SUNSET

Sustainable Social Network Services for Transport
www.sunset-project.eu

Grant agreement n°: 270228
Start date: Feb 1, 2011
Duration: 36 months
Area: Smart Cities & Sustainability (CONNECT.H.5)
Project Officer: Mr. Stefanos Gouvras

Deliverable D4.2, D4.3, D4.4 Supplement "Social Traffic Mobile Application"

Version: final
Due date of deliverable: 30/11/2012
Actual submission date: 30/11/2012
Dissemination level: CO
Responsible partner: NOVAY, QMUL, LOCNET

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 270228. The project's website is at www.sunset-project.eu.

Summary

This document provides a more detailed & technical description of the sub-applications of the main tripzoom application. It is meant as an accompanying document to the full software release of tripzoom, which is made available both as source code and as a fully functional mobile application in the two main appstores: Apple's and Google's.

To download and to get started with tripzoom, we refer to the tripzoom portal (http://www.tripzoom.eu/portal/gettripzoom.php) that provides all required information to start using the final release of the application, as it will be used for the trials in the living labs.

Document Information

Authors

Name	Partner	Email
Marko Luther	DOCOMO	luther@docomolab-euro.com
Zhenchen Wang	QMUL	zhenchen.wang@eecs.qmul.ac.uk
Athen Ma	QMUL	Athen.ma@eecs.qmul.ac.uk
Peter Ebben	NOVAY	petter.eben@novay.nl
Koen Jacobs	LOCNET	Koen.jacobs@locatienet.nl
Bertrand Souville	DOCOMO	souville@docomolab-euro.com
Johan Koolwaaij	NOVAY	johan.koolwaaij@novay.nl

Editor

Name	Johan Koolwaaij
Partner	NOVAY
Address	Postbus 589, 7500 AN Enschede, The Netherlands
Phone	
Fax	
Email	Johan.koolwaaij@novay.nl

History

Version	Date	Changes
V0.1	12-10-2012	Table of contents
V0.2	27-10-2012	Initial content by section editors
V0.3	30-10-2012	Added introduction
V0.4	5-11-2012	Added summary plus links to redmine and documentation site
V0.5	9-11-2012	Final edit cycle and consistency check
V1.0	30-11-2011	**Final version, approved by PCPs and PMT, sent to EC**

Distribution

Date	Recipients	Email
30-11-2012	SUNSET partners	sunset@lists.novay.nl
30-11-2012	Project Officer	Stefanos.GOUVRAS@ec.europa.eu
30-11-2012	Project Archive	CNECT-ICT-270228@ec.europa.eu

Table of Contents

1. Overview

This document provides a more detailed description of the sub-applications of the main tripzoom application described in D4.5. These sub-applications were foreseen as possibly separate applications in the DoW, but in the final design these were integrated into one application called tripzoom. The different sub-applications are:

- Mobile Monitoring Application
- Mobile Incentive Marketplace
- Social Mobile Application

Mobile monitoring (Section 2) is responsible for 24/7 sensing of all trips made by tripzoom users. These trips are further analyzed and enriched on the server-side and form the information foundation serving a fourfold purpose:

- to provide users with relevant and challenging incentives,
- to let user receive micro-questionnaires in very specific mobility situations,
- to provide users with overviews and summaries of historic mobility behaviour and patterns therein,
- and to allow users to exchange information about current and historic mobility information, and compare themselves to their friends and colleagues, or even to the entire living lab community.

All recorded and analyzed trips are listed in the personal trip list, where trip details can be corrected and improved by the user manually.

The Mobile Incentive Marketplace is in fact a server-side component described in D2.4. This component is responsible for the creation of new and the management of existing incentives, and providing the users with the corresponding reward if they have fulfilled the conditions for a specific incentive. The mobile counterpart of this marketplace is described in Section 3, and provides the users with an overview of potentially relevant incentives, plus an overview of all earned rewards, and receives all real-time notifications from the server-side counterpart.

The Social Mobile application is described in Section 4 and provides the users with a buddy list enriched with mobility information of friends, colleagues and family. All underlying information is requested from the Context Harvester which bundles different information streams from the various server-side components into one stream. Using the buddy list, a user can be aware of the mobility situation of his/her friends, and compare among themselves in terms of travel duration, distance, costs or emissions.

