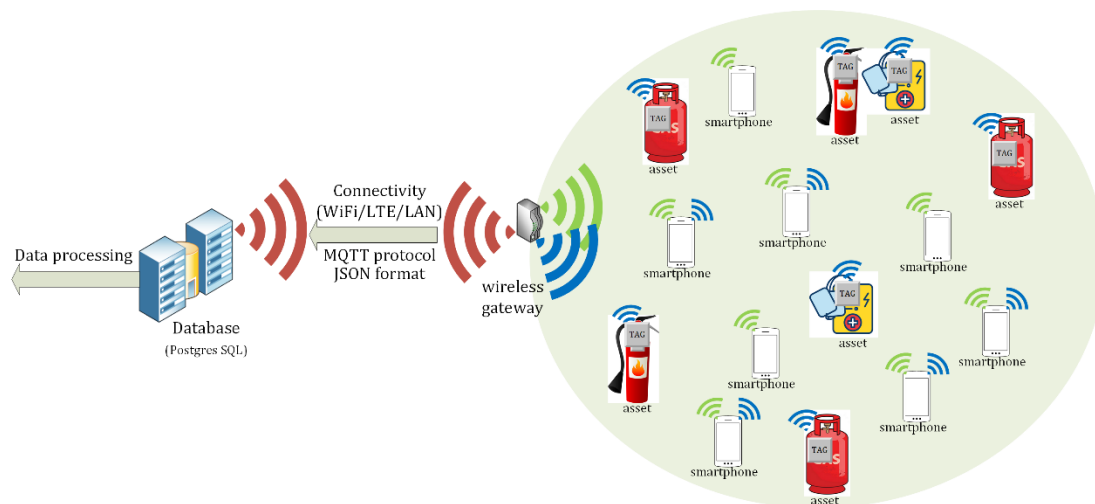


Figure 2: Deployment scheme of fixed devices

### 2.1 Wireless gateway

Due to its relatively low cost and known capabilities, the rPi device (rPi 4 with 2 GB of internal memory) was selected to be used as a wireless gateway. Since the rPi's built-in WiFi adapter and drivers are not capable of capturing data when in the *monitor* mode of operation, it was complemented by an additional USB WiFi dongle (using Realtek rtl88212au chipset), using compiled drivers that enable the *monitor* mode. The rPi built-in adapter is used by default as a connection to the Internet, which is needed to transfer data to the database. In addition to capturing WiFi discovery packets (constituting signaling traffic), the gateway in the *monitor* mode also captures BLE beacon advertisements and RESILOC mobile APP beacons (if enabled).

The rPi with the additional WiFi dongle is installed in a waterproof enclosure suitable for outdoor installation, as shown in Figure 3. In addition to the WiFi-based internet connection, a connection via LAN is also possible. For each rPi, an SD card is pre-programmed and installed with the operating system and the required supported services and applications.

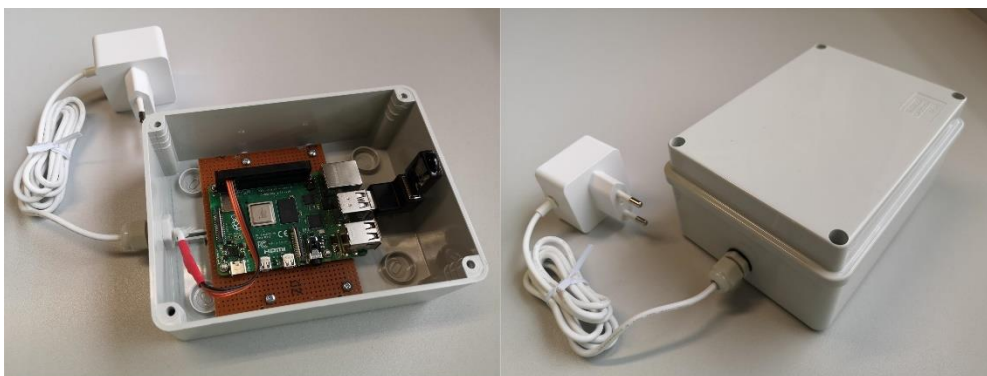


Figure 3: Wireless gateway in a waterproof enclosure

### 2.1.1 Wireless gateway setup

The initial setup of the device must be performed before it is put into operation. During the installation, a phone or a laptop is needed to set the connectivity and location options.

When the device is turned on for the first time, it enters into the *host* mode and creates a WiFi access point (AP) named "**Resiloc AP 1.3**". Using a laptop or a smartphone, the user connects to the WiFi AP "Resiloc AP 1.3" by entering the password "**APforResilocConfig**", opens the browser, enters the address **10.0.0.1** and the screen shown in Figure 4 is displayed. Due to the limited capabilities of the rPi, it may take a few seconds to display the page. If the timeout error occurs, the page must be reloaded. The browser displays a warning that the connection is not secure because the certificate used is self-signed and correct time may not be set on the gateway without the Internet connection. Next, the user selects the WiFi network for the Internet access, enters the credentials and optionally a location name of the installed gateway. If the phone is used to set up the gateway, it must be allowed in the browser to tag the location via the phone (longitude/latitude). The installation is completed by clicking the "Submit Query" button. After this, the device switches to the *client* mode and its WiFi AP is no longer visible. If the password is incorrect or a connection to the network cannot be established for other reasons, it switches back to the *host* mode after one minute and the WiFi AP is visible again. If it stays connected for longer than 2 minutes, the network is saved. Thus, even if the connection is interrupted at some point, the device will attempt to reconnect as soon as the saved network is available again.

