D5.2 - Initial GOEASY data-oriented enablers and applications

Deliverable ID	**D5.2**
Deliverable Title	**Initial GOEASY data-oriented enablers and applications**
Work Package	**WP5**
Dissemination Level	**PUBLIC**
Version	**1.0**
Date	**2018-11-28**
Status	**complete**
Lead Editor	**GAPES**
Main Contributors	**Leonardo Fringuelli, Peter Rosengren**

Published by the GOEASY Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776261.

Document History

Version	Date	Author(s)	Description
0.1	2018-10-17	GAPES	First Draft with TOC
0.2	2018-10-31	GAPES	Added Entity – Relationship Diagram for ApesMobility
0.3	2018-11-26	GAPES, CNET	Added content to paragraphs 2.1, 3.1, 3.2, 4
0.4	2018-11-27	ISMB	Review and suggestions
1.0	2018-11-28	GAPES	Final check
1.1	2019-02-07	GAPES, CNET	Post review corrections

Table of Contents

Document History 2

Table of Contents 2

1. Introduction 3

1.1. Scope 3

1.2. Related documents 3

2. Initial domain data models 3

2.1 Initial domain data model for AsthmaWatch app 4

2.2 Initial domain data model for ApesMobility 4

3. Initial data-oriented enablers and applications 5

3.1. AsthmaWatch - Initial data-oriented enablers and application 6

3.1.1 AsthmaWatch – Specification 6

3.1.2 Initial data-oriented enablers for the Application Scenarios - AsthmaWatch - Base code 7

3.1.3 Initial application AsthmaWatch 9

3.2. ApesMobility - Initial data-oriented enablers and application 10

3.2.1 ApesMobility – Specification 10

3.2.2 Initial data-oriented enablers for the Application Scenarios - ApesMobility 12

3.2.3 Initial Application ApesMobility 12

4. Conclusions 18

Acronyms 18

List of figures 18

1. Introduction

This work package, i.e. WP5, is responsible for developing the two mass-market applications used to validate the GOEASY concept, namely ApesMobility and AsthmaWatch, as well as providing integration and testing for the whole project.
Within WP5, this task, i.e. T5.2, taking advantage of the infrastructure built on Task 4.1, will produce the algorithms both on a client and on a server side and, based on the acquired data, will support the operations of the two mass-market applications. The definitions of the algorithms are detailed in dedicated sections of the present document.
This task will also produce the codebase of the two mass-market applications.
For what concerns the AsthmaWatch app, this task includes the development of the cloud based services, to integrate data from the mobile air pollution sensors within the GOEASY infrastructure, and the development related to the adaptors for the Copernicus Earth Observation system to extract climate and weather forecasting data as a complement to the mobile sensing data gathered, to be used and integrated into the GOEASY platform and The AsthmaWatch app. The task includes also the development of the mobile AsthmaWatch app and its integration with the GOEASY cloud-based services. Details and codebase of the development for AsthmaWatch and above listed cloud services, are listed in a dedicated section of the present document.
For what concerns the app ApesMobility, this task includes the integration with the existing platform greenApes and possible mobility apps in the city of Torino. Details of the integrations are outlined in a dedicated document section.

1.1. Scope

Scope of this document is to present the project initial implementation of the requirements described in Vision Scenarios and Use Case Definition (D2.1) and the design presented in the Initial GOEASY Platform and Pilots reference Architecture (D2.2) document, with reference to the mass market applications and their integrations with the GOEASY Platform and 3rd party services. The deliverable in this document will serve as the basis for D5.4 and D5.6, among others.

1.2. Related documents

ID	Title	Reference	Version	Date
[RD.1]	D2.1 - Vision Scenario and Use Case Definition		1.0	2018-02-27
[RD.2]	D2.2 - Initial GOEASY Platform and Pilots Reference Architecture		1.0	2018-05-31
[RD.3]	D4.1 - Cloud-based enablers for LBS provision		1.0	2018-10-31

2. Initial domain data models

2.1 Initial domain data model for AsthmaWatch app

As described in deliverable D4.1, the IoT generated data streams will be stored according to the OGC SensorThings data model. This will be complemented with a domain specific data model for each GOEASY application. Below is the initial data model for the AsthmaWatch use case.

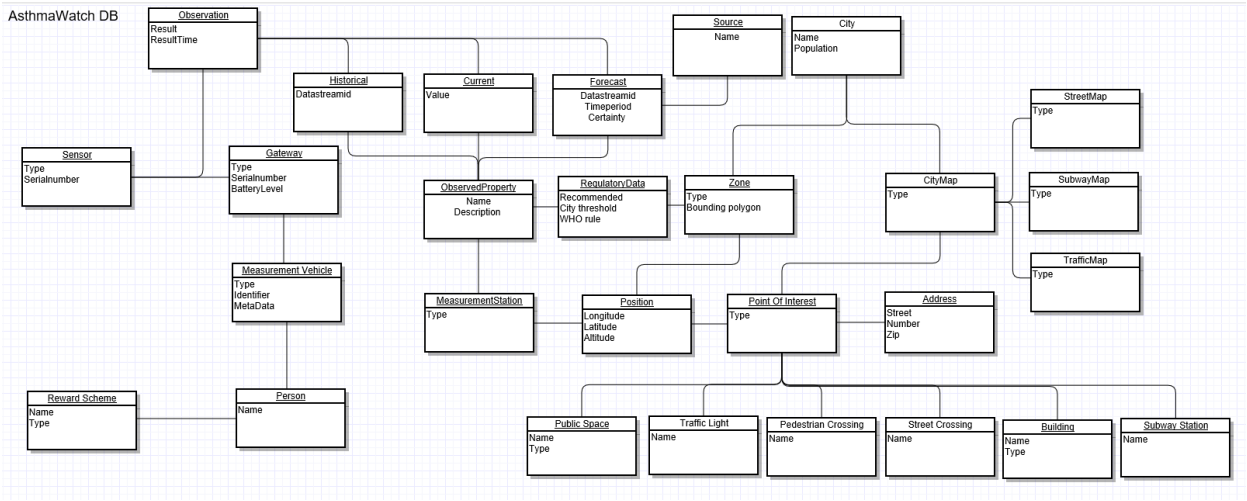

Figure 1: AsthmaWatch E-R Diagram

2.2 Initial domain data model for ApesMobility

Hereafter the initial Entity – Relationship Diagram (Chen Notation) for ApesMobility. This diagram is likely to be updated as the development progresses.

