WP5 – END-USER APPLICATIONS

D5.2.1 –

End-user applications techniques and tools (initial)

Due date: 30.09.2014
Delivery Date: 6.10.2014

Author(s): Antti Nurminen (AALTO), Stefan Schaffer (DFKI),
Annapaola Marconi (FBK), Giuseppo Valetto (FBK)

Partner(s): AALTO, DFKI, FBK
Editor: Antti Nurminen (AALTO)

Lead Beneficiary of Deliverable: AALTO

Dissemination level: Public
Nature of the Deliverable: Report

Internal Reviewers:
Benjamin Dittwald (Fraunhofer), Mika Vuorio (CGI), Vesa Meskanen (CGI)
EXECUTIVE SUMMARY

Work Package 5, End-user Applications, aims at delivering applications for people on the move, featuring personalisation, gamification and advanced graphical interfaces, to leverage low carbon ways of travel. We attend both the planning phase with pre-experiencing, and the contextual real-time situation on the road. We have previously defined use cases, requirements and ideas for implementation specifications (D5.1). In this document, we present intermediary progress of work towards these use cases.

We have created a general routing concept and are progressing in creating a lightweight real-time adaptable multimodal Open Source routing solution. We have set up a mock-up route of virtual mobility with 360° photo and 3D visualisation methods. We have matured a gamification engine for general use, and developed a demonstrator of advanced graphical interfaces with mobile augmented reality and 3D maps, based on a functional optimisation and transmission pipeline from static and real-time data sources up to a mobile device. We have developed higher level application user interfaces for our use cases. Existing apps have been extended for STREETLIFE and new apps developed, such as a mobile parking availability app.

Disclaimer: This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 608991.

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

© Copyright in this document remains vested with the STREETLIFE Partners
D5.2.1 – End-user applications techniques and tools (initial)

Table of Contents

EXECUTIVE SUMMARY ................................................................. 3
ABBREVIATIONS .............................................................................. 6
EXPLANATIONS FOR FRONTPAGE........................................................ 7
PARTNER ......................................................................................... 7
LIST OF FIGURES ............................................................................. 8
LIST OF TABLES ............................................................................... 9

1. INTRODUCTION ............................................................................. 10
   1.1. STREETLIFE ARCHITECTURAL PLACEMENT ........................................ 11
   1.2. GOALS AND CONCEPTS ...................................................................... 12
   1.3. TASKS .............................................................................................. 12
   1.4. MAIN USER CATEGORIES .................................................................. 12
   1.5. OUTLINE OF THE DELIVERABLE ..................................................... 13

2. INTERMODAL PERSONALISED TRAVEL ASSISTANCE AND ROUTING (T5.1) .......... 14
   2.1. DESCRIPTION OF A GENERAL CONCEPT FOR PERSONALISED ROUTING AND TRAVEL ASSISTANCE ............. 15
       2.1.1. Basic routing process ................................................................. 15
       2.1.2. Personalised routing ................................................................. 16
       2.1.3. Travel assistance ...................................................................... 19
   2.2. STATUS OF BER TRAVEL ASSISTANCE AND MULTIMODAL ROUTING SERVICE .................................................. 21
   2.3. STATUS OF CGI/TRE TRAVEL ASSISTANCE AND MULTIMODAL ROUTING SERVICE ............................................. 23
   2.4. STATUS OF AALTO/TRE TRAVEL ASSISTANCE AND MULTIMODAL ROUTING SERVICE ........................................ 24
       2.4.1. Goals of AALTO’s Open Source routing solution ........................................... 24
       2.4.2. AALTO’s Open Source routing solution plan ............................................ 25
       2.4.3. Current status of AALTO’s Open Source data compression for routing .............. 26
       General Transit Feed Specification (GTFS).................................................. 29
       2.4.4. Current status of AALTO’s Open Source routing solution ................................ 30
   2.5. STATUS OF ROV TRAVEL ASSISTANCE AND MULTIMODAL ROUTING SERVICE ........................................ 31

3. VIRTUAL MOBILITY (T5.2) .................................................................. 34
   3.1. A MOCK-UP VIRTUAL MOBILITY ROUTE ........................................ 35
   3.2. IMPLEMENTATION AND VALIDATION PLAN ......................................... 37

4. CITIZEN PARTICIPATION AND GAMIFICATION (T5.3) ............................................ 38
   4.1. DESIGN OF GAMIFICATION SUPPORT ........................................... 38
   4.2. GENERAL CITIZEN PARTICIPATION ................................................ 42

5. ADVANCED GRAPHICAL INTERFACES (T5.4) ...................................................... 43
   5.1. SPATIAL INTERFACES ...................................................................... 44
       5.1.1. Augmented Reality ..................................................................... 44
## 5.2. FIRST USE CASES FOR MIXED REALITY

### 5.2.1. 3D Maps

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
</tr>
</tbody>
</table>

### 5.2.2. End-user applications techniques and tools

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
</tr>
</tbody>
</table>

### 5.2.3. Architecture of the AALTO Platform for advanced graphical applications

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
</tr>
</tbody>
</table>

### 5.4. CURRENT IMPLEMENTATION STATUS OF 3D MAPS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
</tr>
</tbody>
</table>

### 5.5. CURRENT IMPLEMENTATION STATUS OF AUGMENTED REALITY

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
</tr>
</tbody>
</table>

### 5.6. IMPLEMENTATION AND VALIDATION PLAN

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
</tr>
</tbody>
</table>

## 6. MOBILE APP DEVELOPMENT (T5.5)

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
</tr>
</tbody>
</table>

### 6.1. TAMPERE APPS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
</tr>
</tbody>
</table>

#### 6.1.1. CGI’s Tampere Pilot app

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
</tr>
</tbody>
</table>

### 6.2. BERLIN APPS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
</tr>
</tbody>
</table>

### 6.3. ROVERETO APPS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
</tr>
</tbody>
</table>

#### 6.3.1. ViaRovereto – extended

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
</tr>
</tbody>
</table>

#### 6.3.2. Bike Sharing app

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
</tr>
</tbody>
</table>

#### 6.3.3. Parking availability mobile app

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
</tr>
</tbody>
</table>

## 7. CONCLUSION AND NEXT STEPS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
</tr>
</tbody>
</table>

## APPENDIX A: LITERATURE

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
</tr>
</tbody>
</table>

## APPENDIX B: REQUIREMENTS AND IMPLEMENTATION IDEAS FROM D5.1

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
</tr>
</tbody>
</table>
**ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Augmented Reality</td>
</tr>
<tr>
<td>AV</td>
<td>Augmented Virtuality</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the Consortium (including the Commission Services)</td>
</tr>
<tr>
<td>BER</td>
<td>STREETLIFE Berlin-Pilot</td>
</tr>
<tr>
<td>D</td>
<td>Deliverable</td>
</tr>
<tr>
<td>DoW</td>
<td>Description of Work</td>
</tr>
<tr>
<td>FP7</td>
<td>Seventh Framework Programme</td>
</tr>
<tr>
<td>FLOSS</td>
<td>Free/Libre Open Source Software</td>
</tr>
<tr>
<td>GTFS</td>
<td>General Transit Feed Specification</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IJP</td>
<td>Intermodal Journey Planner</td>
</tr>
<tr>
<td>IPR</td>
<td>Intellectual Property Rights</td>
</tr>
<tr>
<td>MGT</td>
<td>Management</td>
</tr>
<tr>
<td>MR</td>
<td>Mixed Reality</td>
</tr>
<tr>
<td>MS</td>
<td>Milestone</td>
</tr>
<tr>
<td>OS</td>
<td>Open Source</td>
</tr>
<tr>
<td>OSS</td>
<td>Open Source Software</td>
</tr>
<tr>
<td>O</td>
<td>Other</td>
</tr>
<tr>
<td>OTP</td>
<td>OpenTripPlanner</td>
</tr>
<tr>
<td>P</td>
<td>Prototype</td>
</tr>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PM</td>
<td>Person Month</td>
</tr>
<tr>
<td>R</td>
<td>Report</td>
</tr>
<tr>
<td>ROV</td>
<td>STREETLIFE Rovereto-Pilot</td>
</tr>
<tr>
<td>RTD</td>
<td>Research and Development</td>
</tr>
<tr>
<td>RTPIS</td>
<td>Tampere region passenger information system and journey planner</td>
</tr>
<tr>
<td>TRE</td>
<td>STREETLIFE Tampere-Pilot</td>
</tr>
<tr>
<td>VR</td>
<td>Virtual Reality</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modeling Language</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>Y1</td>
<td>Year 1</td>
</tr>
</tbody>
</table>
EXPLANATIONS FOR FRONTPAGE

Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Partner(s): Name of the partner(s) whose employee(s) the author(s) are. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author – if he has generated the Foreground - as well as an Editor – if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the STREETLIFE DOW. The lead beneficiary partner should be listed on the frontpage as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: These should be a minimum of two persons. They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically this review includes: Identifying typos, Identifying syntax & other grammatical errors, Altering content, Adding or deleting content.