The mobile application tripzoom is tightly integrated with the SUNSET platform. In the following sections describes how the individual server-side interfaces are used by the different mobile components. But for a full description of these interfaces we refer to the deliverables D2.1 [3], D2.2 [2] and D2.3 [1], or to the documentation page, which serves as a primary reference for the documentation of the architecture of the SUNSET system. This page can be accessed using the following login details:

URL: http://www.tripzoom.eu/sps/
Reviewer user name & password: reviewer / s3sam3

All information about the different software releases and evaluations, accompanied by all implemented features and reported/resolved bugs during development time, can be found in the SUNSET RedMine system:

URL: https://dermine.docomolab-euro.com/projects/sunset/wiki
Reviewer user name & password: reviewer / s3sam3

The sources of the tripzoom mobile clients can be found in the confidential deliverables folder, using the following resource:

URL: https://doc.novay.nl/dsweb/Services/Document-137400

2. Mobile Monitoring Application (D4.2, NOVAY)

The Mobile Monitoring Application is part of the tripzoom application and it runs mostly in the background to track the whereabouts of individual users and to make sure that the relevant data gets uploaded to the tripzoom server for further processing (detailed trip analysis, map matching, modality detection) so that it can be used for the calculation of the personal scores (scores of the users compared with the scores of the community). Based on the uploaded trip data, experience sampling questions may be issued to the users, and they might get a reward if their behaviour complied with one of the incentives or challenges that have been defined in the system.

In this section, the lower-level parts of the tripzoom application will be described in detail. The way the gathered data is used / presented to users is described in section 4.

Given the differences between the Android and iOS operating systems, there are some variations in the implementation of the sensing parts on Android and iPhone. Where applicable, the core differences will be described.

2.1 Features and Modules

The Mobile Monitoring Application consists of two main building blocks:

- **The Sensing Machine**
  This is the part that collects information based on the available sensors, carries out the required pre-processing, and decides which data to store for future upload. This part of the application is described in more detail in section 2.2.

- **The Communication subsystem**
  This is the part that decides when to upload data to the server. It tries to bundle (upload) requests as much as possible in order to reduce battery consumption. A second part of the Communication subsystem uses the cloud to device messaging (push notifications) functionality provided by Google and Apple so as to present notifications of any new challenges, rewards earned, experience sampling questions or invitations to become friends to the users. Finally, the Communication subsystem provides a single interface for all UI component that need to retrieve data from the tripzoom server. The Communication subsystem is described in more detail in section 2.3.

2.2 Sensing Machine

The Sensing Machine consists of a Sensing Manager, a number of Data Collectors and a pre-processor to reduce the amount of data that needs to be uploaded to the server.

The Sensing Manager provides access to relevant device sensors:

- Network Location sensor
- GPS
- Accelerometer
- Battery sensor
- Magnetometer – on Android only

- External Ant+ sensors (Bike, Footpod, Heart Rate Monitor [4]) – on Android only

The Sensing Manager allows the data collectors to start and stop individual sensors.

Two data collectors are implemented: A simple data collector that just collects battery statistics, logging an entry to the database every time the battery load increases (while charging) or decreases (during normal operation) with a given percentage, or when the battery is connected to (or disconnected from) a charger.

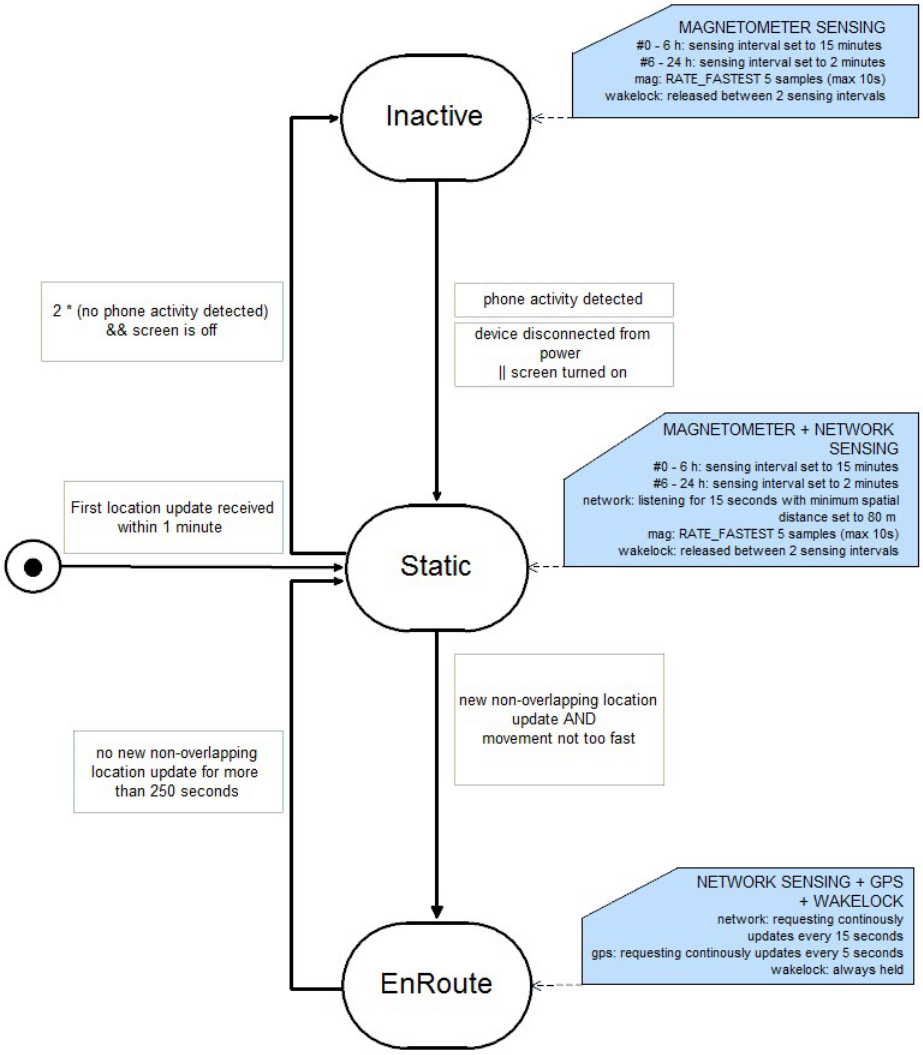
The second data collector is more complex: the location collector gathers location data in a battery-efficient way. It uses the Network Location Sensor and GPS to determine if a person is travelling. During travelling (Enroute State in the Sensing Machine), it will record a location trace, that will later be uploaded to the server via the Communication subsystem. Pre-processing is done to restrict the number of measurements that are uploaded: measurements that overlap with the previous location are discarded, unless they are more precise.