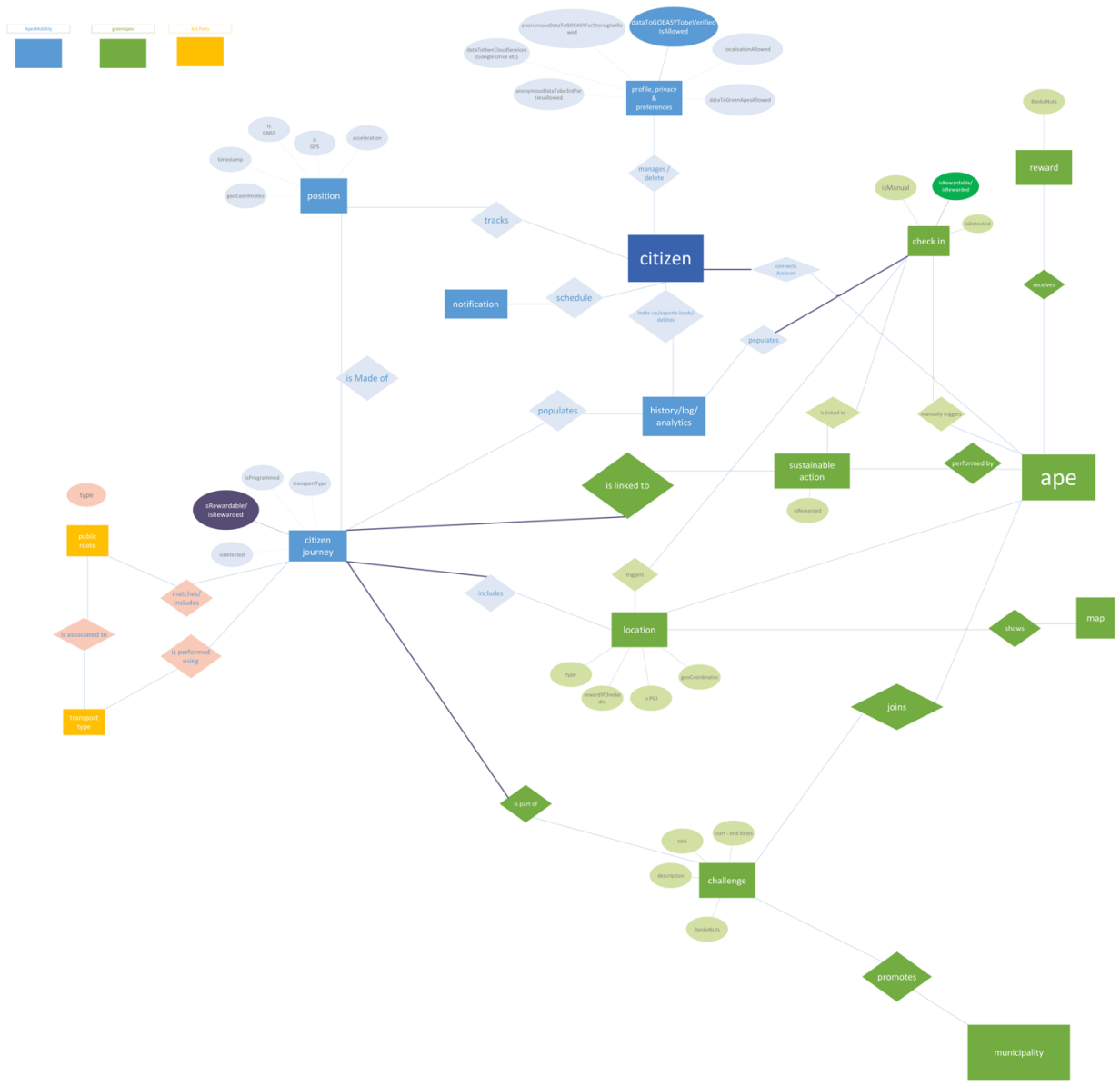

Figure 2: ApesMobility E-R Diagram

3. Initial data-oriented enablers and applications

3.1. AsthmaWatch - Initial data-oriented enablers and application

3.1.1 AsthmaWatch – Specification

Below is a listing of the user stories expressed for AsthmaWatch.