PARTNER

Fraunhofer Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
FBK Fondazione Bruno Kessler
SIEMENS Siemens AG
DFKI Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
AALTO Aalto University
DLR Deutsches Zentrum für Luft- und Raumfahrt
CAIRE Cooperativa Architetti e Ingegneri - Urbanistica
Rovereto Comune di Rovereto
TSB Berlin Partner for Business and Technology
Tampere City of Tampere
Logica CGI Suomi Oy
VMZ VMZ Berlin Betreibergesellschaft mbH
**LIST OF FIGURES**

Figure 1. STREETLIFE functional blocks .................................................. 10
Figure 2. Functional elements of STREETLIFE Apps. .................................. 11
Figure 3: The concept of travel assistance ....................................................... 16
Figure 4: The process of personalisation with 3rd party routing ............................ 18
Figure 5: The process of personalisation integrated in routing .............................. 19
Figure 6: The control mechanism ...................................................................... 20
Figure 7: Confirmed rerouting ........................................................................... 21
Figure 8: Automatic rerouting ............................................................................ 21
Figure 9: Tampere Pilot Travel Assistance and Multimodal Routing Service .......... 24
Figure 10: Tampere region that is covered with local public transportation .......... 27
Figure 11: Structure of the OSMSqueeze map data ........................................... 28
Figure 12: OSMSqueeze process stages ............................................................. 29
Figure 13: 360° cameras. The Ladybug (left) and consumer level Ricoh THETA (right) 34
Figure 14: A 360° photo from Tampere in rectangular (left) and spherical (right) form 35
Figure 15: Immersive head-mounted display Oculus Rift (SDK v2) for consumers .... 35
Figure 16: A mock-up Virtual Mobility visualisation of a multimodal route in Tampere 36
Figure 17: the design of the STREETLIFE gamification solution ....................... 40
Figure 18: basic gamification concepts and plugins ........................................... 41
Figure 19: Internal elements of the gamification engine ...................................... 42
Figure 20: State-of-the-art in real time traffic services: Lissu Traffic Monitor ........... 43
Figure 21: Sensor based mobile augmented reality (left; from [17]) and computer vision based mobile AR with a fiduciary marker (right; from [18]) ........................................ 45
Figure 22: A 2D map (left; ©City of Tampere) and a 3D map of the same area on the STREETLIFE Mixed Reality app prototype ................................................. 46
Figure 23. Ambiguous bus platforms – unmarked in Lissu service (left) but physically present (right). Do buses stop on them or not? ...................................................... 47
Figure 24: The Tre-D model of Tampere city center (up) with details added for STREETLIFE

Figure 25: The AALTO Back End architecture for advanced graphical interfaces

Figure 26: 3D map pointing out a bus stop

Figure 27: Tampere 3D map with real-time tracked buses

Figure 28: Pointing out a proximal bus stop with Augmented Reality

Figure 29: Pointing out a distant bus stop, when it is visible (upper) and occluded (lower)

Figure 30: Augmented reality for bus labelling. Bus number 21 has been overlaid onto the video feed based on real time vehicle tracking and spatial sensing of the viewing device

Figure 31: A worst case scenario for sensor based augmented reality. Due to occlusions, misalignments and latencies in tracking, augmented labels are more confusing than helpful

Figure 32: STREETLIFE Berlin mobile app standard features

Figure 33: ViaggiaRovereto mobile app standard features

Figure 34: ViaggiaRovereto mobile app - STREETLIFE features

Figure 35: Bike Sharing mobile app

Figure 36: Parking availability mobile app

LIST OF TABLES

Table 1. Streetlife multimodal routing implementations

Table 2. Schedule compression. Sizes expressed in MB, compression factors for 30-day schedule data
1. INTRODUCTION

Work Package 5 focuses in development of STREETLIFE end-user applications in a research-oriented and user-centred manner. Our end users are mainly individuals, ranging from local people, travelling daily between their home and work, to casual visitors, experiencing a new environment. We assume that our travellers have a variety of travelling means available to them, and require our applications to provide multimodal alternatives.

STREETLIFE end-user applications will aid end users in mobility planning, pre-experiencing routes and act as online travel assistants with real time features. The design and development of these applications will be driven by the key motivation of encouraging end-users participation and energy-efficient/carbon-low behaviours. To this extent, the work package will develop participation and gaming techniques that will be exploited within the mobile applications for engaging end-users in creating and sharing their mobility information and experiences, and to set-up and manage green mobility incentives and rewards.

In the general STREETLIFE functionality (please refer to D2.2.1 for STREETLIFE architectures), this deliverable focuses on the green upper right hand side of Figure 1.

STREETLIFE end-user applications are suited for the three main sites: Berlin, Tampere and Rovereto. Each city has its own specific features and available data, which affect the resulting end-user applications. The maturity of components that were developed during the first project year varies as well, depending on the baseline, ambition level, available data and other factors. For example, an existing multi-modal travel assistance app ViaggiaRovereto has been extended at ROV with high level features such as sustainable trip recommendations, while a new low-level core function, a lightweight Open Source routing solution based on OpenStreetMap and Open Data, was born for TRE’s real time travel assistance.

![Figure 1. STREETLIFE functional blocks](image-url)
1.1. STREETLIFE architectural placement

Figure 2 presents the functional elements of STREETLIFE apps, described as Generic App Functionality, Trip Planning, Multi-Modal Routing, Personalisation and Gamification. This view aggregates both site specific and general functionalities utilised in STREETLIFE apps into a single schema. Below, we describe selected functionalities.

![Generic App Functionality includes methods to infer and manage connections between mutually important entities (Interest Matching) or, in the context of maintenance of a dynamic real-time situation, Interest Management. Using this functionality, a Context is created for and propagated to every entity in an efficient, compact manner. This Context includes the current state of the environment (such as traffic flow for each road segment) and states of other entities that have been chosen to be Interesting (such as the current location of the next bus). The state, including input from Trip Planning, Gamification and Personalisation is Visualised in the apps, varying from traditional graphical user interfaces and maps to mixed reality representations.

Trip Planning includes various mechanisms to support users before and during a trip. It utilises Personalisation and Multi-Modal Routing to present suggestions to users, with Gamification elements to set up incentives and goals. On the other hand, these functionalities consider factors such as carbon footprints and real-time situation of traffic (Multi-Modal Routing), historical trip data and profiles (Personalisation) and the persistent state of the Game.

Communication between functional blocks is synchronous or asynchronous, depending on the situation and available communication methods. Both request/response (GET) and publish/subscribe (PUSH) paradigms are supported where applicable.
1.2. Goals and Concepts

This Deliverable presents the work toward supporting our Use Cases by presenting the current status of each Feature. We focus on the highlights and the effort put forth by each partner.

1.3. Tasks

WP5 is divided to five Tasks, which set up the main technologies for the foreseen application features:

1. **Intermodal Personalised Travel Assistance and Routing** sets up a common toolset for personalised travel assistance and routing algorithms. This toolset is the basis for advanced context-aware real time assisting features supported with multimodal routing that can proactively aid the end user before and during a trip.

2. **Virtual Mobility** seeks to lower the threshold of using alternate and greener means of travel via visual pre-experiencing of the suggested routes.

3. **Citizen Participation and Gamification** provides development of engagement techniques that will a) facilitate end-users with tools for creating and sharing their mobility information and experiences, and b) increase enjoyment of greener means of travel via games, implemented with a customisable gamification engine.

4. **Advanced Graphical Interfaces** advances the visualisation of traffic, routes and virtual mobility via mixed reality techniques, namely a) 3D virtual environments and b) augmented reality.

5. **Mobile App Development** integrates selected techniques, algorithms and engines from the four previous Tasks into actual applications, with advanced user interfaces and personalisation features.

Tasks 1-4 form the basis for the actual integrated application as fundamental features. These features depend heavily on available static, real time and crowd sourced data (T3.1, T3.2, T3.3). This data dependency also leads to site dependency, for any localised data. For generally available data, such as OpenStreetMap, a routing solution can be a general feature. However, even in this case, data quality may vary, and lead to site specific differences in quality of service.

The last task (5), **Mobile App Development**, utilises the developed enablers along with available data to form actual mobile apps.

1.4. Main User Categories

In our case descriptions, we will assume the following basic user types to be experimenting and trying out our end-user applications:

1. **Local people.** As STREETLIFE aims at reducing transport related carbon emissions with multimodal journeys offered for individuals, our main focus group is the local people, who could potentially alter their daily method of travel permanently. Locals know their environment, and have possibly accustomed to their habits, such as driving to work by car. If we could affect even a small fraction of people in this focus group, results would be visible.
2. *Visitors.* Easy access to transport in cities makes *visitors* our second focus group. With simple, straightforward and visually compelling mobile applications, we encourage them to try public transportation.

As our work package is research oriented, we will perform focused studies for the effectiveness of the advanced interfaces. We will also assume that some of our end users would belong to the sub-group of *early adopters*, willing to try out leading edge applications.