As soon as the location collector detects that a user has not moved for a few minutes (Static State in the Sensing Machine), the expensive (in terms of battery consumption) GPS sensor is turned off, and the device periodically checks if the user has moved away from their previous location (which is a trigger to start tracking their location in detail).

On Android, the Magnetometer is used to detect periodically in the background whether the phone orientation has changed. This can be measured by comparing the 3D magnetic field vectors between two sensing intervals (every two minutes during the day, every fifteen minutes during the night). As long as the magnetometer does not detect a phone orientation change, the CPU is sleeping (the wakelock is released), the network location sensing and the communication subsystem are turned off to save energy.

This "Inactive Mode" State in the Android sensing machine only applies when the screen is turned off and is disabled for Android device models that are not sensing in the background. As soon as the magnetometer detects a phone orientation change, the screen is turned on or the device is disconnected from power, the communication subsystem and network location tracking are enabled again. If a non-overlapping location is detected (signalling the start of an "EnRoute" trip) GPS is enabled again.

The figure below details the sensor configurations for the three application states (Inactive, Static and EnRoute) of the Android sensing machine.



On selected Android devices, it is also possible to log data from external Ant+ sensors. If the phone has detected that the phone orientation has changed, and the phone provides support for Ant+ devices, it tries to search for bike, footpod and heartrate monitor sensors. If these devices are found, their measurements are recorded as well.

The iPhone implementation relies on OS level functionality to determine if a user has moved significantly as to initiate the detailed sensing mode. This functionality wakes the application up with a notice that such an event has occurred, allowing the device to suspend this background application. This results in only little battery usage when the user is not moving. The application subscribes to two events to be woken up from the background. First, significant location changes are used to determine if a user has moved, presumably relying on cell switches and

passively consumed location data as requested by other applications. Second, region monitoring notifies the application of the event where a user has left a certain area.

The sensing subsystem provides hooks for other components to register for status updates on trips: interested components can register a trip listener, and they will get notified upon every status change (from enroute to static or vice versa).

Note that the Sensing subsystem itself does not upload any data to the tripzoom server; that is the responsibility of the communication subsystem described in the next subsection.

2.3 Communication

The Communication subsystem provides the following main functions to the tripzoom application:

- Periodical upload of sensed data to the tripzoom server
- Periodical download of (enriched) data from the tripzoom server, for use by user interface components
- Support to upload manual changes to the trip details by the user, e.g. correcting a erroneously detected modality
- A single point to initiate synchronous communication with the server
- Support for notifications

2.3.1 Periodical upload of sensed data

Periodically, the communication subsystem will check if there is new data to upload. It typically does this every few minutes. In an effort to save battery, we do not upload every piece of data as soon as it becomes available. This would cause the WiFi or UMTS radio to be active for most of the time, especially during trips (since locations are collected and stored).

The Communication subsystem therefore checks every time if there is a new trip that needs to be uploaded. If a new trip has been added to the local database, we know for a fact that the trip has ended and that all locations that are part of the trip are stored in the local database as well. The existence of a new trip in the database triggers the upload of the trip, the relevant locations, the collected battery statistics, and – if applicable – bike and footpod measurements.

It invokes the following API calls on the Measurement API:

Data type	API Endpoint	Description
Trips & Locations	~api/measurement/Trips/withLocations	HTTP POST Payload consists of a JSON object containing an array of trips and location measurements
Power	~api/measurement/Power	HTTP POST Payload consists of a JSON object containing an array of power measurements
Bike	~api/measurement/Bike	HTTP POST Payload consists of a JSON object containing an array of bike measurements

Footpod	~api/measurement/Footpod	HTTP POST Payload consists of a JSON object containing an array of footpod measurements

All these calls are signed via OAuth. The full specification of the server-side interface to store and analyze trip and location data is provided in D2.1 [3]. The patterns used to analyze trips, to enrich these, and to deduce new or aggregated information from the collection of uploaded trips are described in D2.2 [1].