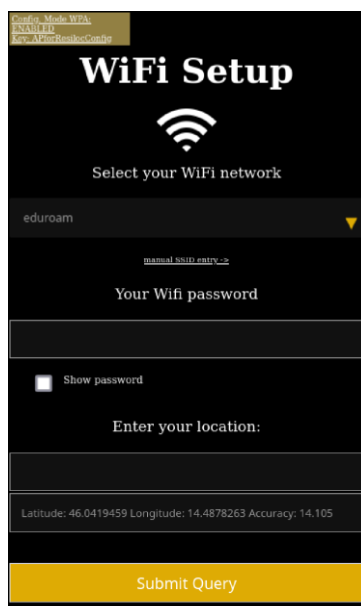


Figure 4: Wireless gateway WiFi setup screen

If the device is connected via a wired LAN connection, it still needs to be configured using the procedure described above. The website looks slightly different, as shown in Figure 5. Additional text is displayed on the screen: "Connected via Ethernet, optionally enter a backup WiFi and hit submit", and the WiFi drop-down menu contains the option "Don't connect to a WiFi".

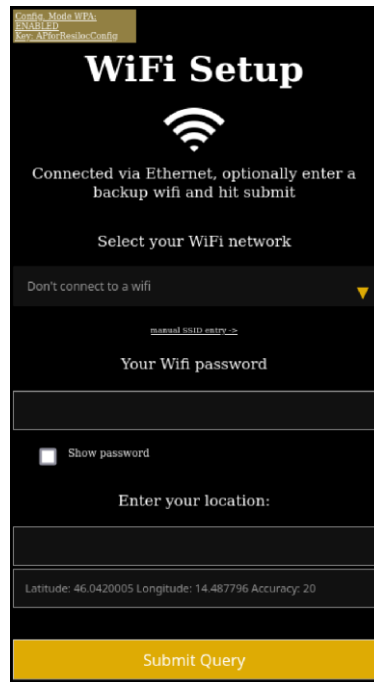


Figure 5: Wireless gateway LAN setup screen

The last step of the setup procedure is to verify the connection. When a new connection is established, the AP is no longer visible. If it does not become visible again within a minute and a half, the connection has been successfully established. Otherwise, the connection process must be started again from the beginning.

The source code and documentation for this part of the gateway is available in the "AP" folder of the repository available at [https://gitlab.com/e62Lab/resiloc\\_project/public/gateway](https://gitlab.com/e62Lab/resiloc_project/public/gateway).

### 2.1.2 WiFi signaling traffic capturing

Passive signaling traffic monitoring is performed by listening to probe request packets in a 2.4 GHz WiFi band. However, since the WiFi implementation on devices are being privacy conscious by standard, the WiFi MAC addresses of the devices sending the requests are randomized. This prevents MAC addresses from being used as unique identifiers for trivial tracking of an individual device or a user and in the context of the RESILOC project also guarantees high level of anonymization.

On the monitoring device, certain minimum computing power is required for preprocessing, filtering, and transmitting the collected data. This dictates the minimum capabilities of CPU and connectivity options (besides an adapter that can monitor WiFi) and limits the software stack used. Once the data is stored in the database, it can be further analyzed either in real time or on demand.

The operating system used for the node is a minimal installation of rPi OS (64bit). An additional driver module is compiled and installed for interfacing the RTL8812au chip. The monitored data is collected using the Tshark [6] packet capture program, which is filtered in the next steps and parsed in Python using the PyShark package. The collected data is transferred to the PostgreSQL database in JSON format via Rest API [7] calls to the remote server.

With respect to privacy and security, the collected public data is already anonymized by the design (random MAC addresses, changing properties) of the probe requests (PRs). The parsed data is transmitted to the database via HTTPS rest calls, ensuring that the sensitive data is not exposed at any point in the communication pipeline. In addition, all traffic from the node is routed over a secure VPN connection. This ensures that traffic to the database and communication with the node (e.g., for upgrades, backups...) is also encrypted when running over the public Internet. Simplified system design and data flow can be found in Figure 6.

The source code for the capture and transmission of the data on the gateway is available in the folder “gw\_service” of the repository available at [https://gitlab.com/e62Lab/resiloc\\_project/public/gateway](https://gitlab.com/e62Lab/resiloc_project/public/gateway).

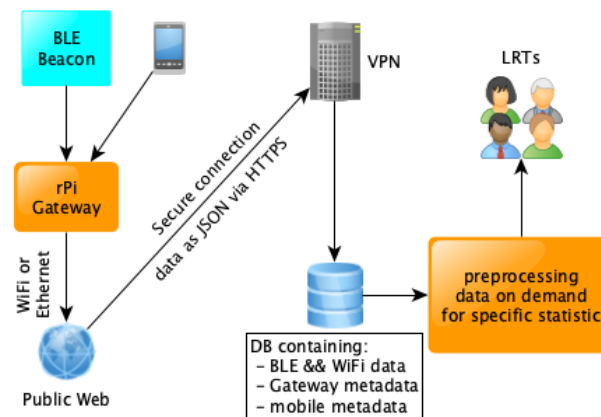


Figure 6: System design and WiFi data flow

## 2.2 BLE beacons

For the BLE beacons, we selected passive off-the-shelf devices from Minew[8] (left image in Figure 7). The tags are of the size of a coin and can be easily attached to an asset or some other surface of interest with double-sided adhesive tape. They are battery powered and are activated by pulling out the plastic tab that blocks the button battery. The operating time depends on the frequency of data advertising and can last from a few months to a year.

The tags support BLE 5.0 and are primarily used for asset tracking and indoor location-based services. They can advertise the MAC address, unique ID, battery status, asset location, comments, etc., and can be reprogrammed via a computer or a mobile application that can be downloaded from the Google Play repository at <https://play.google.com/store/apps/details?id=com.minnwb.beaconset>. A screenshot of the application is shown on the right in Figure 7.

BLE tags are scanned using both wireless gateways and the RESILOC mobile application developed for Android operating systems. Similar to WiFi data, information about BLE beacons is stored in the database, where it is available for further processing according to the requirements of the use case.