GOEAS-16	As a user I want to know about the condition in an area to decide if I will go there or not.
GOEAS-17	As a user I want to know the alternative routes based on a specific criteria in order to avoid an asthma attack.
GOEAS-18	As a user I want to keep track of my health in order to react in time on changes.
GOEAS-19	As a user I want to inform specific persons about an asthma attack in order to get immediate help.
GOEAS-21	As a user I want to define rules for a specific criteria in order to be warned in relevant situations.
GOEAS-22	As a user I want to be notified when I am within a certain range of a dangerous spot.
GOEAS-23	As a user I want to be able to view information of a single measuring station to decide if I will go to this area or not.
GOEAS-24	As a user I want to be able to define a preferred route in order to get advised whether to take it or not.
GOEAS-25	As a user I want to be able to import a route from 3rd party app(s) in order to reduce redundant tasks (defining the same route in different applications).
GOEAS-26	As a user I want to get alternative routes suggested automatically if conditions along my preferred route exceed my rules in order to avoid an asthma attack.
GOEAS-27	As a user I want to be able to view details of any suggested route in order to avoid an asthma attack.
GOEAS-28	As a user I want to be able to export a picked route in order to start navigation with a 3rd party app.
GOEAS-29	As a user I want to be able to enter health information manually to keep track of my health.
GOEAS-30	As a user I want to be able to add health information automatically by a connected device to keep track of my health.
GOEAS-31	As a user I want to be able to enter health information to the application in order to keep track of my health.
GOEAS-32	As a user I want to get immediate feedback after adding health information in order to react in time on changes.
GOEAS-33	As a user I want to be able to view historical data in order to know about the development of my health.

GOEAS-35	As a user I want to control the map by known interaction patterns to get the required information as fast and easy as possible.
GOEAS-36	As a user I want to be able to specify multiple persons as emergency contacts to increase the probability to get help.
GOEAS-37	As a user I want to be able to activate an alarm in case of an asthma attack in order to get immediate help.
GOEAS-38	As a user I want to inform my emergency contacts about my position when I activated the alarm to get help as fast as possible.
GOEAS-39	As a user I want my phone to play an alerting sound when I activated the alarm to get immediate help by people around me.
GOEAS-40	As a user I want to view all outgoing calls in the app due to an activated alarm to know who was informed and who I should update about my situation now.
GOEAS-41	As a user I want to be able to cancel a call at any time to avoid unnecessary attempts of contacts.
GOEAS-42	As a user I want to be able to select text messages to send to all emergency contacts in case I cancelled a call in order to let them know about my current situation.
GOEAS-43	As a user I want to be able to create and update templates for text messages to inform people faster about my current situation.
GOEAS-44	As a user I want to be able to determine the order in which contacts are contacted to prioritize based on criteria.

3.1.2 Initial data-oriented enablers for the Application Scenarios - AsthmaWatch - Base code

Integration of GOEASY sensor data

The sensor data from IoT gateways will be transmitted as OGC (Open Geospatial Consortium) message and stored according to the OGC (Open Geospatial Consortium) SensorThings Data model. This model has been described in deliverable D4.1 “Cloud-based Enablers for LBS provision”. The domain data model is described in section 2 above.

Integration of third party sources

Obtaining data from third party sources is done in order to complement measurements made with the mobile air quality sensors and help validating the results. The data is retrieved from four sources:

- Swedish Environmental Protection Agency (Naturvårdsverket)
- World Air Quality Index website (WAQI)
- Swedish Meteorological and Hydrological Institute (SMHI)
- Copernicus Atmosphere Monitoring Service (CAMS).

Each of these services provide open access to different types of environmental data, where mainly the level of different pollutants is of interest for the AsthmaWatch use case.

Each of the sources provide an API for retrieving specific data which is utilized in the implemented service. All sources except CAMS depend solely on data from fixed measurement stations with a location and a set of sensors providing the information. Therefore, each of the station’s measurements are obtained by

specifying a unique identifier in a REST call made to the endpoint. The response from the services are JSON strings containing information such as pollutant type, pollution level, time, location only to name a few.

Data from CAMS is defined as Gridded Binary (GRIB) files, a standard used in meteorology for storing historical and forecasted weather data. Provided in this service is historical and forecast data, which are mainly based on calculations and models of how the predicted state of the different pollutants will be. It is possible to retrieve weather data up to three days in the future. The main difference from the other sources is that data is not retrieved for a specific location. Instead, an area (grid) together with pollutant type and time for when the forecast should be valid. This results in a GRIB file where values together with locations within the defined grid are defined in a checkerboard-like pattern for the specific time. The points are spaced by approximately one to two kilometers depending where on Earth the grid is defined.

Common between all data-sources is how the information gets formatted before being sent for storing. The format used follows the OGC SensorThings format, a standard used for defining and connecting IoT devices to the web. Using JSON to define the structure of a message, they contain fields to define the identifier of the device, time of the phenomena, time of creating the message, coordinates, value and many others. An example of the format used can be found in Figure 4.

For transmitting messages from the service to a storage solution, the lightweight MQTT protocol is used. JSON messages, illustrated in Figure 4, are sent as payload to the message broker defined at the Edge Connectivity Manager. Measurement stations, real or virtual (CAMS), have set topics which they post to which they initially receive by announcing themselves to the GOST service. If there exists a data stream with the provided name, the message will be posted to that stream, otherwise GOST will create it.

```
{
“resultTime”: “2018-11-22T10:08:45”,
“Datastream”: {
“@iot.id”: 209
},
“phenomenonTime”: “2018-11-22T10:06:33”,
“result”: {
“valueType”: “NO2”,
“Position”: {
“type”: “Point”,
“coordinate”: [
59.3993652, 18.0343221
]
},
“response”: {
“value”: 6,53
}
```

Figure 3: Example of the OGC message format used as a JSON string

Common Retrieval API

Both sensor data and third party information source data are made available for developers through a standardized OGC (Open Geospatial Consortium) SensorThings API. This has been described in deliverable D4.1 “Cloud-based Enablers for LBS provision”.