### 1.5. Outline of the deliverable

The deliverable reports the status of Work Package 5, Tasks T5.1-5 after one year of development. The basis of this work is in the *use cases, requirement ideas and implementation ideas* defined in D5.1. We provide an implementation plan for our components and early validation of selected ideas, where available.
2. INTERMODAL PERSONALISED TRAVEL ASSISTANCE AND ROUTING (T5.1)

Intermodal personalised travel assistance and routing is a core functionality needed for the end-user applications developed in STREETLIFE. Fundamental requirements and use cases for multimodal routing as well as for personalisation have been defined in D5.1. Starting from these ideas we develop concepts for the required components and implement first instantiations. Section 2.1 begins with a general description of an adaptable concept for personalised travel assistance and routing. The following sections 2.2 to 2.5 describe the current implementations on pilot level. In STREETLIFE, we will have four different routing services, each with its own characteristics and motivation. Table 1 briefly outlines fundamental characteristics and goals.

Table 1. Streetlife multimodal routing implementations

<table>
<thead>
<tr>
<th>Service developer</th>
<th>Site</th>
<th>Characteristics and goals</th>
</tr>
</thead>
</table>
| DFKI              | BER   | • Integration of available VMZ routing services  
|                   |       | • Majority of services are server side, lightweight client  
|                   | (Sect. 2.2) | • Research focus in intelligent user interfaces and personalisation  
|                   |       | • Gamification and crowd-sourcing integration |
| CGI               | TRE   | • Built on existing technology  
|                   | (Sect. 2.3) | • Highly robust, large scale long term commercial use  
|                   |       | • Improved with real-time data to enable personalised travel advice and park and ride |
| AALTO             | TRE   | • Created as a new service  
|                   | (Sect. 2.4) | • All internal features such as routing graph management fully available for STREETLIFE  
|                   |       | • Lightweight design, client side solution as target  
|                   |       | • Research focus in real time data management  
|                   |       | • Open Source, Open Data (OSM, GTFS) |
| FBK               | ROV   | • Routing service based on open source OpenTripPlanner journey planner and its REST APIs\(^1\)  
|                   | (Sect. 2.5) | • Server-side processing of routing requests  
|                   |       | • Integration with Android mobile app ViaggiaRovereto\(^2\) developed by SmartCampus.  
|                   |       | • Itinerary monitoring  
|                   |       | • Integration of crowdsourced information  
|                   |       | • Integration of real-time information (e.g. public transportation delay notices) |

\(^1\) For details go to: [http://www.opentripplanner.org/](http://www.opentripplanner.org/)

\(^2\) [http://www.smartcampuslab.it/viaggioovereto/](http://www.smartcampuslab.it/viaggioovereto/)
2.1. Description of a general concept for personalised routing and travel assistance

At first it will be outlined how an intermodal journey planner in conjunction with a personalisation concept and a component for travel assistance build the basis for the emerging mobility service. After that we give a more detailed description of the personalised routing concept and the mechanisms for travel assistance.

2.1.1. Basic routing process

2.1.1.1. Concept and Components

As a part of the concept we propose 3 fundamental mechanisms: an intermodal journey planner (IJP), a personalisation plan, and a travel assistance plan.

*Intermodal journey planner* (IJP): Intermodal journey planning means that travellers are provided with itineraries involving two or more means of transportation. The itinerary does not necessarily have to be intermodal which means that unimodal suggestions can also occur. Our approach allows both the integration of 3rd party services or own journey planner modules. 3rd party software may be used, if appropriate routing solutions are already available. Further development can then also take place within the other concepts or components. Some 3rd party IJP, integrating personalisation options like e.g. “finding the shortest, cheapest or fastest connection”, already exist. Using such a black box system can minimise implementation expense and allow for other priorities within the routing concept. If available IJPs do not fulfil provided requirements an own implementation may be necessary. Some examples for the different reasons for own implementations are:

- The necessity to modify the underlying algorithm independently
- Real time updated traffic situation not supported by available solutions
- Personalisation options should directly affect the weights of the graph during routing
- Client side routing
- Efficiency reasons

*Personalisation plan*: Personalisation means to make personal *mobility recommendations* out of *routing proposals*. Routing proposals represent the output of the IJP. Mobility recommendations can e.g. be generated by filtering and reordering of routing proposals. In general, we distinguish between two personalisation approaches, namely pre- and post-personalisation, which can be used individually, in combination or partly integrated in the routing process. In subsection 2.1.2 we will describe differences in the personalisation plan depending on the usage of 3rd party routing or an own routing service in more detail.

*Travel assistance plan*: Travel assistance means to define, control and adapt currently taken routes. In this sense the plan concerns the entire routing process, ranging from the specification of travel preferences to the arrival at the final destination. Two fundamental concepts of travel assistance are the monitoring and the adaption of routes. A brief outline of the whole process is provided in the next section.
2.1.1.2. Process

The travel assistance plan encompasses the whole routing process. A concrete realisation depends on many specific matters regarding implementation, device and integrated services. We will therefore describe a rough concept illustrated in Figure 3. The process starts with the definition of a route. A travel assistance system should aid the user in finding the right route according to the user’s needs. Apart from the usual parameters such as origin, destination and travel time, there are a number of other options (e.g. real time data, car or bike sharing, etc.), which can be considered to find a suitable route. A high amount of available data and intermodal travelling pose challenges for an easy to use travel assistance user interface, as the complexity of the UI usually increases with the number of possible options. Aiming at the definition of routes by using an intuitive UI, we recommend focusing on specific user groups, an app purpose and the intelligent fusion of several parameters into individual UI elements. The options, also reflecting the personalisation possibilities of the user, are gathered into a request that is processed by the IJP. The list of mobility recommendations represents the personalised system response. Once the user has decided for a particular route by selecting one of the provided itineraries, the define part of the process is completed.

A control process monitors the selected route. Information such as real-time data and current crowd sourcing messages are matched with the route during the monitoring. If a disturbance is detected, a re-routing can be initiated by an adapt process. In subsection 2.1.3 we describe the control process and adaption concepts in more detail.

![Figure 3: The concept of travel assistance](image-url)

2.1.2. Personalised routing

The fundamental difference between the two routing concepts presented in this section is that 3rd party routing can easily be realised as a client server service, whereas the presented concept for an own service rather refers to an implementation on a single device. Both concepts disclose advantages and disadvantages. According to the prevailing circumstances one of the approaches should be followed.

The processes are described on three layers. The information layer deals with the information presented by or to the user. The service layer describes underlying services involved in process. A data layer indicates type and characteristics of the considered data.
2.1.2.1. Personalisation with 3rd party routing

Personalisation with 3rd party routing will be used for the BER and ROV pilots as well as for the TRE development by CGI. A 3rd party router is able to handle route requests with the aid of different kinds of knowledge sources. For an IJP system we assume, that at least information for several means of transportation is considered for routing. Some 3rd party systems also integrate further data like e.g. past and real time information about road traffic. Within Figure 4 the IJP can be understood as a black box that receives its configuration from a pre personalisation component that handles the route request. With pre-personalisation, route requests may encompass already some personal preferences; common among those are the desired means of transportation, prioritisation criteria such as shortest, fastest or cheapest routes, or special information for users who are bicyclists. Other kinds of personalisation are of course possible, depending not only on the content of the user profile, but also on the set of options and parameters that the 3rd party router can accept for route requests.

The router output, the routing proposal, is a list of calculated itineraries. For the implementation of routing services for specific applications often so far not integrated data are necessary. We address this issue by taking into account further data after the actual generation of routes. The routing proposal can further be processed by a post personalisation component. The general request is therefore divided into two parts, the already described routing request and an additional personalisation request. The personalisation request, as a configuring input of the post personalisation component, determines the kind of data and how data are being involved. The general aim of this process is to make mobility recommendations from routing proposals. Mobility recommendations are the output of the post personalisation component and can be understood as routing proposals enriched with additional information. The main functionality of post-personalisation is reflected by reordering and filtering of itineraries, according to the included data. This data may include personal preferences (possibly a wider array of preferences that what could be conveyed to the router by the pre personalisation component); it can additionally include historical information e.g. of itineraries accepted and liked by the user; and the same post-processing mechanisms can be leveraged to account for real-time information (e.g., weather), crowd-sourced information (e.g., messages and advice about traffic or road conditions), etc.

Further – for the scope and goals of STREETLIFE – incentive mechanisms can intervene to further post-processing, besides and beyond the facets mentioned above, which leads to the production of mobility recommendations. The goal of incentive mechanisms, such as those offered via gamification, is to promote specific mobility behaviour and possibly induce modification of users’ habits, such as the exploration and acceptance of more sustainable modes and option of transportation.