2.3.2 Periodical download of (enriched) data from the tripzoom server

Since the tripzoom server does some over-night processing to enrich trips and to calculate new mobility profiles and personal scores, the communication subsystem takes care of downloading these data structures every morning between 5 and 6 AM or when the last download is more than 24 hours ago.

The detailed APIs that are used for this are described in sections 4.2 (personal scores) and 4.4 (mobility profile, trips).

2.3.3 Synchronous communication with the server

The communication subsystem offers a single entry point for all communication with the tripzoom server. Every time a component needs to retrieve data from the server, it can use the communication subsystem to do so. The communication subsystem takes care of signing all requests using OAuth, parsing the response and delivering the retrieve data to the calling component.

These components can then use this data as they see fit, whether they render it on screen, or store it in a local database for later use. Please refer to section 4 for examples of API calls.

2.3.4 Notifications

The last part of the communication subsystem deals with push messages or cloud messaging. On startup of the application, the application registers itself with the Google Cloud Messaging (Android) or Apple Push Notification Service (iPhone) so that it can receive push messages.

Once registration has succeeded and the registration token has been received from the Google or Apple servers, the communication subsystem informs the tripzoom server, most notably the Push Notification Dispatcher, by invoking the following call:

API Endpoint	**Description**
~api/pushnotification/rest/Client/addToken	HTTP POST Payload consists of a JSON object containing the platform (Android or iOS) and the obtained token.

Since this call is signed with OAuth and therefore includes the unique user id of the user, the server can use this information to deliver the correct notification to the correct user on the

correct device. The purpose of sending a notification is to inform users that new data are available on the server (an experience sampling request, a new incentive is available, a reward has been earned) or that a new event that requires their attention (someone invited you to become friends, a pending incentive is about to expire in a couple of hours). Upon reception of the notification, the message is either displayed to the user if the application is running (on iPhone only) or placed in the notification bar (on both platforms). Users can disable or enable push notification messages at any time by invoking the following API calls:

API Endpoint	Description
~api/pushnotification/rest/Client/disableNotifications?type=	HTTP POST The notification type can be set to allow fine-grained control of push messages
~api/pushnotification/rest/Client/enableNotifications?type=	

3. Mobile Incentive Marketplace (D4.3, QMUL)

The Mobile Incentive Marketplace (MIM) is designed to support the delivery of incentives to travellers so as to promote sustainable travel behaviours. There are four types of incentives provided by the SUNSET platform, namely, *traveller mobility pattern, targets and challenges, loyalty points and social networks*. The MIM is designed to focus on delivering incentives in the forms of targets and challenges and loyalty points by referring to users' travel routines and conditions of the transport network of interest.

3.1 Features and Modules

The MIM is the key component in delivering incentives in Tripzoom, and its operation is dependent on the central incentive management unit, Incentive MarketPlace (IMP) in which incentives are registered, managed and monitored. A key feature of the IMP is to examine a user's mobility information, his/her preference and transport network information, and identify a target or challenge that would help the user to achieve a certain goal, such as avoiding travelling in rush hour. The MIM is designed to optimise the delivery of such target or challenge in terms of the time and the right users. In addition, the MIM acts as a gateway between the users and Tripzoom, as users' responses to an incentive is directly fed back to the system via this module, providing crucial statistics for the evaluation of the incentive of interest.

The overall features of the MIM can be summarised as follows:

1) Incentive management
   - To release and notify mobile users new incentives

2) Monitoring travellers
   - To monitor historical trips status including travelling cost, co2 emission, travelling distance and travelling modality
   - To examine travellers' incentive execution status
   - To assess travellers' reward status

3) Issuing rewards to travellers
   - To compare traveller's behaviour with reward criteria
   - To send reward notifications to users

3.2 Data Models

Data models in the Incentive Marketplace can be categorised as incentive-centric and user-centric. Both types of data models provide the criteria used to define an incentive.

3.2.1 Incentive Data Model

Figure 1 shows an overview of the Incentive data model, and an incentive in the form of a target or challenge is described by the identity, reward measurement and trip condition modules. The identity module provides definitions of a target or challenge, ranging from the intended target groups of users and living lab of interest to start and expiry dates of such incentive. The reward measurement module provides descriptions of individual targets and

challenges, and keeps track on the value of each of the associated reward. Finally, the trip condition module stores metrics that describe individual trips, such as the distance measured and modality detected for a trip.