Here are some examples of the query API:

- http://goeasy.cloudapp.net:8081/#/Things(id)?$expand=Datastreams
  - Retrieve the data streams for a given thing
- http://goeasy.cloudapp.net:8081/#/Datastreams(id)/Observations
  - Retrieve all observations in a certain datastream.
- http://goeasy.cloudapp.net:8081/#/Sensors
  - List all sensors
- http://goeasy.cloudapp.net:8081/#/Datastreams(id)?$select=name,Observations&$expand=Observations/FeatureOfInterest
  - Returns the name property of the Datastream entity, and all the properties of the entity identified by the Observations and FeatureOfInterest navigation properties
- http://goeasy.cloudapp.net:8081/#/Observations?$top=3&$orderby=phenomenonTimedesc
  - Returns the first three Observation entries after sorted by the phenomenonTime property in descending order.

3.1.3 Initial application AsthmaWatch

The AsthmaWatch app, see Figure 4, aims at providing users with relevant and accurate air quality information in their daily activities. Some screenshots from an early prototype app is shown below.

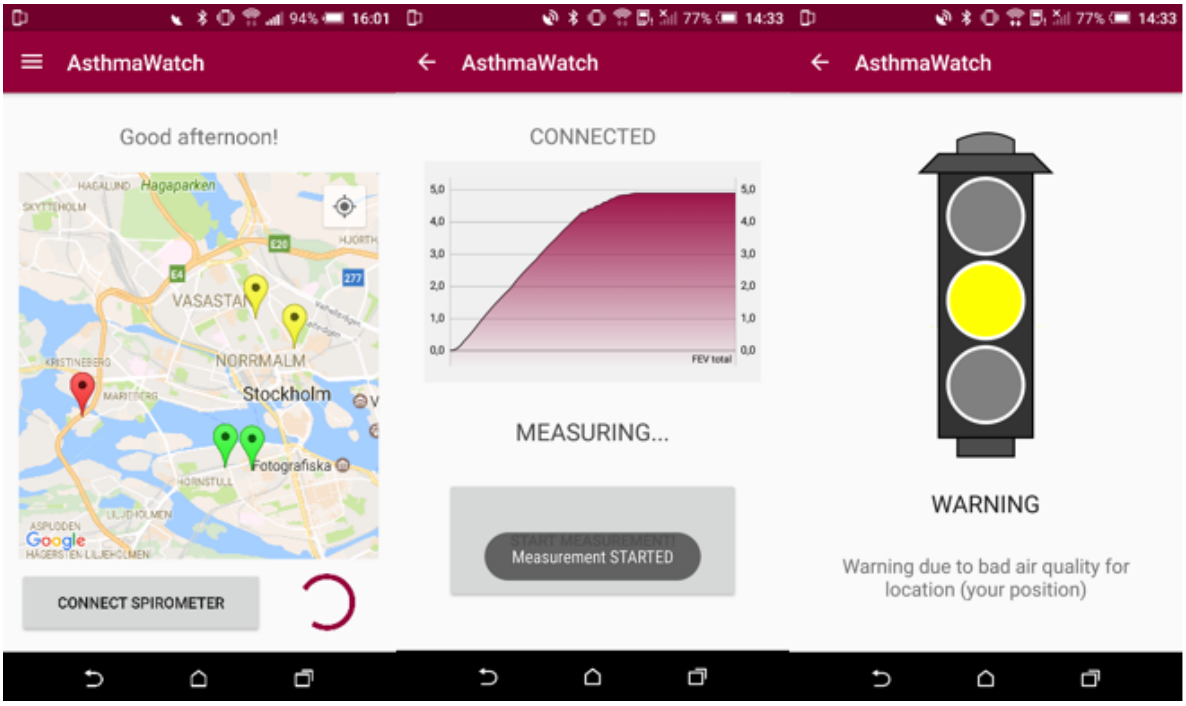

Figure 4: AsthmaWatch - Early Prototype Screens

The AsthmaWatch application consists of a smart phone user app and a GOEASY cloud backend which has been configured for the AsthmaWatch requirements. Specific micro services will be developed and instantiated, see Figure 5 below for the GoEasy Microservice architecture. This is explained in detail in deliverable D4.1 “Cloud-based enablers for LBS provision”