Personalisation and incentives promoted via gamification are highly contextual, since they must keep into account the current state of the gamer/user, as well as the best strategies for her progress within the game logic and strategies. For example, multi-modal routes including bike segments could e.g. ranked higher among the mobility recommendations, if only a few bicycle ride points are missing to reach the next level, or award in the game; or, certain routes can be highlighted and recommended because they yield specific achievements like badges, point bonuses, etc.
2.1.2.2. Personalisation integrated in routing

AALTO will use personalisation integrated in routing for the TRE development (Figure 5). The concept is used, if an IJP is realised as a new service under the control of the STREETLIFE developers. The fundamental difference to personalisation with 3rd party routing is that post personalisation is not necessarily required, as the router is not a black box. All desired and available data can directly be interfaced. Users’ routing demands are joined together into one request, containing the route and the personalisation information. The handling of this request and the configuration of the router is still necessary. Pre personalisation can therefore still be employed and is generally performed the same way like in personalisation with 3rd party routing.

All desired data, including personalisation and gamification information, is directly used for the calculation of itineraries in a single pass. Thus, a further processing of routing proposals is not required, and a post personalisation process is not considered, since the original router output directly constitutes the mobility recommendations. Another fundamental difference is that the routing process is not finished until the destination is reached, or the process is cancelled. The selected route is instead permanently recalculated according to the current situation, by constantly updating the weights of the routing graph, for example as incoming real-time or crowd-sourcing information comes in. In principle both approaches can integrate the same data.
2.1.3. Travel assistance

As outlined in Sect. 2.1.1 travel assistance means to define, control and adapt routes. In this section the monitoring and the adaption of routes will be described in more detail.

2.1.3.1. Monitoring of routes

As described above the user gets a mobility recommendation from the system as a response to the routing demand. The mobility recommendation is usually a list, containing more than one itinerary. Figure 6 depicts the control mechanism. Assuming that one of the recommended itineraries fits reasonably well to the users’ demands, the user will select that itinerary. Once a route has been selected it is monitored by the control mechanism. The control mechanism permanently updates its current state, according to the desired real time data (e.g. weather or traffic) and crowd sourced traffic information. The system state is constantly correlated with the monitored route. The actual position of the user can also be utilised. Information that is no longer relevant for the routing, as it refers to a part of the itinerary that already has been completed, can then be discarded. If the correlation results in an impaired itinerary the adaption of the route is triggered.
2.1.3.2. Adaption of routes

After the system recognised that the monitored route is impaired, an adaption has to be performed. Our concept foresees two ways for the adaption of routes: confirmed rerouting and automatic rerouting.

**Confirmed rerouting:** In many use cases the user’s device is not visible as it is in a handbag or pocket. The rerouting notification is typically displayed in the GUI and is possibly unnoticed. In order to inform the user that a rerouting notification has to be handled a warning strategy like auditory feedback over headphones or tactile feedback over actuators has to be implemented. In a system with confirmed rerouting the notification has to be viewed by the user, as user input is needed. Typically presented options are:

- Cancel: stops the routing.
- Ignore: goes on with the current routing.
- Adjust: begins a new routing from the current position to the given destination.

After adjustment the routing process starts all over again. Confirmed rerouting will be used for the BER and ROV pilots. The concept is sketched in Figure 7.
**Automatic rerouting**: This concept is integrated in most navigation systems. It is used if the distraction of the user (typically a driver) should be minimised and if the device is in the visual range of the user. In most existing systems the user is informed via a spoken or a graphical message. The adjustment of the itinerary starts automatically. Automatic rerouting will be used for AALTO and CGI developments for TRE. The concept is sketched in Figure 8.

### 2.2. Status of BER Travel assistance and multimodal routing service

As outlined in Error! Reference source not found., the BER pilot app will be implemented as a lightweight client integrating an available 3rd party route planner from the Verkehrsmanagement Zentrum Berlin (VMZ). Even though the IJP is already available as a server service, it is still under further development. For the BER pilot the user should be encouraged to use the bicycle. Therefore the integration of respective features of the bicyclist community is planned. The concrete requirements as a basis for a specification of the modifications have to be defined within the next steps of the extension of the journey planner.
The app should also provide useful mobility recommendations if the use of the bicycle is not an option for the user due to differing requirements. Public transport and car routes will therefore be supported as well.

The personalisation plan for BER follows the concept presented in Section Error! Reference source not found.. The aim is to provide the user with mobility recommendations generated from the routing proposals that are delivered by the router. The IJP is configured via pre personalisation. The user is able to choose desired means of transportation, cost functions and routing constraints. Cost functions enable the user to set an emphasis for duration, length, CO₂ emissions and monetary costs of the routes. Via routing constraints it is possible to state a maximum distance for each transportation mode. In this way a user can e.g. indicate that routes should comprise a maximum walking distance of 1,000 meters.

With respect to post personalisation, the BER routing service will allow the user to choose how the results of the IJP output should be rated and ordered, based on the following considerations:

- greenest (minimising the amount of CO₂ emission)
- fastest
- shortest
- cheapest
- with the least number of mode changes

Regarding the planned integration of bicyclist features a number of further rating and reordering opportunities will be taken into account. A focus will be set on the consideration of “safest” bike routes. Beyond these post personalisation options that are derived from the user’s preferences contained in the route request more post personalisation opportunities will arise within STREETLIFE. For the BER pilot we plan to integrate the following:

- post personalisation considering crowd-sourced traffic messages. If itineraries are impaired by crowd-sourced messages about traffic disruptions, routing proposals may be filtered respectively.
- post personalisation considering the user’s individual route history. The users previous choices (recorded by STREETLIFE) can be taken into account for filtering and ordering of routing proposals.
- post personalisation considering the user’s gamification information. The actual state of the user within defined games may have impact on the rating and ordering of routing proposals.

The mentioned post personalisation mechanisms require software components (e.g. crowdsourcing data mining and gamification engine), which need to be developed and deployed beyond Y1 of the project. We plan to use and evaluate the described concepts in the second BER pilot iteration.

The travel assistance plan for the BER pilot includes monitoring and adaption of selected itineraries. The current solution allows the user to start a companion mode that (in preparation of server side monitoring) saves the current route on the STREETLIFE server. During the
companion mode the selected route and the current position are displayed on a map within the BER pilot app.

The current service provides only basic mechanisms needed for the travel assistance functionality described in Section Error! Reference source not found.. The integration of the control mechanism is planned. Thereby a server side list of “active” routes should include the itineraries that are currently taken by users. Another server side list of traffic messages has to be integrated. The traffic messages list should be feed by various data sources including crowd-sourced and real-time traffic information. For traffic messages resulting from crowd-sourcing a reliability checking strategy has to be defined. Possible approaches could be:

- automatic checking
- manual checking by the mobility management or
- user assigned checking

The “active” routes and the traffic messages have to be correlated. If routes are being impaired by traffic disruptions, the server side adaptation mechanism will send a notification to the user’s device. It is planned to implement an adaptation mechanism according to the concept of confirmed re-routing described in Section Error! Reference source not found.. The features of the travel assistance will be further developed within STREETLIFE, and the BER pilot app will use and evaluate it during the second iteration of the project.

2.3. Status of CGI/TRE Travel assistance and multimodal routing service

The Tampere pilot travel assistance and multimodal routing service builds on the investment the City of Tampere already has made in their Intelligent Transport Systems. The Tampere STREETLIFE pilot will utilise Tampere region passenger information system and journey planner (RTPIS) and enhances it with integration of real-time data feeds.

Multi-modal router will be able to route all transit modes taking into account real time events regarding bus arrival estimates and delays or cancellations while providing travel assistance. In addition the real-time router will check in park and ride mode the availability of parking places in park and ride locations. This processing of real-time data in the multimodal routing service is pre-processing as described in the blueprint architecture where the router can be affected almost on the fly. The multi-modal router keeps all the needed static data in memory structures for fast access. The real-time feeds sent updated information to these memory structures meaning routing algorithm can take into account if bus has been delayed and user will get up-to-date information for their trip itinerary. In the park and ride mode it is important that we guide people to the locations where parking is available. Therefore the real-time router will check the parking availability during the route itinerary calculation.

Real-time monitoring and tracking are key elements of travel assistance. They will gather all real-time events like vehicle location, user location and crowd sourcing data. If the user has enabled the tracking of their route itinerary, they will get notifications and updates regarding their trip. Tracking and Notification are context aware services. This component enables personalised travel assistance to the passengers.

The work carried out is mostly in integrating real-time data flows into the multimodal router, and its configuration. The goal is to get car drivers to use more public transport means.
The TRE pilot travel assistance offers citizens’ safe, personalised, and real-time routing solutions covering all modes of transport to achieve the best experience. Multi-modal integration of park and ride will also check availability of parking slots as part of the trip suggestion and will calculate the CO₂ footprint. Carbon footprint calculation is post-processing. One of the additional aspects in the TRE pilot are open multi-modal router APIs for third party access further promoting an active developer community in Tampere.