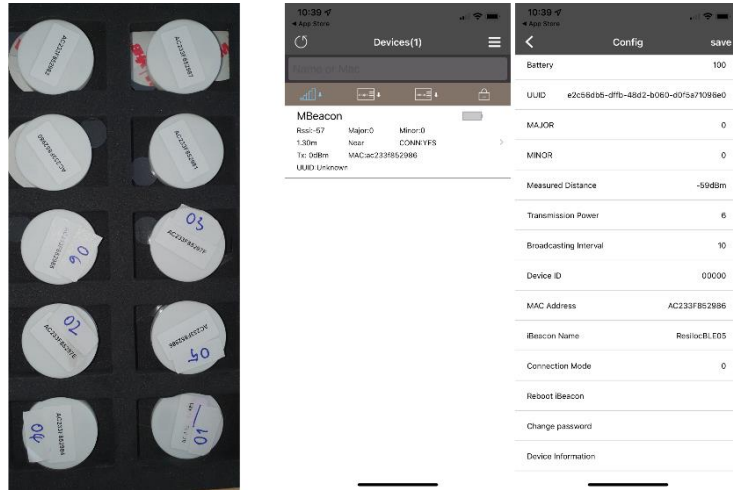


Figure 7: BLE tags and a user interface for tag reprogramming by a mobile application

### 3 Mobile application

Mobile applications for the Android and iOS operating systems have been developed and published on the Google Play and App Store repositories, respectively. They support several functionalities:

- collection of information from LRTs through questionnaires,
- collection of information from the "community", i.e. citizens, through questionnaires,
- displaying various notifications,
- collection of data through built-in radio interfaces - only supported by the version for Android smartphones.

Both versions of the application are easy to install and are managed through a simple user interface. They require basic user permissions when first launched, and do not affect the smartphone's performance. The application runs in the background, it does not collect, store or publish any of the user's personal data, and is completely unobtrusive with only a notification bar visible to the user.

The source code for both applications is available in their respective folders in the repository at [https://gitlab.com/e62Lab/resiloc\\_project/public/mobile](https://gitlab.com/e62Lab/resiloc_project/public/mobile).

#### 3.1 Android application

The Android version of the application consists of two screens: the notification screen and the user settings screen.

To address local language issues the automatic language selection is enabled based on mobile phone settings. The support for Italian, Bulgarian and Greek has been added (Figure 8) in addition to English. If the local language is not supported, the application uses the default language - English.

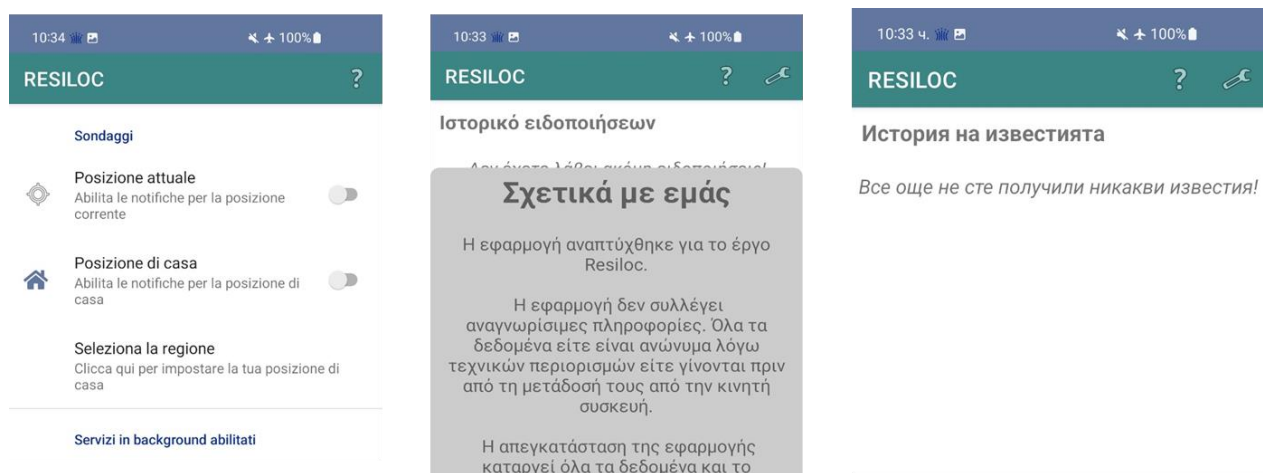


Figure 8: Example of screenshots in different languages

Questionnaires created in the RESILOC survey platform can be sent to the users of mobile APP. The questionnaire screen with the surveys to be answered and tray notification of sending is shown in Figure 9. When receiving a new questionnaire, a user is notified about it through the notification bar (left side of Figure 9). The user receives the notifications depending on the setting in the application (Figure 11). On the settings screen, the user can allow receiving questionnaires for his home location and/or for the current approximate location. Please note that the home location must be set by the user in the settings menu ("Select region" button).



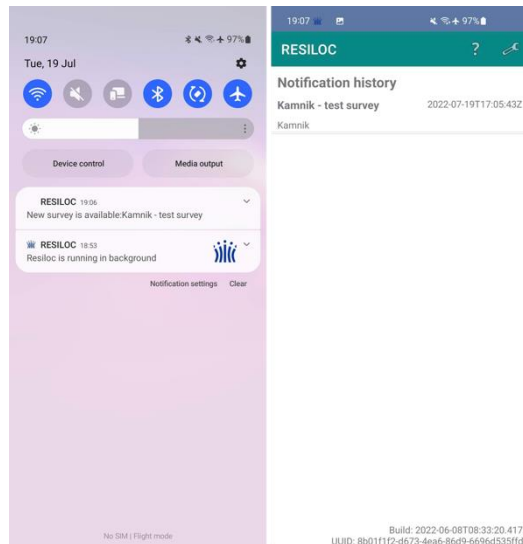


Figure 9: Notification of the questionnaire on the smartphone home screen and a link in the mobile APP

A click/tap on an unanswered questionnaire opens the default web browser for providing an answer to avoid possible display issues on various mobile devices. For an example of a screenshot, see Figure 10.