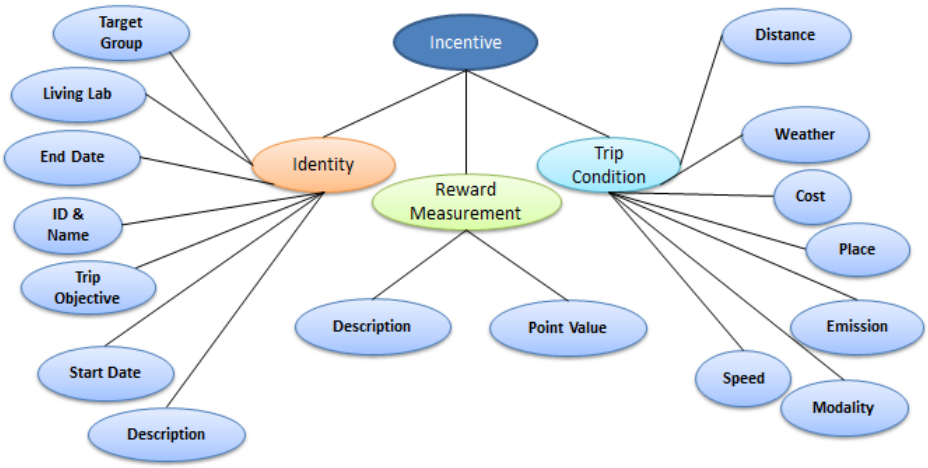


Figure 1 Incentive Data Model

3.2.2 User Data Model

Figure 2 presents an overview of the User Data model, and a user is described via the identity and trip information modules. The former consists of personal information of users with respect to their ID, location and modality profile. The latter has the same data fields as the trip condition module defined in the Incentive data model but for individual users rather than incentives.

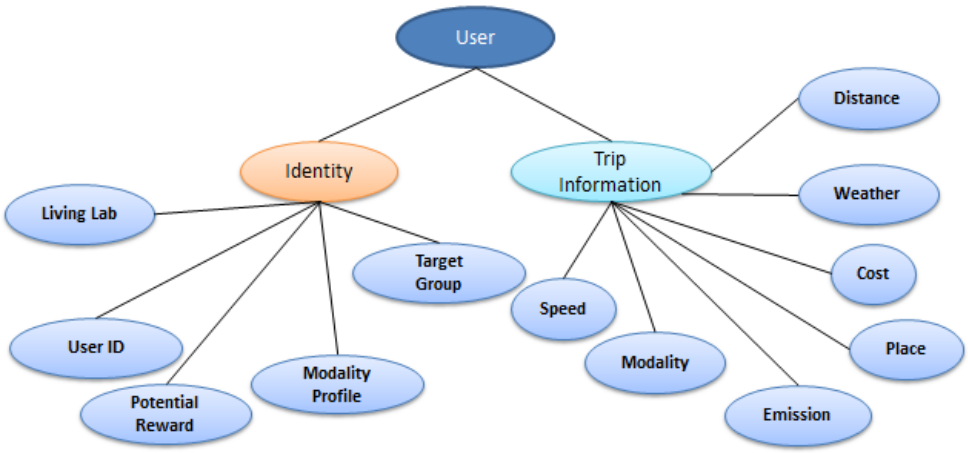


Figure 2 User Definition

3.3 Communication Patterns

Communication between the MIM and a mobile client is two-way and messages are exchanged via HTTP using RESTful Services.

3.3.1 MIM to Mobile Client (I2M)

The MIM sends the following messages to a mobile client

1. Notification of a new challenge.
2. Notification of a reward upon completion of a challenge.
3. A list of current challenges applicable to an individual user.

Data Type	API Endpoint	Description
Incentive	~api/IncentiveService.svc/DesignAnIncentive	HTTP POST, used to create and release an incentive
N/A	~api/IncentiveService.svc/tripEndTrigger/{TRIPID}/{USERID}/{TRIPENDTIME}	HTTP GET, used to check and release a reward when a trip is finished
Incentive[]	~api/GetPotentialIncentives	HTTP GET, used to get a user's potential incentive list

3.3.2 Mobile Client to MIM (M2I)

A mobile client sends the following messages to the MIM

1. Trip information of an individual user (see 2.3.1).
2. A request to find out all the current challenges applicable to an individual (see 3.3.1).

3.4 Matching Process

Matching the right group of users with the right target or challenge is a critical process that takes place inside the IMP. Such process is semi-automatic, but it also provides the flexibility for a living lab coordinator to define a target group as well as an appropriate target or challenge with respect to specific requirements.

1. **Target/Challenge validity check**. This is to check whether a specific target/challenge is active; expired targets/challenges will not be further considered.
2. **Living lab identification**. Subdivide Tripzoom users by allocating them to the corresponding living labs: Enschede, Gothenburg and Leeds.
3. **Target group identification**. User can be further divided into subgroups by referring to a range of available demographic properties including gender, age, home city and household status.
4. **Trip modality Matching**. A target/challenge is often designed to change a user's mode of transport, and modality information of a user obtained from the PMS will be compared with the modality requirements set for a specific target/challenge. A reward will be issued if the matching result shows that the user has fulfilled the given target/challenge.