2.4. Status of AALTO/TRE Travel assistance and multimodal routing service

2.4.1. Goals of AALTO’s Open Source routing solution

AALTO’s routing solution aims to provide an Open Source multimodal routing service that is based on Open Data. As a research effort, it is developed from scratch with the intention of allowing anyone to improve any detail of the algorithm and utilise it in any service. The design goal is to be lightweight in all resource usage, including computational and data requirements. It will depend on efficient filtering and encoding of OpenStreetMap data and other relevant data sources. It will support real time adjustments of the underlying routing graph, therefore directly supporting personalised route requests and immediate adaptation to events that might affect the current route.
2.4.2. AALTO’s Open Source routing solution plan

AALTO’s travel assistance features will rely on the openness of the routing solution. The core cost function can be manipulated to yield basically any alternatives, as long as these alternatives can be defined in a feasible and reliable manner in numbers. For example, consider the cost of ‘greenness’. If walking is considered to cause zero emissions, the greenest result would always be just walking unless there is a similar alternative, such as bicycling, which would then be the greenest solution. Similarly, how much cost does mode switch induce? Should such annoyance be personalised or an attempt made to provide a reasonable default? There are no limitations to pre or post personalisation in Aalto’s solution per se. In general, we will search optimal solutions for these numbers to provide a common trip configuration, for example with a selection of preferred mode, incorporation of personalised data such as speed of bicycling, recovery of previous trip parameters and other higher level features.

Real-time data will be received as 1) events that may cause re-calculation of the current route and 2) entity updates such as floating GPS tracking. Events may be provided by crowd-sourcing mechanisms or by other external sources, defining for example traffic jams, accidents or other factors that may affect routing. Entity updates can be handled in two ways: a) for transportation vehicles, schedules will be kept up-to-date and deviations may launch re-calculation of routes; b) for travelling individuals, the data is handled as floating GPS data, which can be used to infer current traffic flows on given road segments. Again, once deviations from previous state exceed a pre-determined threshold, updates will take place and potentially trigger route re-calculation. In AALTO’s implementation, all these updates will be directly reflected in the actual memory structures that hold the route graph.

In the first phase, the routing solution will reside at server side, where all relevant data is directly accessible. Deviations in public transportation and other events will all be updated in the centralised memory of the router and a straightforward polling scheme from the client side checks for changes. In the second phase, the routing solution will be ported to Android. Here, clients can perform off-line multimodal routing based on schedule data, or when online, solve routes full real-time tracked traffic situation. The first variant with server-side solution is needed for validation and testing purposes, especially in handling real time data updates. Once the solution matures, it will be placed to mobile clients. This second variant will require a functional interest management component at server side. Interest management will then solve a set of road segments and dynamic entities that are of interest to each client on their way. This solution will combine simple radius-based events, topological distance from a planned route and relevant entities to a look-up list. When deviations happen that match with the look-up list, the client is updated on them. Route re-calculation may take place, and depending on the severity of the altered state, the user interface may simply present an updated arrival time or re-route the client completely.

The first implementation of the routing solution will be ready after the first R&D cycle of the project, with first prototype of the routing algorithm implemented at Milestone 2, followed by iterative evaluation and improvement until Milestone 3. During Phase 3, the solution will be implemented on mobile clients and higher-level application features implemented on the user interface. Again, by Milestone 5, the full version will be evaluated on the field.
2.4.3. Current status of AALTO’s Open Source data compression for routing

**OpenStreetMap**

Currently, the OpenStreetMap data filtering and compression stage is ready. The OSMSqueeze software consists of three steps which we call *pruning, serialisation* and *encoding*.

In pruning, less important geometry information is removed. First, roads not useful for routing are removed. Any part of a named road can be entered as a routing target so they are the most important to preserve. Names in OpenStreetMap are mainly attached to road geometry for cars, and in dense urban areas or along highways there is separate parallel geometry for unnamed sidewalks and pedestrian roads. Finally, many unnamed roads connect named roads between each other or public transport stops. They are all potentially necessary for finding shortest walking routes.

The purpose of serialisation is to organise data for decompression and parsing in a single pass as far as possible, with the goal of having each relationship to refer only to earlier data. For example if a road A is stored along with the information that it crosses road B which is introduced later, then the crossing has to be kept incomplete in memory and re-examined after road B is found to link them together. Avoiding such issues dictates the large scale file structure.

The actual filtering and encoding process is quite complex, with the overall approach of eliminating all repeating elements, binding close elements together or using relative coordinates and discarding unnecessary representations and descriptions that are not needed for routing. The end result is a compact, serialisable file. Figure 11 presents the file structure.

As a preliminary test case with public transportation in focus, we have compressed the larger Tampere region that is covered by local transportation services (Figure 10). Roughly 20x40km and with a 2km maximum distance to nearest bus stop, the full data set with all multimodal layers currently takes 1309410 bytes (1.3MB).

Figure 12 presents the full pipeline of *OSMSqueeze*. All stages have been implemented in Java. We are prepared to extend the software by more data types, such as parking locations and traffic lights, as need arises and data becomes available.
Figure 10: Tampere region that is covered with local public transportation
Figure 11: Structure of the OSMSqueeze map data
We use GTFS data for Tampere region. Compression is performed in two stages, each reducing the data by roughly 90%. The first stage converts source data into an intermediate format, summarising it by removing redundant information. The intermediate format is human readable text consisting of lists of names and tables of numbers. It's designed to be easy to produce and interpret in software.

The second stage compresses data into a binary format using three basic techniques. The first one is delta coding, which is applied to sequences of large numbers known to be close to each other such as departure times or latitudes of nearby bus stops. They are stored as backward differences, which are likely to require fewer digits than the original numbers. The first number of each list is stored in full, every following number as the difference from its predecessor. This process can be repeated analogously to computing higher order derivatives.

Storing \((t_0 - t_{10})\) in base 2 would take 10 numbers with 9 digits each. Same information is contained in \((t_0, \nabla t_1, \nabla t_2 - \nabla t_{10})\) which takes only 4 bits for the last 9 numbers, or \((t_0, \nabla t_1, \nabla^2 t_2, \nabla^2 t_3 - \nabla^2 t_{10})\) where the last 8 numbers are mostly zeroes.

To benefit from the reduction in bits and still be able to represent arbitrary sequences, a second compression technique is needed. A variable length code can store small numbers using fewer bits than large numbers and still distinguish numbers from each other without any marker between them.

The goal is to produce a data file that's easy to transfer as text between different platforms without character set encoding problems. Therefore, we are limited by the ASCII set to 96-97
possible values that can be encoded in a single printable byte. We omit the discussion of this optimisation challenge.

We tested the compression algorithm with public transport networks of varying sizes. Compression results are shown in Table 2. The "Zip" column illustrates sizes of original data files in GTFS format after removing line geometry information which is not included in this compression method. All files were re-compressed for an evenly matched comparison using the same program, Zip 3.0 developed by Info-ZIP.

The "Raw" column shows the size of uncompressed route, stop and schedule data. Transport agencies have different policies for the period that schedules are included and verified, so even in the same region some transport services are scheduled for one year ahead and others only for some months. This makes it hard to estimate how far into the future the entire schedule is correct because partially missing data may be waiting to be filled in or legitimately missing because a line is replaced by another in the near future. For this reason and to achieve more uniform results, we cut all schedules to 30 days before compressing. The size of this portion of original data actually included in the compressed result is included in the "30-day" column.

After compression stages 1 and 2 the data size decreases rapidly. Finally, the ratio column compares final result in the "Stage2" column to the most representative initial situation in the "30-day" column. The resulting data sets are quite reasonable for mobile transmission. For example, Tampere schedule data takes only 65kB.

Table 2. Schedule compression. Sizes expressed in MB, compression factors for 30-day schedule data.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>DATE</th>
<th>RAW</th>
<th>ZIP</th>
<th>30-DAY</th>
<th>STAGE 1</th>
<th>STAGE 2</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netherlands</td>
<td>Dec 11</td>
<td>1148.83</td>
<td>231.62</td>
<td>900.54</td>
<td>47.71</td>
<td>3.56</td>
<td>0.40%</td>
</tr>
<tr>
<td>Manchester</td>
<td>Dec 12</td>
<td>584.44</td>
<td>58.82</td>
<td>564.59</td>
<td>12.34</td>
<td>0.78</td>
<td>0.14%</td>
</tr>
<tr>
<td>Helsinki</td>
<td>Dec 11</td>
<td>543.39</td>
<td>48.60</td>
<td>480.91</td>
<td>8.67</td>
<td>0.45</td>
<td>0.09%</td>
</tr>
<tr>
<td>Tampere</td>
<td>Jul 30</td>
<td>23.79</td>
<td>3.94</td>
<td>20.40</td>
<td>1.37</td>
<td>0.065</td>
<td>0.31%</td>
</tr>
<tr>
<td>Trentino</td>
<td>Sep 17</td>
<td>7.26</td>
<td>0.83</td>
<td>3.95</td>
<td>0.48</td>
<td>0.038</td>
<td>0.97%</td>
</tr>
</tbody>
</table>

2.4.4. Current status of AALTO’s Open Source routing solution

The most important requirement for maps to be used in routing is topological correctness. This means that connections between roads are stored so that it's possible to follow any road from start to end, find intersections with other roads and continue following those roads. The OSMSqueeze ensures that the resulting data is topologically correct.