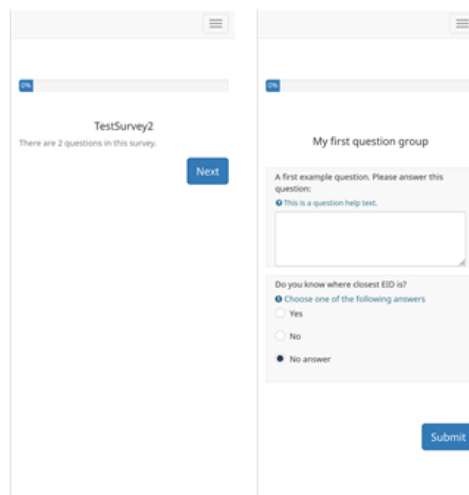


Figure 10: Questionnaire interface

Additional features of the mobile APP, developed for the Android OS, include collecting data from built-in radio interfaces and geolocation data. The background services enabled in the settings screen of the mobile APP periodically send the information about how many and which radio interfaces are seen by a particular smartphone (BLE, WiFi, GSM/LTE). In addition, the smartphone can also announce its own BLE beacon and collect geolocation information. To this end, before the first use the user's consent is required by confirming the pop-up windows. The user of the application should also specify in the user settings whether they are an LRT in their home location.



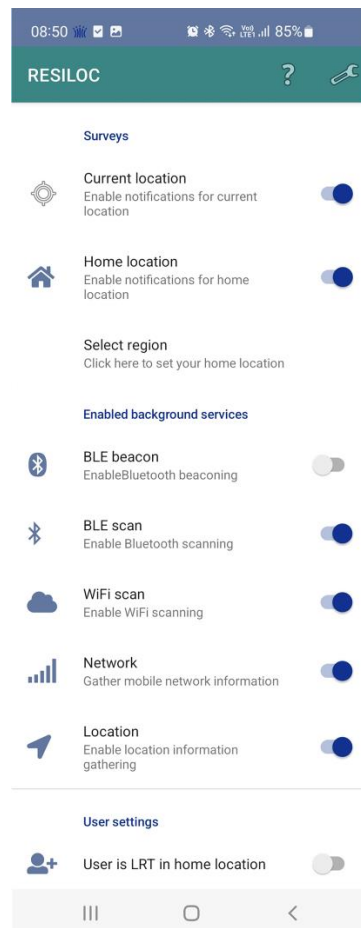


Figure 11: Mobile APP settings screen

The various background services are available via the settings screen, as shown in Figure 11:

- BLE beacon - the smartphone periodically sends BLE beacons that are detected by other smartphones running the RESILOC mobile APP or by fixed deployed RESILOC wireless gateways (rPi gateways);
- BLE scan - scans for BLE beacons transmitted by devices within the range of the smartphone;
- WiFi scan - scans WiFi interfaces within the range of the smartphone;
- Network - provides information about available mobile network cells (of the selected operator);
- Location - provides information about the current geographical location.

The collected data is sent to the database at predefined time intervals for further processing according to the defined use cases and made available to the experts.

The application is available for download from the Google Play at:

<https://play.google.com/store/apps/details?id=si.ijs.e6.resilocApp>

### 3.2 iOS application

The mobile application developed for the iOS operating system has limited functionality compared to the Android application. While the Android mobile APP also supports the collection of data via built-in radio interfaces, the iOS application only provides notification and questionnaire functionalities which are described in details in previous Section. However, unlike the Android mobile APP, it allows receiving questionnaire notifications only for the home

location and opens the respective questionnaire in the default web browser. The screenshots of the application can be seen in Figure 12.

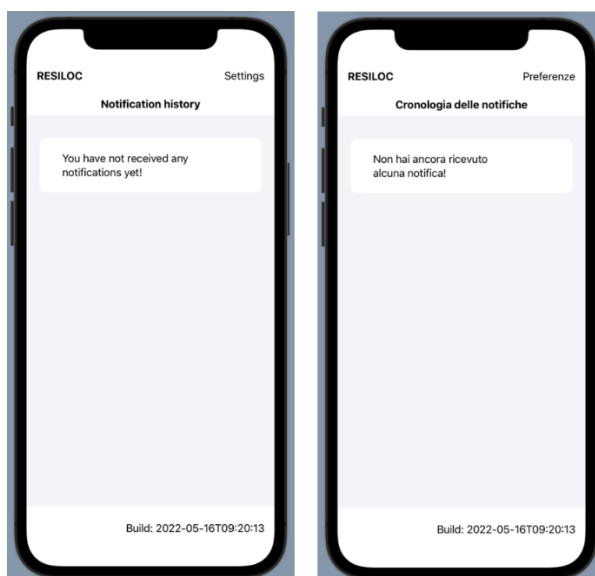


Figure 12: iOS application interface

The application is available for download from the App Store at:  
<https://apps.apple.com/us/app/resiloc/id1636471068>.

## 4 Survey platform

The survey platform leverages the existing survey tool called LimeSurvey [9] (<https://github.com/LimeSurvey/LimeSurvey>). It is an open source tool that natively supports internationalization, multiple question types, user roles, and the ability to extend functionality via plugins or extensions.

The source code for the deployment and plugins is available in the “survey” folder of the repository available at [https://gitlab.com/e62Lab/resiloc\\_project/public/survey](https://gitlab.com/e62Lab/resiloc_project/public/survey).

### 4.1 Deployment

The base platform is deployed on a virtual machine at <https://www.di.uoa.gr/> and is accessible via the domain name <https://resiloc10.di.uoa.gr/> (Figure 13).

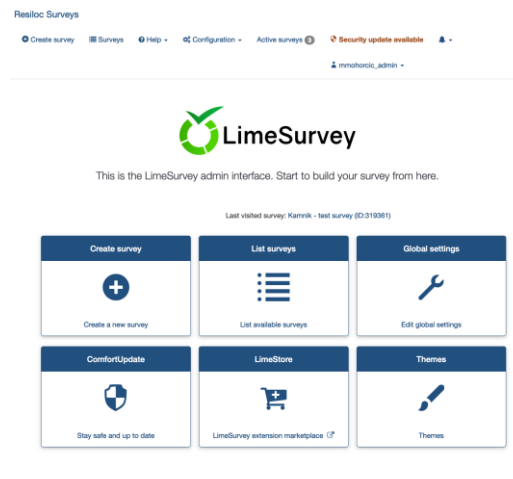


Figure 13: RESILOC survey home screen

The system requirements for running the platform are not demanding, but the virtual machine should be extended in case of additional work (e.g. data analysis directly on the host). The deployment uses 2 virtual cores with 2 GB RAM and 20 GB of disk space. It runs on a base instance of Ubuntu server 20.04 and the web stack consists of Apache 2.4.41, PHP 7.4.3 and MariaDB 15.1. HTTPS certificates are automatically managed via Letsencrypt.