3.5 Incentive Distribution

A multi-component life cycle is in place to distribute target and challenges as well as loyalty points to users, as shown in Figure 3. The IMP is a gateway between the application Tripzoom and the city dashboard which is the control interface for the living lab controllers. The IMP communicates with the mobile client through the Google Cloud for the Android implementation, and in a similar fashion using Apple Push Notification Services (APNS).

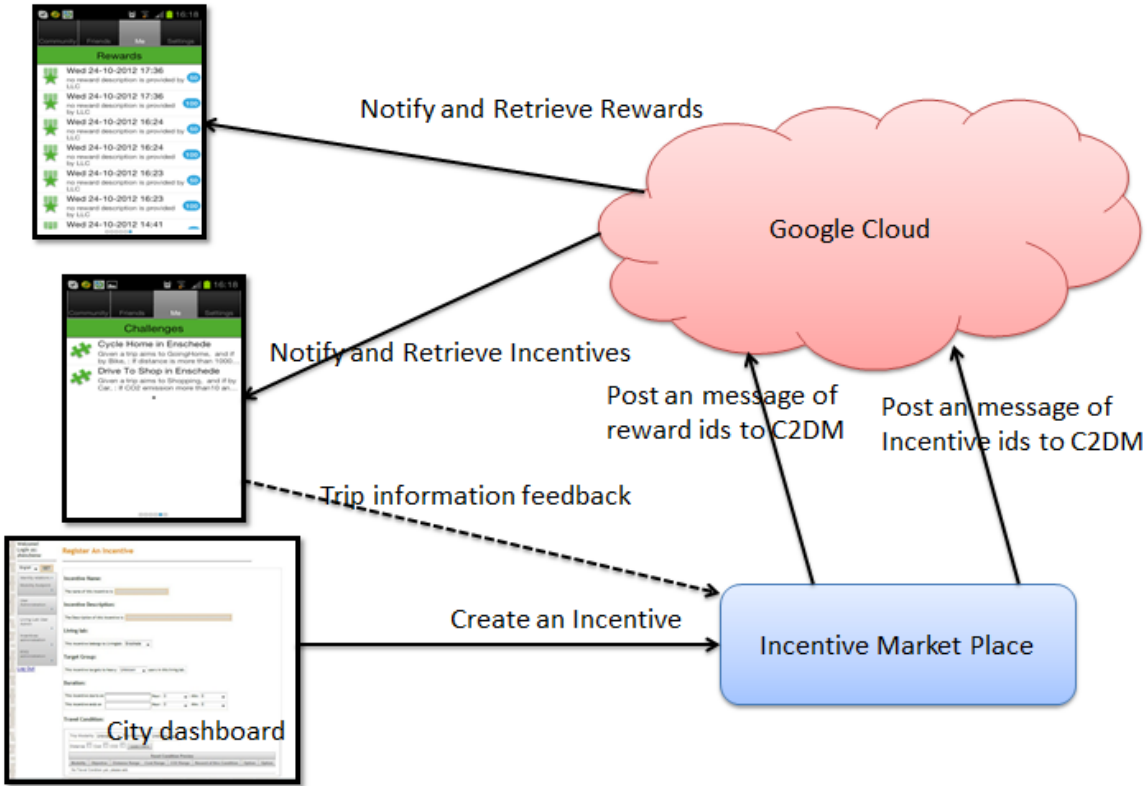


Figure 3 Distributing challenges and rewards to the Andoid tripzoom app

4. Social Mobile Application (D4.4, LOCNET)

The mobile application tripzoom can roughly be divided into two large functional blocks. The sensing module, responsible for tracking mobility patterns (as explained in Section 2), and the user interface. This section will highlight the latter, focusing on the social and mobility related features of the application. These features contribute to the SUNSET main innovations by enabling the travelers to inform and help each other to improve their commute in terms of sustainability and pleasure using social networks.

4.1 Features and Modules

Tripzoom encourages a change in travel behaviour by gaining insight to a user's personal mobility pattern. The social features put this into a broader perspective by comparing their personal mobility patterns against other users' travel patterns. This is done on three levels: First, tripzoom provides means to compare personal behaviour against the entire SUNSET community in an anonymised way (Section 4.2). Second, a social network enables users to compare their behaviour against their friends, family and colleagues. This will be referred to as the buddy list, and will be described in Section 4.3. Finally, the SUNSET system provides insight into the user's personal mobility choices, by presenting various sources of mobility statistics derived from the measurements obtained by the mobile sensing components (Section 4.4).

4.2 Community overview

Data delivering components	Measurement API
API endpoints	~/api/measurement/scores/personal
Description	The community overview provides an overview of personal mobility statistics compared against the entire SUNSET community.
Visualisation & user interactions	A configurable set of measures is provided for the community overview, expressed on a scale from 1 to 10. Each measure is visualised using a set of images, expressing the range of values from negative to positive (i.e. 1 to 10). Moreover, a tag line provides additional insight into the performance by providing a textual explanation of the expressed value.