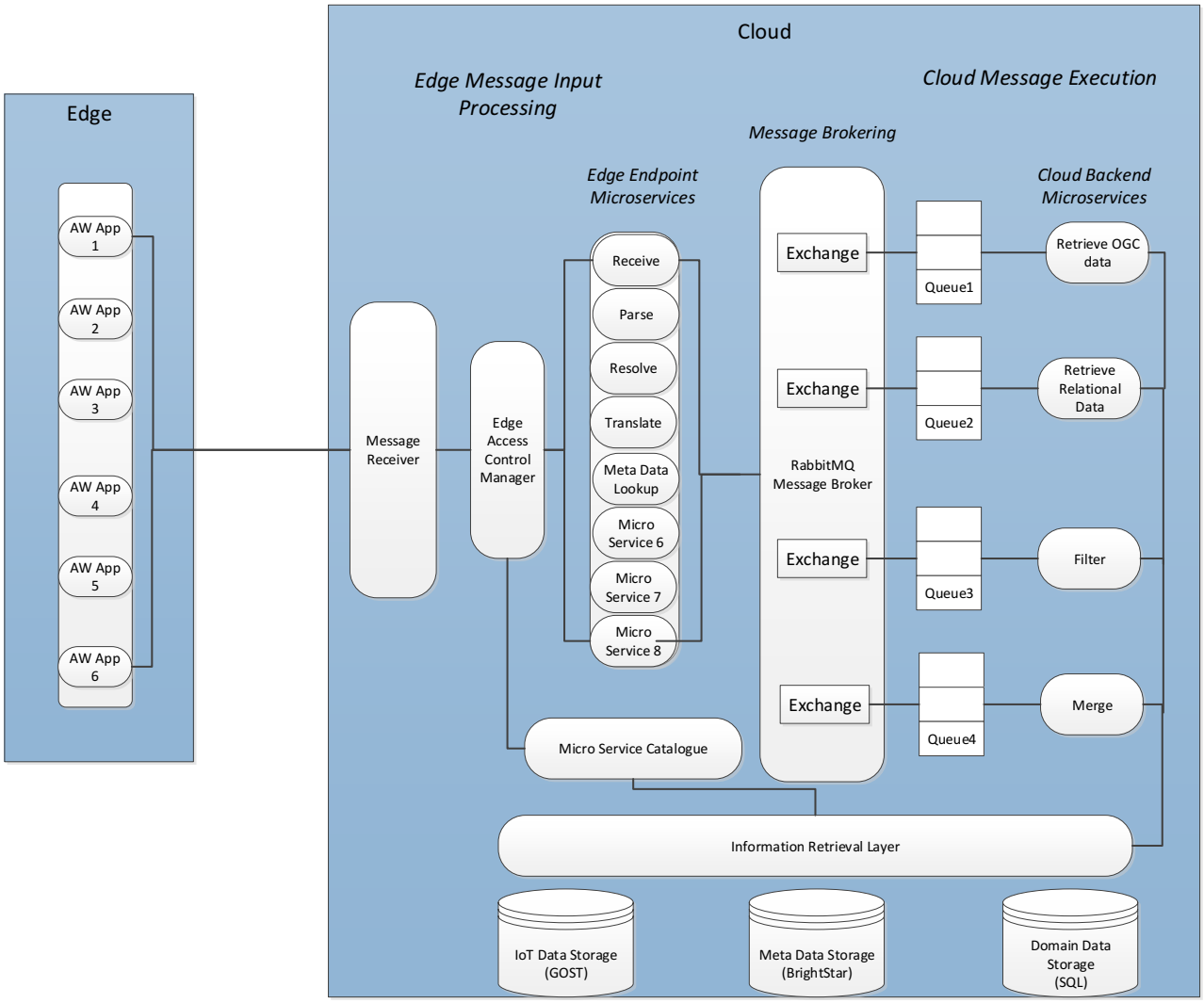

Figure 5: GOEASY Microservices architecture

3.2. ApesMobility - Initial data-oriented enablers and application

3.2.1 ApesMobility – Specification

Requirements for ApesMobility app have been outlined as User Stories on the Jira GOEASY repository and referenced with the the tag “ApesMobility”.

The main purpose of the app is to take advantage of location based services linked to GNSS data as measured on the Android smartphones with the goal of certifying user behaviour related to sustainable mobility patterns. As such key stories are:

- "GOEAS-56: As a citizen, I want to be able to trigger detection of mobility behavior manually to share my activity immediately." This story is linked to one narrated from a developer perspective: "GOEAS-54: As a SW application developer, I want to be able to detect specific mobility behavior of users without a data connection in order to reduce data and battery consumption of the mobile device." Accessing the app the user is prompted with the possibility of tracking with GNSS sensor. As further option the user has the possibility of scheduling notifications that will act as reminder about the activation / deactivation of the tracking functionality at specific times of the day. There is also the possibility to repeat the notifications on a daily basis or Mon to Fri. Since Android 8.0 the possibility of activating GNSS through background services has been disabled hence this implementation of a notification system.

- "GOEAS-13: As a citizen, I want to get my location certified to get additional rewards."

- "GOEAS-10: As a citizen, I want to get my activity certified to get additional rewards." This story has an alter ego on the developer side that is "GOEAS-53: As a SW application developer, I want to certify specific mobility behaviour of users in order to reward them for sustainable behaviour."

- "GOEAS-9: As a citizen, I want to be able to confirm my certified activity to be additionally rewarded."

- "GOEAS-57: As a citizen, I want to be notified about a detected, certified activity in order to not miss additional rewards."

Privacy and profile management.

- "GOEAS-65: As a citizen, I want to be able to delete identifiable location data to protect my privacy."

- "GOEAS-67: As a citizen, I want to be able to manage permissions of my data in order to protect my privacy". -> this story is linked to the following -> GOEAS-48: As a SW application developer, I want to be able to aggregate data to hide details that endanger personal identifiable information.

- "GOEAS-62: As a citizen, I want to be able to en-/disable location tracking to protect my privacy.

- "GOEAS-72: As a citizen, I want to have a profile of my gathered data in order to help me changing my behaviour."
-

Some of the Jira stories are or will be implemented as part of the GOEASY project in the greenApes platform. These are the following:

- "GOEAS-61: As a citizen, I want to subscribe to specific areas or topics only in order to get relevant information only."

- "GOEAS-60: As a citizen, I want to be notified about new challenges in order to improve my sustainable behaviour."

- "GOEAS-55: As a citizen, I want to be able to share my certified activities in order to motivate fellow citizens."

- "GOEAS-15: As a citizen, I want to join challenges to get motivated and to support the community."

Stories that are more specific to the AsthmaWatch scenario but that can find also an implementation, maybe in a second step, on the ApesMobility app are:

- "GOEAS-70: As a citizen, I want to know which route is the best/ healthiest one in order to not further pollute specific areas."

Some stories are more related to the technical architecture and the way apps exchange data with each other.
Third party apps will be able to send location based data to the GOEASY platform to exploit the location based services brokered by GOEASY. The story that generically mentions this feature is the following:

- "GOEAS-47: As a SW application developer, I want to transmit data to the GOEASY platform to make it available for location-based services." This is the story that generically sets the case for third party apps

In the final version of the app there might be a functionality related to looking up on a list or on a map POI or Events with which the user can interact by checking in and with the goal of getting rewarded (being certified and / or by sharing).

- GOEAS-59: As a citizen, I want to see additional information about registered locations/ events in order to decide whether to go there or not.