### 4.2 RESILOC specific customization

After the basic installation of LimeSurvey, it is then customized with the RESILOC notification plugin and authentication extension.

The notification plugin is installed in a default plugin directory. After installation, it must be enabled by a system administrator of the platform. After that, the user creating a survey has the ability to send notifications to selected locations (Figure 14) via the MQTT protocol to the mobile devices or subscribers of the notifications. The notifications are stored in a database and can be retrieved later if the application user or the administrator wants to view the history of the notifications.

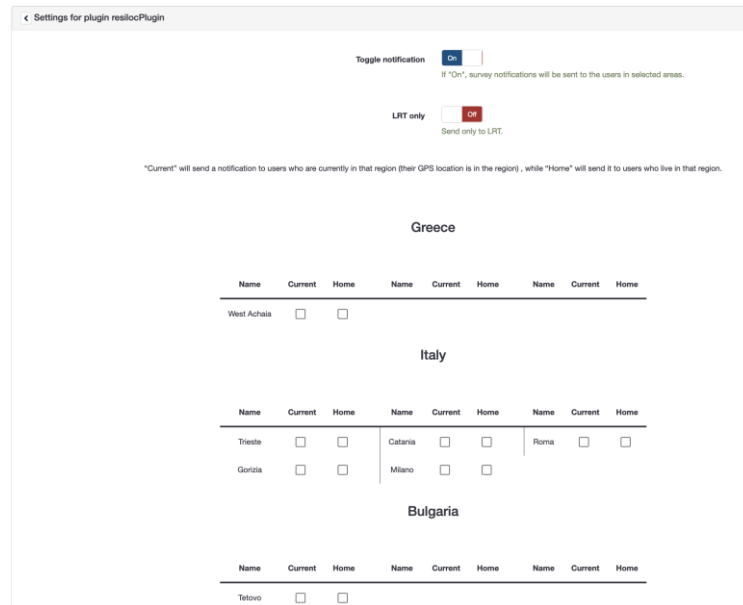


Figure 14: RESILOC survey location notification plugin

The authentication extension is installed in the root folder. It interacts directly with the database and verifies the validity of the user session by exchanging tokens between the RESILOC platform and the survey deployment. The extension also optionally allows programmatic addition and modification of user settings.

In case of problems with single sign-on exchange between the RESILOC platform and the survey tool, users can still log in with a valid password via <https://resiloc10.di.uoa.gr/admin>, as shown in Figure 15.

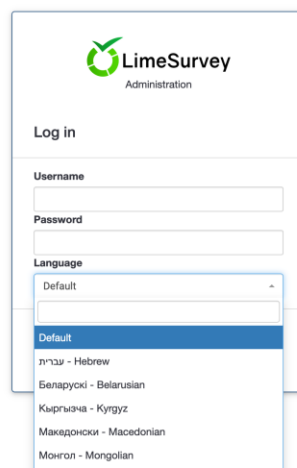


Figure 15: LimeSurvey login screen

## 5 Statistical analysis tool and database

### 5.1 Database for storing raw data

The database used to store raw data collected from the fixed deployed devices (rPi gateways) and by the Android mobile APP, is an instance of the PostgreSQL [10] database running on the JSI infrastructure. The collected raw data is stored with metadata in a JSON format. An example of such packages in the database can be seen in Figure 16. The data packages contain only a single instance of a scan with anonymized data points. The raw data must be processed according to the predefined requirements of the use case. Preprocessed data is stored in a separate data stream and is available to the expert for further use.

```
1 {
2   {
3     "dataStreamId": 38,
4     "payload": {
5       "type": "wifi",
6       "scanned_device": {
7         "2c:9f:fb:cb:65:5c": -82,
8         "be:52:71:70:d4:a8": -82
9       },
10      "start_timestamp": 1643583591,
11      "end_timestamp": 1643583601,
12      "device": "Latitude: 46.0455936 Longitude: 14.4867328 Accuracy: 1146.6127349598546\n",
13      "serial": "000000063a2d9d9"
14    },
15    "timestamp": "2022-01-31 00:00:06"
16  },
17  {
18    "dataStreamId": 38,
19    "payload": {
20      "type": "wifi",
21      "scanned_device": {
22        "2c:9f:fb:cb:65:5c": -57
23      },
24      "start_timestamp": 1643583602,
25      "end_timestamp": 1643583612,
26      "device": "Latitude: 46.0455936 Longitude: 14.4867328 Accuracy: 1146.6127349598546\n",
27      "serial": "000000063a2d9d9"
28    },
29    "timestamp": "2022-01-31 00:00:13"
30  },
31  {
32    "dataStreamId": 38,
33    "payload": {
34      "type": "wifi",
35      "scanned_device": {
36        "2c:9f:fb:cb:65:5c": -81
37      },
38      "start_timestamp": 1643583612,
39      "end_timestamp": 1643583622,
40      "device": "Latitude: 46.0455936 Longitude: 14.4867328 Accuracy: 1146.6127349598546\n",
41      "serial": "000000063a2d9d9"
42    },
43    "timestamp": "2022-01-31 00:00:25"
44  },
45 }
```

Figure 16 Example of data in DB gathered from a smartphone

### 5.2 Detecting unique WiFi interfaces

To define the actual number of smartphones in a certain area covered by a specific wireless gateway device in a completely anonymous and unobtrusive way, we developed and implemented an approach for de-randomization of MAC addresses of the WiFi interfaces integrated in smartphones. In this respect we exploit the data collected by wireless gateways (rPi) which can be, after the processing phase, further used for different use cases, namely defining number of people at location of interest in given time or at different time frames, analyses of population movement trends (streets, shopping malls, etc.) in different timeframes, crossings / passages counting (bridges, entrances, streets, etc.), emergency exits/directions adoptions, etc.