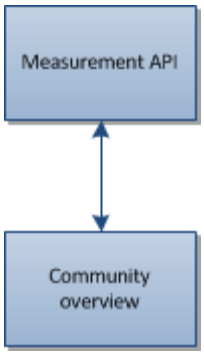


Figure 4: Interactions with community overview

4.3 Buddy list

The buddy list provides the client side visualisation of the social network within the SUNSET project. The buddy list is responsible for visualizing other users' user profile and mobility related characteristics (Section 4.3.1) and to support searching and connecting to friends (4.3.2).

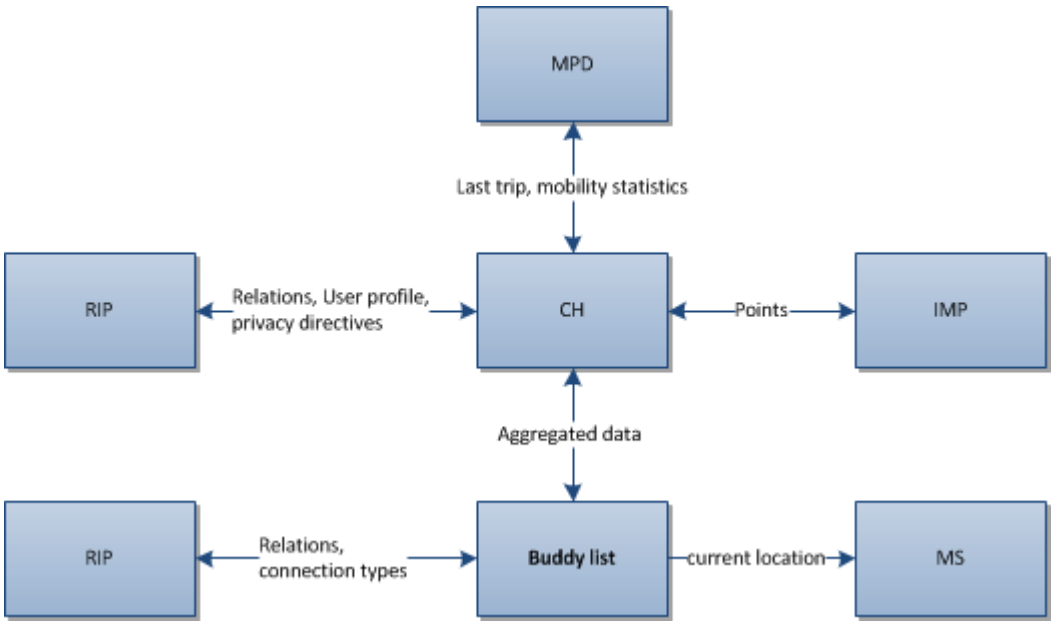


Figure 5: Buddy list interactions

4.3.1 Buddy list visualisation

Data delivering components	CH (Context Harvester)
API endpoints	~/api/harvester/overview/?me=<true/false>
Description	The buddy list retrieves its data from the Context Harvester (CH), an aggregating component that alleviates the large amount of API calls otherwise needed by the mobile client in order to construct this list. The CH consumes information from the following API's: • RIP: Relations (friendships), user profiles, privacy directives • PMS: Last trip, modality statistics (emissions, travel costs, etc.) • IMP: Points earned This information is aggregated into a single API method by the CH. With query parameter *me* the user is included in this response. The distance based sorting is supported by the mobile sensing component, which delivers the user's current location. In addition, the search functionality of the buddy list is supported by the RIP.
Visualisation & user interactions	The buddy list is visualized as a list of friends which allows sorting based on emissions (CO2), distance (supported by the mobile sensing component) and points earned. Buddy detail pages show more detailed information like mobility statistics (total travel time, cost,

	emissions etc) and details on a user's last trip. By including the user itself in the buddy list, the user can compare itself to other users on these criteria.

D2.2 [1] further specifies the functionality of the Context Harvester.
D2.3 [2] further specifies the functionality of the RIP.

4.3.2 Searching and connecting with buddies

Data delivering components	RIP
API endpoints	Search: ~api/Identity/user/property?searchTerm=<searchTerm> Connect: ~api/relation/connections/<toUserId>/<relationType>
Description	The RIP supports connecting to users by means of adding relationTypes. While friendships in this system are reciprocal, a user is only connected to another user when the friendship status is confirmed from two sides.
Visualisation & user interactions	Searching and connecting users is performed by utilizing the search functionality in the buddy list. The user has the option to select users in this result set, on which the application adds a relationType on the user's approval.

4.4 Personal Mobility Visualisations

Personal mobility visualisations on the mobile client can be separated into 4 components with their respective data types on the SUNSET platform: the mobility profile (Section 4.4.1), trails (Section 4.4.2), trips (Section 4.4.3) and places (Section 4.4.4). Figure 3 highlights the interactions with the data delivering components. Figure 3 highlights the interactions between these visualizing components and the server-side components.