- GOEAS-58: As a citizen, I want to be able to find registered POIs and challenges in order to collect additional rewards by check-in (and share).

At present there is only one story related to City of Torino or other organisations and this is related to the capability of involving citizens/users in challenges to inspire them on sustainability themes:

- "GOEAS-14: As an organization, I want to offer challenges for users in order to familiarize them with sustainable topics."

We should probably deepen further use cases related to the organisations and for instance would like to measure data and KPI on population or on the initiatives.

3.2.2 Initial data-oriented enablers for the Application Scenarios - ApesMobility

The main purpose of the GOEASY platform is acting as a hub between apps and location based services while providing an additional layer of security to GNSS. ApesMobility is one of the app that will exploit these features. As seen in the previous paragraph user's journey composed by the tracked locations will be certified within the app ApesMobility. The certification will be possible through data oriented enablers that could differ depending on the geography of the users because algorithms or services that works best in the context of a specific city might not have the same efficacy in a different location. The flexibility of the GOEASY platform to connect to different LBSs will give the added value to app developers and users of using the ones that best fit their needs. ApesMobility pilot will be in Torino so a selection process is ongoing that aims to find the LBS that will give the best accuracy while certifying the transport system in the context of the Torino urban mobility. In updated versions of this document details of the chosen LBS for the app ApesMobility will be provided, including methods for invoking such services.

3.2.3 Initial Application ApesMobility

In its final version this paragraph will contain references to the code developed for the app apesMobility. Being this the initial issue it presents the status at the current stage of the project. Wireframes have been drafted that even if have to be considered nor definitive nor complete and in need to be translated in actual mock ups, are capable of conveying a solid picture of the main interactions within the app.

Using the app user will be encouraged to adopt sustainable transport habits, that will be measured through the location based serviced leveraged by the Galileo GNSS signal and the GOEASY platform. There will be a gamification mechanism that consists in progressing through different levels gaining points that will be increased by tracking sustainable trips. This will keep in good health an avatar that will assume different shapes on each level.

The app will not require logging in but will have an on-boarding flow which will prompt the user with some options that include, but are not limited to, the consent given to the app for using the phone location based services or the consent to let the positions recorded by the app be verified by the GOEASY platform cloud services. These steps are not described by the current wireframes.
Once the on-boarding process is completed, the user lands on the main page of the app that is the first of the 3 pages reachable through a horizontal menu (see Fig. 6).

Figure 6: ApesMobility - Main page - Wire frame

Above the horizontal menu a section, visible always in the app, showing the Name of the application and an icon to reach the Setting / Profile page.

The main screen will show at the bottom a button to start / pause / stop recording the geographical coordinate.
In the middle shows the avatar corresponding to the level the user is and the indication of the level. Just below the avatar a couple of indicator are shown such as indicator related to the health status of the avatar and another related to the KMs travelled in a range of time.

The second icon in the menu bar will lead the user to the Logs page (see Fig. 7).

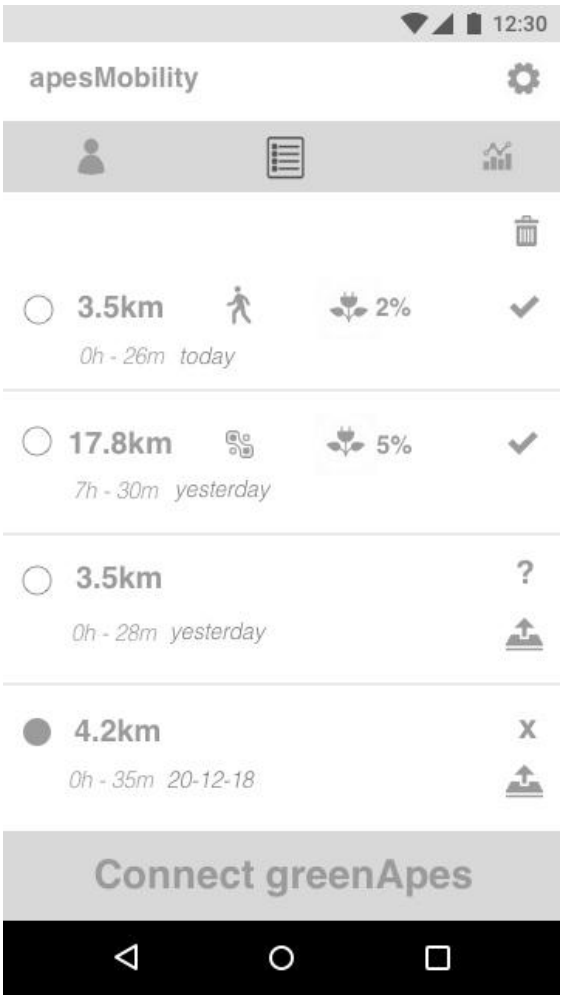

Figure 7: ApesMobility - Logs page - Wire frame

At the bottom of the page the user will have the possibility to connect to an existing or new greenApes account. This will give the chance to communicate the events related the sustainable journeys had by the users to the greenApes platform and as such to be rewarded for them. In the platform there will be also functionalities related to checking in to POIs or joining challenges linked to sustainable actions.
The central part of the page shows the sessions that have been recorded in a range of time (that can be navigated by scrolling the app vertically) and for each the main info such as duration, time and date, whether this session have been validated by the GOEASY services and the result of the validation. If the result of the validation is successful, then there will be indication of the transport system used in the session and any health points related to it. If the session has not been validated, then the user will have an icon that will allow the submission of the session to the GOEASY platform for validation.
In this screen the user will also be capable of deleting one or more sessions.
Selecting a session, a user will be able to reach a Logs details page (see Fig. 8) that will show more details for that specific session. In particular, sessions with a mixed range of transport systems used will show the breakdown of any leg of the journey.