#### 5.2.1 Data collection

To de-randomize WiFi devices and identify the number of WiFi enabled smartphones in the specific area, the information about probe requests from IE fields, time of arrival (TOA), RSSI and SSIDs are crucial. With the collected information two different WiFi devices can be distinguished or a WiFi device sending probe requests with different MAC addresses can be

detected. The proposed procedure considers that data in probe requests are strongly influenced by the chipset, device driver and WiFi software stack.

The first step in collecting the information regarding probe requests was to identify which IEs and other probe request information are relevant. The goal is to choose IEs with high entropy, meaning that the information in IE are considerably different. On the other hand, the information in IEs has to be stable for a particular device not to estimate a device as more devices.

The following information about each received probe request are collected: Supported Data Rates, Extended Supported Rates, HT capabilities, Extended Capabilities, data under Extended Tag and Vendor Specific Tag, Interworking, VHT capabilities, RSSI, SSID and timestamp when probe request was received. Note that not every probe request includes all specified parameters (all IE fields except Supported Data Rates are optional) so information about which IEs WiFi device is transmitting is also relevant information.

As mentioned, probe requests are collected by wireless gateways (rPis) with an additional WiFi dongle configured in the *monitor* mode. The gateways scan for probe requests for a predefined time (scanning time). In the next step the data are uploaded to the remote database and pre-processed.

### 5.2.2 Data pre-processing

For each detected probe request in "scanning interval" IEs data in the following structure are retrieved:

```
pr_IE_data =
{
    'DATA_RTS': {'SUPP': DATA_supp , 'EXT': DATA_ext},
    'HT_CAP': DATA_htcap,
    'EXT_CAP': {'length': DATA_len, 'data': DATA_extcap},
    'VHT_CAP': DATA_vhtcap,
    'INTERWORKING': DATA_inter,
    'EXT_TAG': {'ID_1': DATA_1_ext, 'ID_2': DATA_2_ext, ..},
    'VENDOR_SPEC': {'VENDOR_1': {'ID_1': DATA_1_vendor1, 'ID_2': DATA_2_vendor1, ..},
                    'VENDOR_2': {'ID_1': DATA_1_vendor2, 'ID_2': DATA_2_vendor2, ..}, ...}
}
```

Supported Data Rates and Extended Supported Rates are arrays with values representing which data rates are supported by a WiFi device. The rest of IEs data is represented in hexadecimal format. Vendor Specific Tag is structured differently from other IEs. In this field multiple vendors IDs with multiple Data IDs with corresponding data can be present. Similarly, Extended tag can contain multiple data IDs with corresponding data. The sender's MAC address, RSSI and SSID of detected probe request are also saved. If the device does not have any of the chosen IEs the key is not present in `pr_IE_data`.

Newly detected probe requests are compared with already saved probe requests in the specific "scanning time" interval. If there is no identical probe request in a specific "scanning time" interval received from the same MAC address, then all data for this MAC address are stored in the following structure:

```
{'MAC': MAC_address, 'SSIDs': [ SSID ], 'PROBE_REQs': [PR_data] }
, where PR_data is structured as follows:
{
    'TIME': [ DATA_time ],
    'RSSI': [ DATA_rssi ],
    'DATA': pr_IE_data
}
```

The specified structure allows saving TOA and RSSI for all probe requests that are from the same MAC address and contain same `pr_ie_data`. In addition, also all SSIDs that the same MAC address broadcasted are saved. Since in our approach all probe requests from the same MAC address are collected and compared to the information received from the other MAC addresses, the SSID information is valuable when comparing two devices. The proposed algorithm for comparing probe requests rewards the degree of matching if common SSIDs are found.

If identical probe requests of the same MAC address of newly detected probe request is already stored, all probe requests saved for this MAC address under key '`PROBE_REQs`' are checked and if a probe request with same IEs and data already exists, then only '`TIME`' and '`RSSI`' keys are appended with data of new probe request. Thus, database is reduced and probe requests can be compared more efficiently. If the same probe request does not exist, a new '`PR_data`' structure is appended to '`PROBE\_REQs`'. The procedure is shown in Figure 17.

At the end of each time interval all processed data with additional metadata regarding collected data (timestamp when scanning started and ended, serial number of rPi that scanned packets) are sent to the database.

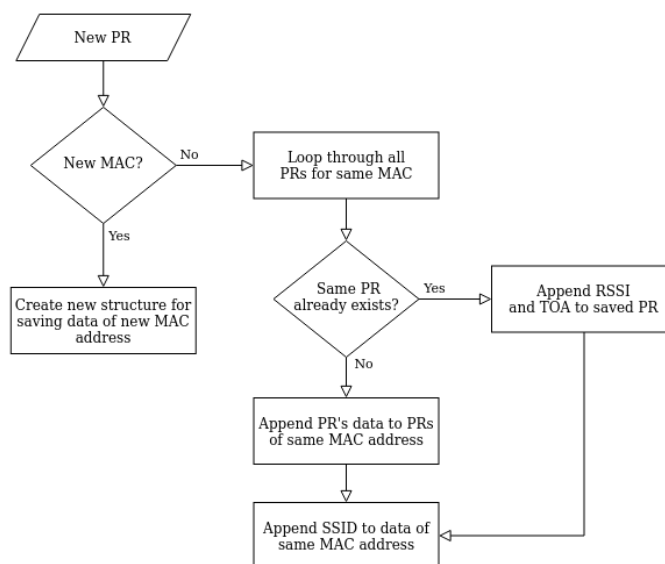


Figure 17: Saving new probe request

To get the rough estimation on the number of unique WiFi devices in a specific area at defined time (from "start time" to "stop time"), a matching algorithm which distinguishes between individual WiFi capable devices is applied on the pre-processed data.