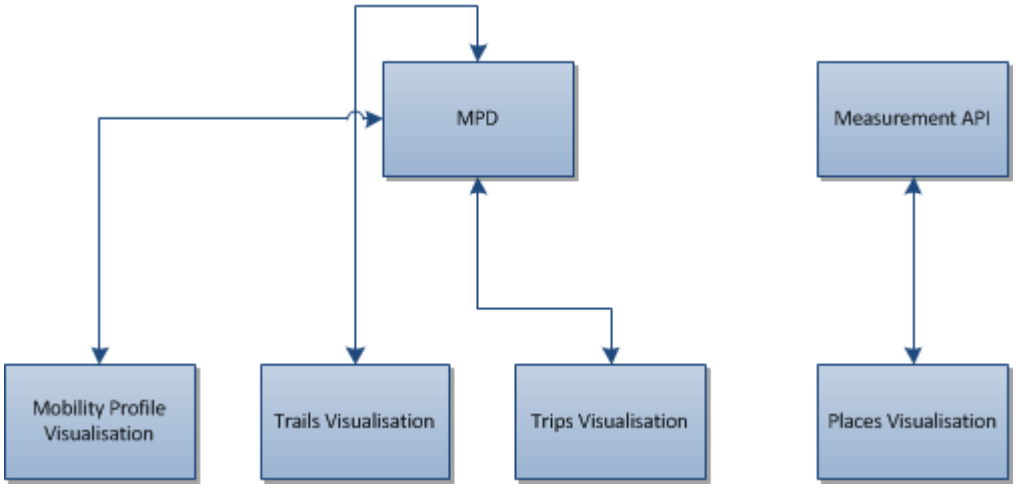


Figure 6: Personal mobility interactions

4.4.1 Mobility profile visualisation

Data delivering components	PMS
API endpoints	~api/personalmobility/mobilityprofilesummaries/?start=&dur=&dur=&dur=
Description	The mobility profile presents a modal split calculated over a configurable time span (i.e., 2 weeks, 1 months, 6 months) based on a number of sorting criteria.
Visualisation & user interactions	The mobility profile visualisation is presented as a sortable list of modalities with their corresponding percentages of travel. Sorting criteria include travel cost, time, distance and emissions per modality, where the the user can choose to split these modalities according to criteria as rain, rush hour travel, fog and temperature based comparison. The time span is also configurable.

D2.2 [1] further specifies the data model regarding mobility profiles.

4.4.2 Trails visualisation

Data delivering components	PMS
API endpoints	~api/personalmobility/mobilityprofilesummaries/?start=&dur=&dur=&dur=
Description	Trails (or: Frequent trips, regular trips) are daily travel patterns computed by the PMS as part of the mobility profile. Trails can be seen as trips between places (Section 4.4.4).
Visualisation & user interactions	Trails are presented as a list in the application which is sorted by how frequent the system has detected the trail. Detail pages also show average travel times and the fastest travel time.

D2.2 [1] further specifies the data model regarding mobility profiles.

4.4.3 Trips visualisation

Data delivering component	Measurement API
API endpoints	Trip list: ~api/personalmobility/TripSummaries/<limit>/?start=<starttime>&stop=<stoptime> Trip detail: ~api/personalmobility/TripDetails/<tripId>
Description	The PMS transforms a set of location measurements into trips and computes and derives mobility related information, thereby also map matching these measurements yielding a geometry that closer resembles the path taken for map visualisation. The trip detail request is used to retrieve this geometry.
Visualisation & user interactions	The trips visualisation splits the user's trips into an overview of last week's trips, last month's trips and older trips. Every trip has a detail page, enabling the user to correct the modality, the trip objective (such as going home, going to work, etc.) and to approve or reject a trip. Further selection enables the user to visualize this trip on the map, showing the estimated start and stop place as well as the map matched geometry.

4.4.4 Places visualisation

Data delivering component	Measurement API
API endpoints	~api/measurement/Places/excluding/None
Description	Places are calculated as part of trip and mobility profile detection, where places are the foundation for trails.
Visualisation & user interactions	The places list visualizes a user's frequent places, ordering those by percentage time of stay. Place detail pages allow editing the place name and type, which are initially derived on the server using Foursquare data.

D2.2 [1] further specifies the data model regarding place detection.

5. References

[1] Koolwaaij, J. et al. (2011). D2.2 Mobility pattern detection and visualisation service. Internal document available for the consortium and EC.
[2] Holleis, P. et al. (2011). D2.3 Relation and Privacy Management Service. Internal document available for the consortium and EC.
[3] Jacobs, K. et al. (2011). D2.1 Mobility sensing and experience sampling services. Internal document available for the consortium and EC.
[4] What is ANT+?, http://www.thisisant.com/consumer/ant-101/what-is-ant/