For correct pedestrian routing it's unnecessary to consider exact roads shapes except in determining the closest road to route start and end locations. Along the route decisions can only be taken at intersections and between them the only relevant pieces of information are walking distance and access restrictions.

There are many very fast and advanced algorithms for routing with cars or public transport but combining the two is difficult [1]. For cars the hierarchy of road classes is helpful.
Motorways are fastest for large distances and their network is sparse while small dense residential roads become less relevant in the middle of the route. Focusing on a small subset of the data speeds up calculation. Similarly when walking, switching to public transport is nearly always faster and the algorithm has far fewer transit network stops to consider compared to the entire roadmaps' intersections where it's possible to turn several ways.

Bicycle routing has many of the same complexities as cars because riders should obey traffic signs and speed is competitive with public transport so taking the bus or tram when possible may not be the fastest option. Additionally, the cycleway network has less obvious hierarchy and small roads with little or slow traffic may be preferable.

Dijkstra’s algorithm [2] is an effective way to find all possible routes from a single starting point. Exploring the entire road network wastes some time going in the wrong direction when the desired destination location is known, but it's a good way to take into account all the complexities of multi-modal routing [2, 3].

We have implemented Dijkstra’s algorithm for multimodal routing in C++, utilising the OSMsSqueeze data structure. We are improving it with various heuristics, following the guidelines above. We will provide benchmarks on the performance once we are satisfied with our optimisation solution. After that, we will proceed by real-time support and higher level features such as greenest route or healthiest route, once we determine related weight costs for the core cost function. At the moment, this solution can already yield over 100 routes per second in a desktop computer.

AALTO’s routing solution supports requirements TRE-RI-2, TRE-RI-3, TRE-RI-4 and TRE-RI-6. Instead of an Open API (TRE-RI-9), the whole source code will be released.

2.5. Status of ROV Travel assistance and multimodal routing service

The solution that will be used in the ROV pilot leverages – as outlined in Error! Reference source not found. - the integration between (on the client side) the “ViaggiaRovereto” Android app, which has been developed and is maintained by the SmartCampus initiative to which FBK participates, and (on the server side) the OpenTripPlanner (OTP)-based journey planner service currently hosted in the Cloud PaaS infrastructure of SmartCampus.

As the computation of mobility recommendations is concerned, the development of a full-fledged travel assistance and multi-modal routing service for ROV starting from the above follows the design described in Section Error! Reference source not found., since it incorporates a 3rd-party routing solution. With respect to pre-personalisation, the current service already accommodates the possibility by the user to choose what array of transportation means and options should be considered by the OTP-based planner. Moreover, related to the issue of pre-personalisation, the user can mark a request to the planner as a recurring trip, specifying the period or frequency of such trip. This changes the computation of the route planning service, which will output itineraries that are suitable for those recurrent specifications (for example, work days vs. weekend recurring trips are computed differently, because of the different availability of public transportation means.)

With respect to post personalisation, the current service allows the user’s choice about whether and how the set of results output by the route planner should be rated and ordered based on the following considerations:
• fastest
• safest
• healthiest (encompassing legs with most physical exercise, such as walking or cycling)
• with the least amount of walking distance
• with the least number of mode changes
• greenest (minimising the amount of carbon emission per traveller)

The kind of (both pre and post) personalisation currently in place is therefore limited to the user’s preferences, which can be either expressed explicitly, or saved as part of the user’s profile for the ViaggiaRovereto app, and used as her implicit default by the journey planner service. In the course of STREETLIFE, more post personalisation opportunities will arise; we plan to integrate the following:

• pre OR post personalisation based on user history, which is recorded by STREETLIFE and can be mined, including user’s previous choices for compatible itinerary in similar context; this personalisation requires a data mining component, which needs to be developed, and plans are to integrate it beyond Y1 of the project, and use and evaluate it in the second ROV pilot iteration;

• post personalisation based on incoming crowd-sourced information and real-time information sources, which may allow to prune the array of itineraries, for example because some travel transportation have become unavailable/available, or some roads have become blocked/usable. Crowd-sourced and real-time information of this kind is currently used, but only for itineraries that have been already chosen by the user of the journey planner, and are being monitored by the system on her behalf.

Further use of post personalisation in the planned solution for ROV will occur by means of the addition of the STREETLIFE gamification component, and the design of games that reward the user with game-based incentives when picking specific itineraries. These post personalisation options may vary, since they are necessarily contextual to the rules and goals of the specific game being developed, as well as the game state accrued by each individual user/player during game participation. In general, however, they will favour green and healthy itineraries, and will promote those user’s choices by assigning incentives (either in-game, such as points or badges, or real-world, such as mobility-related benefits or coupons) to specific itineraries.

As the monitoring and adaptation of chosen mobility recommendations is concerned, the current solution for ROV allows a user to “save” chosen itineraries for monitoring. For such itineraries, the service then provides notifications about various types of events, including delays, strikes, accidents, parking availability, road works, etc., which occur in locations that are relevant to the itinerary. These notifications originate either from an array of real-time data feeds and services on the ground (such as the information services of local and national transport operators, reports from traffic aides on the ground, etc.), or from of crowd-sourced information provided by the app users themselves. All of these information sources are integrated in the current service as data feeds that are hosted in the SmartCampus PaaS that hosts also the OTP-based journey planner service.

Therefore, the current service provides only a subset of the travel assistance functionality described in Section Error! Reference source not found., that is, only monitoring
capabilities, and only for a subset of chosen itineraries of interest, as opposed to monitoring
the current itinerary as it is followed by the user and for the duration of the trip. Moreover,
currently, incoming notifications are strictly informational, and do not trigger per se any re-
routing functionality (either automatic, or confirmed). In STREETLIFE, the travel assistance
functionality will be extended from mere monitoring towards adaptation support, according to
the concept of confirmed re-routing described in Section Error! Reference source not
found., which will be adopted and evaluated in the ROV pilot during the second iteration.
Additionally, further sources of information will be incrementally integrated, which can
output notifications of use; enhanced crowd-sourcing support will also be provided, based on
advances in crowd-sourcing validation coming from data management activities, as well as
progress in promotion of crowd-sourcing which will be enabled via gamification of this form
of user participation.
3. **VIRTUAL MOBILITY (T5.2)**

Virtual Mobility refers to the use of the new Information and Communications technologies (ICT) as an alternative to physical mobility. In our context of personalised mobility, virtual mobility allows users to pre-experience their travel plan. Virtual Mobility depends on several enablers. First, a routing solution must be present. A route should consist of start and end points, and the topology between them. Second, the route needs to be represented in a manner that conveys the key characteristics and visual properties of the route to the user. In this task, we focus on realistic visual expression with traditional 2D images, 360° spherical photos and realistic 3D models. Our case is the user at home.

A realistic 3D model, registered with detailed route data, allows most freedom to a Virtual Mobility application designer. Any viewpoint or a full ride of a planned route can be provided to the user, with the 3D model annotated with route assisting labels and markers. Further assisting features such as landmark and bus stop databases can be used to guide the attention of the user. Task 5.4, Advanced Graphical Interfaces, presents 3D virtual environments that are used within this task (see Section 5).

Where accurate, realistic 3D models are not available, we expect 2D photos and 360° spherical photos to be more easily acquired. In order for them to be useful for navigational assistance, they need to be registered to the route data as well: position and accurate orientation are essential. There are already multiple choices for consumer level cameras with internal geo-tagging capability, but only very few of them yet offer the bearing via an e-compass. However, external devices for DSLR cameras are available with such a feature. Also, mobile phones and tablets are quite often embedded with GPS and orientation sensors.

Figure 13: 360° cameras. The **Ladybug** (left) and consumer level **Ricoh THETA** (right)

Acquiring 360° (4π) photos has been earlier available only with special hardware such as **Ladybug**. However, it is now possible for consumers with the **Ricoh THETA** (Figure 13). The **THETA** records orientation, but geo-coordinates need to be added by a controlling mobile phone. Our goal here is to study the potential in low-cost acquisition of such images, binding

---


5 [http://ww2.ptgrey.com/spherical-vision](http://ww2.ptgrey.com/spherical-vision)

them and 3D models to multimodal routing solutions, and creating a compelling visualisation to the users.

A 360° photo captures the entire environment from a single viewpoint. The images are stored in rectangular form, and should be viewed with special software. Figure 14 presents a 360° photo from Tampere taken with a THETA, the start point for our case route (see Error! Reference source not found.), in both rectangular and spherical form. A well-known case of such photos is Google’s Street View service.

![Figure 14: A 360° photo from Tampere in rectangular (left) and spherical (right) form](image)

Viewing a virtual route can be implemented in traditional means with a desktop computer on its display, but we are hypothesising that an immersive view might convey the environment in a more prominent manner. Recently, consumer level immersive displays have emerged to the market, led by the Oculus Rift, now in the 2nd development version (see Figure 15).