### 5.2.3 Algorithm for distinguishing individual WiFi capable devices

To estimate the number of unique WiFi devices during a selected time interval in the area covered by a specific rPi WiFi the de-randomization of gathered probe requests received from different MAC addresses has to be performed. This can be accomplished by clustering of probe requests which are most similar. The data from two MAC addresses are compared in a way that the "distance" between two MAC addresses is calculated. MAC addresses with very similar data have small distance and vice versa.

The proposed algorithm for matching MAC addresses is composed of the following steps:



- Sorting MAC addresses in two groups; the first group contains global MACs while the second group contains all random MACs gathered which are further sorted regarding CID part of MAC address - all MACs with the same CID are moved to a separate group.
- Grouping MACs from one WiFi device by applying a matching algorithm to all groups in the random group – stage 1 clustering.
- Grouping global MACs with random MACs groups from (2) by a matching algorithm - stage 2 clustering.
- The number of WiFi devices is estimated by counting the number of groups.

### 5.2.3.1 Additional pre-processing data from the database

Since data is sent to the database in chunks for each "scanning time", these packets are merged to correspond to the interval from the "start time" to the "stop time". Additionally, the packets are filtered based on the serial number of the rPi gateway that sent data to the database, thus only data from one rPi gateway are processed at a time. Next, the algorithm for merging probe requests from the same MAC address is applied. A similar merging algorithm is used as in the first pre-processing stage in order to merge probe requests from the same MAC address within the corresponding "scanning time".

Furthermore, random devices are mapped according to CID. If more MAC addresses share the same CID they are grouped together. If no CID group is found for a specific MAC address, then it is allocated to a random group of random devices. Additional grouping enables more fine-tuned matching of similar MAC addresses since it is assumed that a WiFi device that sends probe requests with completely random MAC address does not send probe requests with CID information. The described grouping procedure is shown in Figure 18.

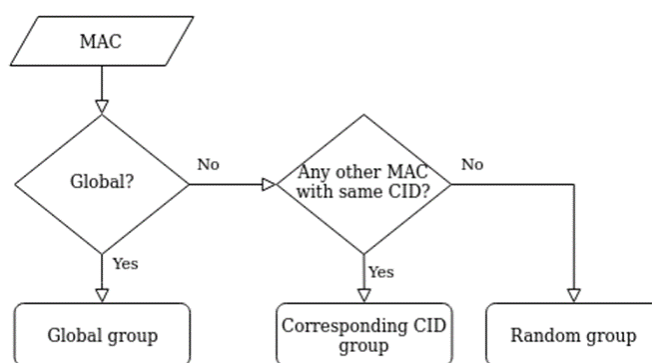


Figure 18: Grouping probe requests to the groups of global and random devices

### 5.2.3.2 Stage 1 clustering:

The first stage of the matching algorithm calculates the distance matrix for every CID group and for the random group. In particular, the distance between all pairs of MAC addresses is calculated and saved to a 2D array. The distance depends on the similarity of the probe request and IE fields, respectively. IEs that vary considerably between different WiFi devices and have also high stability for the same WiFi device, have high scale and vice versa. For example, the distance is decreased for every SSID that the two probe requests have in common while increased for every data rate they do not have in common. The distance is affected also by RSSI (increased regarding the absolute difference) and TOA (decreased if absolute difference is less than a specified threshold). For other considered fields (HT capabilities, Extended Capabilities, Extended Tag, Vendor Specific Tag, Interworking and VHT capabilities) the distance is increased proportionally to the number of different bits.

The distance for every probe request from one MAC address (N) to every probe request of another MAC address (M) is calculated, and an average distance from the lowest three values



of distances is chosen as a distance for these two MAC addresses and written to the NxM distance matrix. Distance matrix of random MACs represents an input to a density-based clustering algorithm OPTICS (Ordering Points to Identify the Clustering Structure) [11]. The algorithm arranges data points in a way that spatially closest points become neighbors. To detect changing density of points the OPTICS algorithm defines two further parameters for each point – the core distance and the reachability distance. The core distance is undefined if a point is not a core point and it is a minimum value of radius required to classify a given point as a core point. The reachability distance is defined for a chosen point in relation to another point. It is the maximum value of distance between these two points or the core distance of a point.

The OPTICS algorithm does not explicitly cluster the data in groups. Its output is visualization of the reachability distances points in the same order as processed by the algorithm. The order in which the algorithm chooses points is based on the reachability distances. The point with the smallest reachability distance is picked first. The resulting 2D plot is called reachability plot and is shown in Figure 19. The points belonging to the same cluster have low reachability distance, so they show up as valleys on the reachability plot. The valleys are separated with spikes that correspond to distances between clusters or a cluster to noise point. In other words, spikes in the reachability plot denote start of a new cluster as shown in Figure 20.