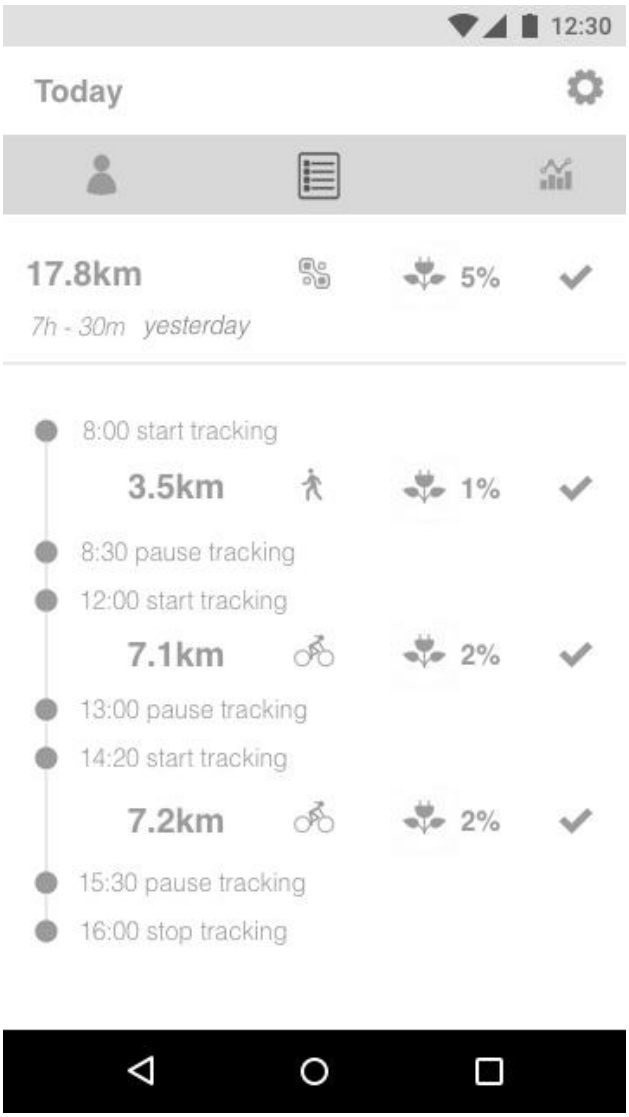

Figure 8: ApesMobility - Logs details - Wire frame

The third icon of the menu bar will lead the user to a page that will show analytics (see fig. 9) related to the user transport habits and to gamification elements in the app.

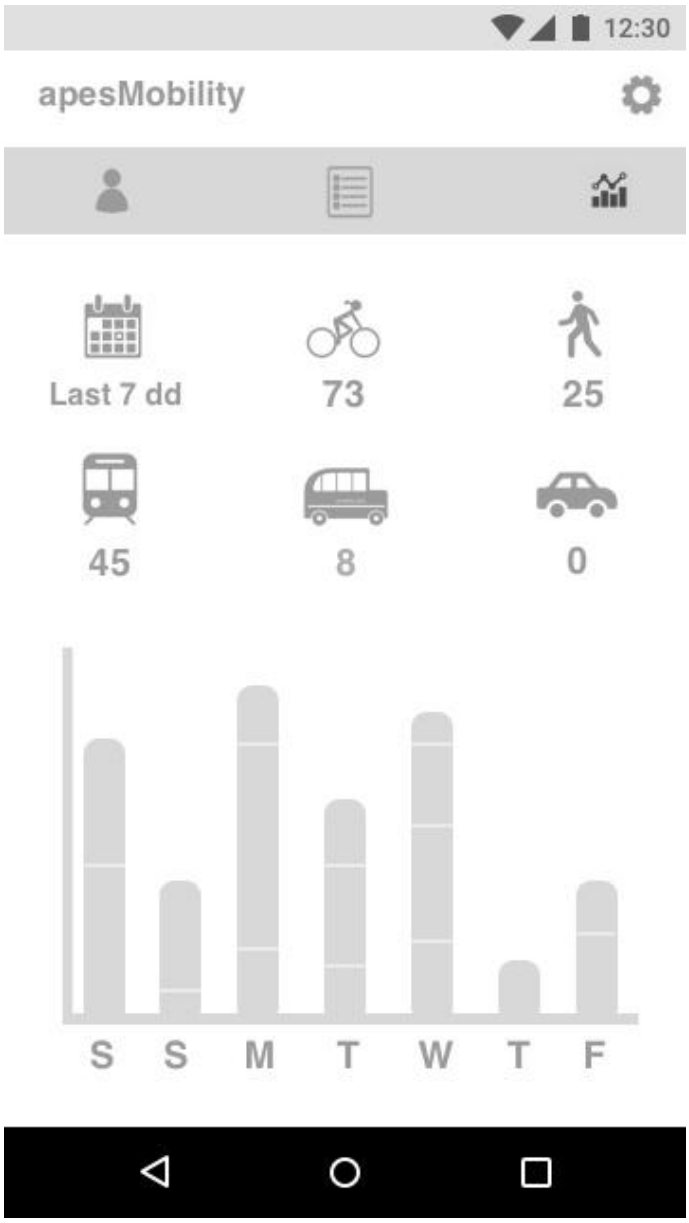

Figure 9: ApesMobility - Analytics - Wire frame

The Settings / Profile page (see Fig. 10) is a composite page that will provide the user with many interactions possibilities.