![Figure 15: Immersive head-mounted display Oculus Rift (SDK v2) for consumers](image)

3.1. A mock-up Virtual Mobility route

To test the possibilities of Virtual Mobility, we have acquired two Ricoh THETA 360° cameras and taken a set of photos from Tampere along a route to the city centre, where the 3D model can be used for route presentation. We have also acquired an Oculus Rift SDK v2 device. We approach the challenge initially with mock-ups and technical feasibility tests.
Figure 16: A mock-up Virtual Mobility visualisation of a multimodal route in Tampere
A mock-up Virtual Mobility route from a parking lot to city center using a bus is presented in Figure 16. The top picture provides a map view of the route. The locations of assisting images are marked by red circles. Next 4 images present the route with 360° images, where a user can view them toward any direction, observing the environment. For legibility, we have cropped them to portray only the essential parts (the first assisting image is cropped from the 360° photo shown in Figure 14). The default orientations show targets and characteristic environmental features along the route. We start at a parking hall at Åkerlundinkatu. The initial orientation includes both the start point (the parking hall) and the direction toward next decision point, a crossing. At the crossing, another 360° image shows a distant bus stop as a directional cue. Next, the bus stop is shown closer by, from the viewpoint of pedestrian walkway, indicating Bus 9 as the vehicle to enter. Another distant cue is shown next, at the Hämensilta bridge, from a bus’s viewpoint. We then move to the 3D model (last row), portraying the bus stop to step out and our target, the Old Church. These views are shown from a higher elevation to enhance the spatial context. The user can smoothly move the viewpoint in the 3D environment.

A proper set of assisting images - 360° or 2D – would require a denser distribution to provide a near continuous experience in Google Street View manner. Our full set of images from the case route includes 50 360° photos and 30 traditional photos.

3.2. Implementation and validation plan

We have implemented bus stop database extraction as part of our work with GTFS public transportation data (see Sections 2.4 and 5.3). We can use this data to embed our 3D model with bus stop labels. In order to create 360° and traditional photo based bus stop guidance, we need a set of properly geo-referenced and oriented images (requirements NAV-IMG-S-*). Our goal is to create a proof-of-concept of a consumer level 360° photo acquisition process with sufficient quality.

Our indicative tests on the THETA have shown that the bearing records of images taken with it are not of sufficient accuracy for registering augmentations correctly. Similarly, GPS positioning is not robust enough. For automated labelling, both positioning and orientation needs to be more accurate. We have obtained Piksi RTK boards in hope for better positioning, and have earlier built more accurate orientation sensors in-house. We expect the related implementation work to be finished by Month 18.

We are currently implementing support for viewing 360° images as part of the work toward Advanced Graphical Interfaces (Task 5.4, Requirements NAV-360-*), expecting functional implementation by Month 18. We have obtained an Oculus Rift SDK v2 display device and have started developing support for it. We expect functional integration by Month 24.

Validation of the Virtual Mobility will involve comparison against interfaces of traditional journey planners. This work will start qualitatively with route mock-ups and individual images during the first Pilot phase. Functional testing will take place during the 2nd Pilot phase.

7 http://www.swiftnav.com/piksi.html
4. CITIZEN PARTICIPATION AND GAMIFICATION (T5.3)

To facilitate and motivate citizens to adopt the sustainable urban mobility solutions supported by the STREETLIFE project, we will introduce in the end user applications that STREETLIFE will deploy and evaluate some features, which implement incentive mechanisms, based upon the concept of *gamification*.

Gamification is a general term that indicates the “use of game design elements in non-game contexts” [4]. The non-game contexts can be very varied, ranging from user interaction with an information system, interactions between multiple users, completion of tasks at work, or in a volunteer community, or other social settings, etc. These interactions occur through – and with the support of - an ICT system that implements some form of game-like logic on top of its core business logic, and directs the end users in their gamified interactions.

We have introduced the main concepts of gamification, as it relates to STREETLIFE, in Deliverable D5.1; in the same Deliverable, we provided some examples of gamified interactions centred upon a personal intermodal personalised travel assistance and routing application. In the context of STREETLIFE, the objective of gamification is twofold: as a principal goal, we want to create personal and social incentives for users of STREETLIFE, which will strengthen their commitment to take advantage regularly and consistently of the advanced mobility solutions made available by the STREETLIFE end user applications; at the same time, we want to ensure that embracing and repeatedly using those solutions is fun and rewarding, which is likely to increase usage, and at the same time reinforce sustainable mobility behaviours in the end users population.

4.1. Design of gamification support

In the first year of the STREETLIFE project, we have developed an initial version of a gamification solution. That solution has been intentionally kept general, so that, in principle, it can be used not only to develop and support a number of different games and the corresponding incentive schemes, but also gamifies a variety of different domains, and encompass user interactions with many diverse ICT-based services. However, we have taken significant advantage of the requirements of the STREETLIFE projects with respect to citizen participation and sustainable mobility incentives for our conceptualisation and design, which enables us to deploy STREETLIFE gamified applications in the project pilots, and validate and evaluate the capabilities of our solution, based upon the project objectives.

This gamification solution has been developed as a modular system comprising several major components, with different conceptual roles:

- **A design environment**, which supports a “gamification designer” in the development and configuration of gamification mechanisms, including the game logic, incentive strategies, and paths of progression of the user/player within the game; a key requirement is that the gamification designer can use the environment through a design interface without being a skilled computer programmer in a conventional programming language; an example and a prototype of the interface of the design environment is a Web application, with a browser-based GUI that interacts with an extendible back-end, which we call the “gamification engine”.

The **gamification engine**, which represents the core of the system, and has a twofold role: at game design time, it interacts with the interface of the design environment, and supports the creation and configuration of all the elements in the game specification; at game execution time, it deploys and run one or more instances of the game that has been developed, is in charge of executing the game logic, and of computing the state of the game for each participating player; the gamification engine is built on top the popular rule engine Drools by JBoss\(^8\), and interacts with other components by exposing REST services.

- **A game persistence layer**, whose responsibility is to provide persistence for the gamification engine; it is in charge of holding and storing both the specifications of the game created via the design time component of the engine, as well as the game state at execution time.

- **An integration enabler** facilitating the integration with the gamification engine of external ICT systems, for example of mobile apps operated by players; the enabler allows the gamification designer to specify what actions, interactions and events occurring within such an external system are “gamifiable”, that is, of interest to the game logic. Technically, the enabler is implemented as a wrapper of the external ICT system (which we call “gamified app” or “gamified system” once it is wrapped) that captures any occurrences of the gamifiable actions, plus an event-based connector relaying those occurrences to the gamification engine, via a specific REST listener; this way, gamifiable actions by players, as they occur, can be processed by the game logic and trigger advancements in the game.

- **A game display and notification component**, in charge of showing to each player her progress within the game (such as her personal game state, rank etc.), as well as to communicate to the player guidance on various facets of the game and how it is going, by pushing notifications to the player’s ICT system (e.g. the player’s app). This component can be implemented either as a stand-alone component, or as part of the gamified app, using any resident notification mechanism, possibly with the support of the event-based connector mentioned above, if necessary.

The structure of the gamification solution and the components described above, with their dependencies, are shown in the UML Component diagram of Figure 17:

---

\(^8\) http://www.drools.org/
In Figure 17, the integration enabler is not visible as a single element; rather, we show its event-based connector, and we also show a gamified app, as an example, although in itself that is not part of the gamification solution proper. The gamified app has already used the wrapper, which is conceptually a part of the integration enabler, and can then communicate occurrences of gamifiable actions to the execution time portion of the gamification engine through the event-based connector. In a game scenario, multiple gamified apps and systems may interact this way with a game being executed in the gamification engine, typically on behalf of the game players.

In Figure 17, we also show the Display and Notification component, whose role is to receive updates from the gamification engine, as it executes the game logic and produces significant game advancements, such changes in the state of the game and its players. It is the responsibility of this component to make these updates available and present them, and it can do it in several ways: for example, it can send a notification to the gamified app of a specific player about her achievement of an in-game goal (such as a badge); or it can display a leaderboard, that is, the current score rank of the players involved in the game, by presenting it as a web page at a dynamic URL, and notifying that URL to the involved players. In the component diagram, for the sake of generality, the Display and Notification component interacts with the event-based connector to issue its notifications; however, depending on the nature of the gamified apps involved in the game and their availability of native notification capabilities, that may become superfluous.