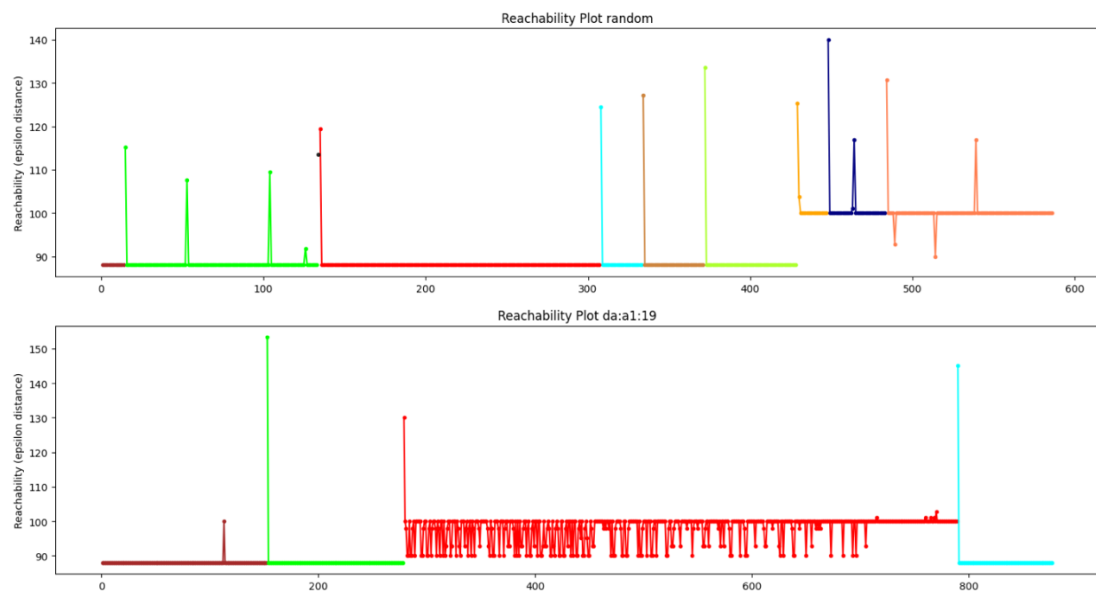


Figure 19: Reachabilityplot

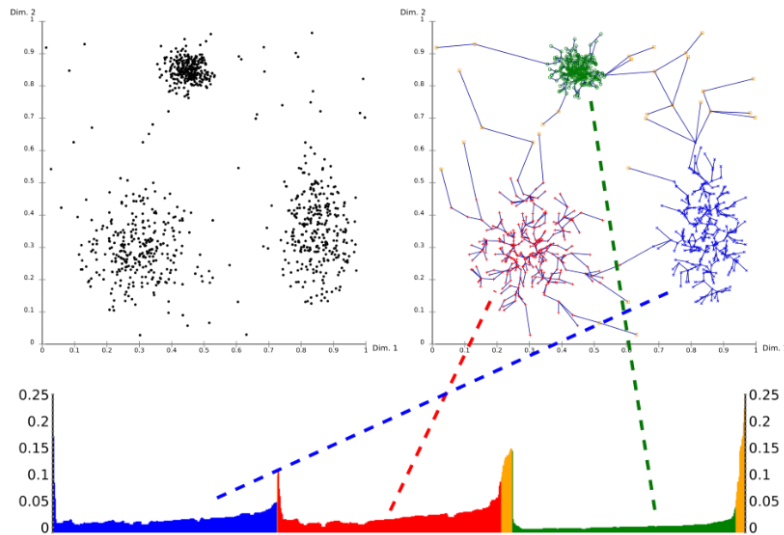


Figure 20: Clustering based on the reachability plot

In the proposed approach the OPTICS algorithm from the scikit-learn library written in Python is used. The specific OPTICS algorithm implementation includes clustering algorithm which has been adjusted and simplified for the given clustering problem. The main principle of the new algorithm is to detect the point in the reachability distances from where the curve begins dropping. If the dropping is more than the predefined threshold the subsequent points are grouped in a new cluster. When the values are increasing and exceed the predefined threshold, the end of the cluster is indicated. The OPTICS algorithm is applied to every CID group and to the random group. As a result, clusters of MACs that likely correspond to the same device within each CID group and the random group are acquired. Data from MAC addresses that are clustered together are merged together and further used for stage 2 clustering.

#### 5.2.3.3 Stage 2 clustering

In the final step, stage 2 clustering, the global MAC addresses are matched with the clustered MACs from stage 1. Similar to stage 1 of clustering, the distance matrix between each global MAC address and each group clustered in stage 1 is calculated. The distances between the global MACs and the distances between the groups from stage 1 containing random MACs are set to infinity to prevent matching.

Stage 2 matching algorithm iterates the global MACs and for each global MAC the distances to the groups of random MACs are checked. If the smallest distance between the global MAC and a given random MAC is less than the specified threshold and the next smallest distance is larger than the first for a given ratio, then the group of random MACs with the smallest distance is a good candidate for matching with the global MAC. In addition, all distances for the selected group of random MACs are checked. If the smallest distance from the previous step matches the smallest distance for the current random MAC and the next smallest distance is larger by a certain factor, these two global and random MACs are matched.

#### 5.2.4 Algorithm implementation availability

The implementation of the described procedure (algorithm) for de-randomizing the MAC addresses of WiFi enabled smartphones is publicly available on the GitHub repository at the following link:

[https://gitlab.com/e62Lab/resiloc\\_project/public/wireless-data-analysis](https://gitlab.com/e62Lab/resiloc_project/public/wireless-data-analysis).

## 6 Conclusions

This report has been prepared with the aim to briefly describe hardware and software components that constitute deliverable D4.5, developed as RESILOC tools for complementary collection of the dynamic data important for strategic planning of resilience measures, and to provide links to the repositories where these components are made available for download. As such, this report represents only a brief reference guide for installation and deployment of those components, while their more detailed descriptions, linking to the RESILOC platform and the rest of the project are provided in the deliverable D4.6.

## 7 References

- [1] Bluetooth Core Specification, version 5.1, Jan. 2019. [Online]. Available: [https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc\\_id=457080](https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080)
- [2] IEEE 802.11 Wireless Local Area Networks. [Online]. Available: <https://www.ieee802.org/11/>
- [3] Raspberry Pi. [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [4] JavaScript Object Notation (JSON). [Online]. Available: <https://www.json.org/json-en.html>
- [5] Message Queue Telemetry Transport (MQTT). [Online]. Available: <https://mqtt.org/>
- [6] Tshark, [Online]. Available: <https://tshark.dev/>
- [7] REpresentational State Transfer (REST). [Online]. Available: <https://restfulapi.net/>
- [8] Minew i9 Coin Tag. [Online]. Available: <https://www.minewstore.com/product/i9-coin-tag/>
- [9] LimeSurvey. [Online]. Available: <https://www.limesurvey.org/>
- [10] PostgreSQL. [Online]. Available: <https://www.postgresql.org/>
- [11] Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2), 49-60.