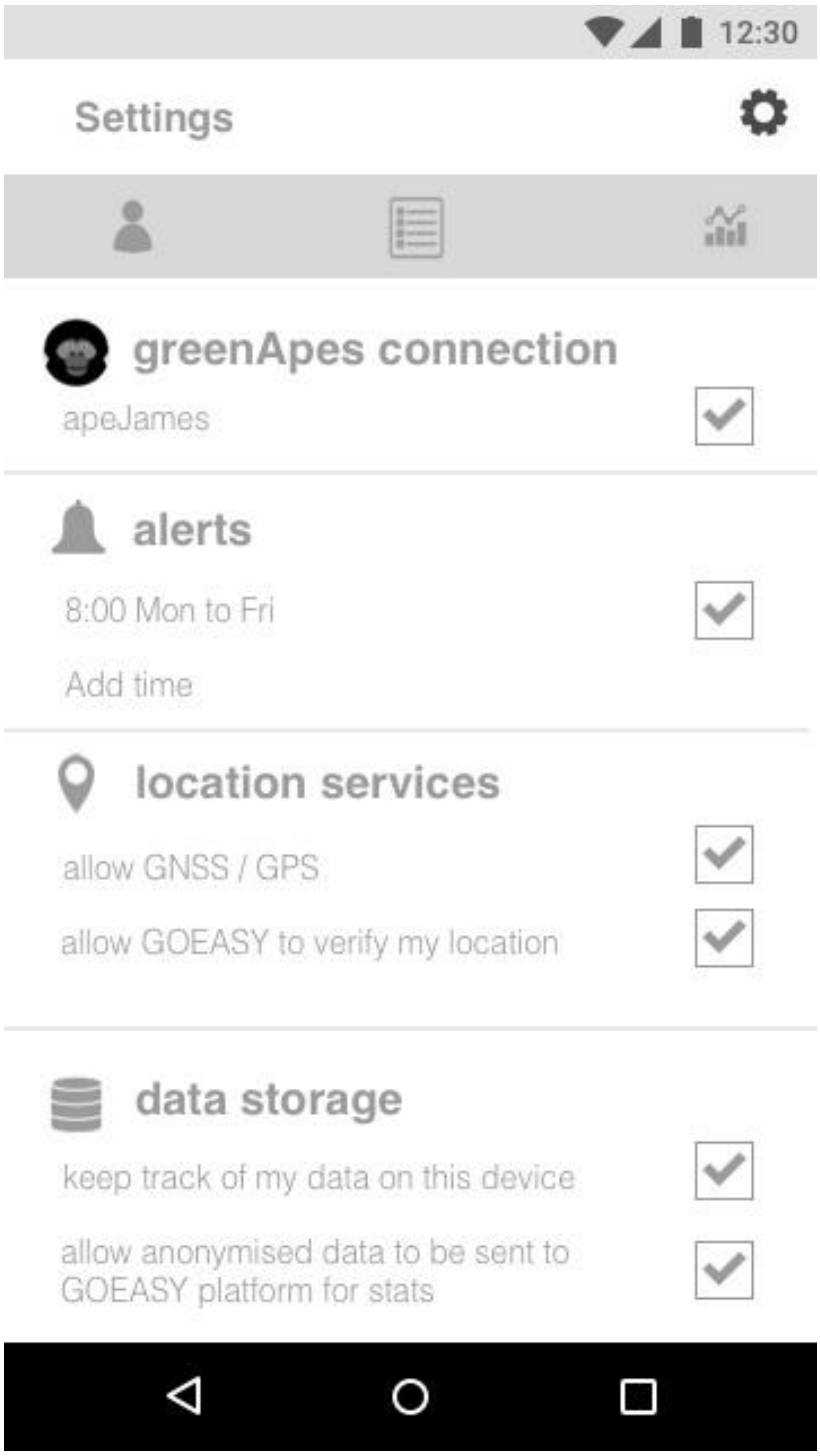

Figure 10: ApesMobility – Settings / Profile - Wire frame

The “Alerts” section will allow the user to schedule in advance notifications the will remind him to activate the GNSS tracking feature in specific also recurring times/days.
The “Connect to greenApes" section will allow connecting / disconnecting to / from a greenApes account.

A “Privacy / Consents” section will enable the user to fully manage his consents in terms of data privacy protection and to enable / disable location based services. Here the user will be able to erase any data stored locally.
In future versions of the app the user will be given the feature of exporting his data to a personal cloud based storage service such as Google Drive for then being able to load again back the data onto the app.

4. Conclusions

This document is dedicated to describe/provide the technical details of the first 2 apps implementing use cases that are fully leveraged by the GOEASY platform services. They rely on on heavy use of cloud based algorithms and services that are or directly part of the GOEASY platform (AsthmaWatch) or provided by 3rd parties (ApesMobility). In this second case the platform acts as hub/broker connecting the apps and location based services.

Acronyms

Acronym	Explanation
Dx	Deliverable number x of the GOEASY project
GNSS	Global Navigation Satellite System
IoT	Internet of Things
LBS	Location Based Services
POI	Point Of Interest
Tx	Task number x of the GOEASY project
WPx	Work Package number x of the GOEASY project

List of figures

Figure 1: AsthmaWatch E-R Diagram 4
Figure 2: ApesMobility E-R Diagram 5
Figure 3: Example of the OGC message format used as a JSON string 8
Figure 4: AsthamWatch - Early Prototype Screens 9
Figure 5: GOEASY Microservices architecture 10
Figure 6: ApesMobility - Main page - Wire frame 13
Figure 7: ApesMobility - Logs page - Wire frame 14
Figure 8: ApesMobility - Logs details - Wire frame 15
Figure 9: ApesMobility - Analytics - Wire frame 16
Figure 10: ApesMobility – Settings / Profile - Wire frame 17