Another important aspect to remark is the extensibility of the gamification engine, which is ensured by its internal plugin-based design. Both the design time and execution time sub-components of the gamification engine leverage plugins: a plugin is a small piece of software carrying the definition of a specific gamification concept and enabling 1) the gamification designer to use that concept in the design environment when constructing the game and its logic, for example writing rules that predicate on that concept; and 2) the execution engine to instantiate that concept and incorporate it in the game as it runs, for example making it part of the game state.
Good examples are the plugins for *points, badges, leaderboards, or levels*. Those are concepts that are extremely common in most games, and widely used in gamification strategies. However, the literature on games and gamification shows that there are many variations of those basic concepts, each with specific semantics, which yields different logic and behaviour. For instance, leaderboards can be simple and cumulative, or periodic (e.g. a weekly leaderboard); also, leaderboards can be defined on the whole set of players or be, e.g. social (i.e., a leaderboard computed among the friends of each given player, in accord to her social network), etc. For this reason, our gamification plugins support extension by specialisation, with a multi-level hierarchy that is rooted in a generic and abstract gamification plugin; the hierarchy can be extended in *depth* (for example, a weekly leaderboard could be seen as a variation derived from a periodic leaderboard, which in turn derives from a “standard” leaderboard concept and plugin), as well as in *breadth* (by adding new elements to the vocabulary of concepts that enrich the spectrum and diversity of games that can be constructed).

Figure 18: basic gamification concepts and plugins.

Figure 18 depicts the highest levels of the plugin hierarchy we have described above. It is noticeable that each plugin provides a service-based interface, which allows the various parts of the gamification engine to interact and manipulate the game concept that the plugin is devoted to, and the instances of that concept, by means of a limited set of exposed operations, and in accord to its specific game semantics.

The UML Component diagram in Figure 17 – for the sake of clarity and readability - does not show the internal design of the gamification engine to the level of detail that can make its plugin-based design evident. Figure 19 zooms in and offers a richer, although informal, view of the element in play within the gamification engine. In an instance of the gamification engine component, several plugins are instantiated, and the vocabulary of game concepts they manage and manipulate (the *game model*). Multiple games can be defined (and then run in parallel) within the same engine installation, based on the same game model or different subsets thereof; each game has its owner and configuration. The run-time sub-component (incorporating Drools) executes all instances of the games that have been defined, deployed and have started execution. The *model* within it contains the instances that correspond to the Game Model vocabulary of the game concepts chosen at design time and that evolve during the game based on the rules defined upon them. Execution is mainly reactive, in response to either *external events* (i.e., gamifiable actions fired by player), or also *result events*, that is, events produced as the side effect of the right-hand side of a game rule. The Figure also highlights the service-oriented interactions that allow connecting in a loosely coupled way the
Gamification Engine to the rest of the gamification solution, as well as to other parts of the STREETLIFE mobility information system.

![Gamification Engine Diagram]

Figure 19: Internal elements of the gamification engine

4.2. General Citizen Participation

While gamification methods provide the main STREETLIFE means for participatory action by citizens, other features are also developed for this end. Our passive citizens (see Section 4, D3.2.1) can act as citizen sensors, providing anonymised or personal floating GPS data for real time traffic flow updating, usable in routing services, or direct tracking with friends.

Active citizens may use STREETLIFE apps for creating events that specify static or temporary situations in traffic. Active citizens may also publish and share their experiences in the form of their recorded mobility tracks.

AALTO’s current version of the Advanced Graphical Interfaces, the mixed reality mobile app, currently supports basic passive citizen features, allowing transmission of users’ GPS updates to the Context Service in real time.

General participation methods for citizen will be presented in D5.2.2.
5. ADVANCED GRAPHICAL INTERFACES (T5.4)

In the framework of public transportation, data visualisation typically focuses in traffic management analysis using traditional 1D and 2D plots, charts and map interfaces [5-7]. On rare occasion, advanced 4D visualisation methods have been applied, for example to incident analysis, to provide a better overall situational view and management of temporal data [8]. For individual travellers, journey planner web sites with map support are nowadays a commonplace. Requirements for such planners are well known, where complexity of data, especially in the case of multimodal routing, often leads to data reduction, leaving out features such as pedestrian guidance and navigational detail [9]. However, as the area of public transportation management is maturing, advanced features have begun to emerge. One of these is accurate real time tracking, improving perceived service reliability [10]. However, visualisation of such data still relies on traditional means. For example, the Lissu real-time bus tracking service by the city of Tampere, Finland, is based on a traditional map interface, lacking small scale visual features (see Figure 20). Lissu presents bus stops with yellow dots buses with numbered and oriented blue circle. The map is here at maximum level of detail.

Figure 20: State-of-the-art in real time traffic services: Lissu Traffic Monitor

Our work for traffic visualisation aims at exactness and detail. In contrast to abstraction, we provide visualisation with full realism in real time. We support both augmented reality (AR) and augmented virtuality (AV), or, 3D Maps. Together our interfaces fill the virtual reality continuum called mixed reality. Our developments in mixed reality form the basis for AALTO’s STREETLIFE app, onto which other features such as multimodal routing are added. Our primary area is an urban environment, namely the city of Tampere, Finland, dense with visual and navigational detail.
5.1. Spatial Interfaces

Traditional spatial interfaces rely mostly on two-dimensional representations, which can be cartographically abstracted or, as is becoming a common alternative, aerial photographs. Such representations are best at displaying configurational spatial information (survey knowledge) [11], which suits for example route planning. Although early maps were populated with intuitive figures, common maps depend highly on compact symbols, annotations that require a legend, which explains the various visual conventions [12]. These conventions are often culturally bound, and visitors may need to put some effort in map reading.

When one is orienting him or herself with the world using an external information source (self-locating), he/she first needs to match cues between the source (the map) and the target (the real world). By such structure matching [13], one can project or superimpose one space on the other [14].

In an urban environment, this process is challenged by the drastically different viewpoints and view content. An abstract map or even an aerial photo do not natively share many directly recognisable visual cues with the ground level view of the observer. An aerial photo lacks building façades and their salient features. For an abstract map, the cartographer has pre-designed the set of cues suited for the expected spatial task. In the case of pedestrian navigation, street names are often given precedence. These cues may or may not be efficient, depending on context, task and situation. For example, road names need to be searched from the physical environment. After this initial search, they become more efficient during navigation.

A GPS can assist in localisation, and a magnetometer (compass) in orientation; but still, a set of matching cues are needed to establish a mental scale. For a local person, who already holds an accurate mental map and is familiar with a given representation, this orientation process is fast, if even necessary.

Modern mobile platforms provide 3D graphics hardware and spatial sensing as standard features along with the common wireless networking capabilities. The task 5.4 Advanced Graphical Interfaces exploits these properties and creates two complementary visualisations of traffic, routes and virtual mobility in general: 1) 3D virtual environments and 2) augmented reality. Together they are called mixed reality interfaces [15]. Applications that are based on the concept of mixed reality provide a basis for embedding content perceptually within the environment, either by direct augmentation (augmented reality, AR) or via a 3D model of it (augmented virtuality, AV). Both these application types use world as the user interface, thereby potentially minimising the need for spatial inferring and cue-based triangulation for simple short-distance target finding tasks

5.1.1. Augmented Reality

Augmented reality (AR) techniques enrich or annotate the real world. They are inherently egocentric. Augmentations lie either directly in the view of a user (using see-through displays) or on top of a camera view of a mobile device [16]. Due to lack of suitable display hardware, the former has given way to the latter throughout the last decade.
An AR system has three key characteristics: (a) it mixes real and virtual imagery, (b) registers the digital data to the real world and (c) provides interactivity in real-time [17]. One of the main challenges is accurate registration, where the virtual content is positioned onto the real world in a perceptually accurate manner. Two fundamental methods for this are commonly utilised: 1) sensor-based registration and 2) computer vision based registration (Fig. 2). For the first case, spatial sensing is utilised to solve the current pose of the viewer, for example using GPS and orientation sensors. For the second, the scene is computationally analysed for recognition and tracked to compensate for motion, simultaneously solving the pose.

The two approaches have their pros and cons. The sensor-based registration is globally applicable, but depends on accuracy of position and orientation sensors. The computer vision based approach could, in principle, yield high accuracy. However, data from potential targets need to be captured and pre-processed for recognition. Previous computer vision based AR systems used pre-defined fiducial markers for recognition (Figure 21, right), but recently research has pushed toward recognising markerless environments with natural feature tracking [18]. This approach has yet to mature: with less analytical and vast larger data sets, varying illumination conditions and commonly occurring occlusions such as all the construction site vehicles and other obstacles in Figure 21 (left), the challenge is immense.

Further challenges in AR are related to perception, especially the hard problem of interposition [21]. There is no depth information in the video feed of a mobile device, and when virtual content is simply rendered on top of the view, there is no cue for human vision to determine if the content is occluded or not; it appears to be in front of everything.

5.1.2. 3D Maps

Another facet of mixed reality is augmented virtuality (AV), where a virtual environment is augmented with data emerging from the real world. However, in our case, our virtual environment will represent the real world. This leads us to describe this case as a 3D map [22].

3D maps rely on 3D models, which are built to resemble the real environment. They could be abstracted with similar task and usage based goals as traditional 2D maps. As we focus on proximal tasks and visual recognition, we have set the design goal to be realism. Figure 22 presents our case area of Tampere, Finland. The traditional 2D map focuses attention to street
names and labels. The 3D map, using the Tre-D model, focuses attention to accurately textured façade surfaces (right). The detail level of the model matches with requirements necessary for direct recognition of building facades [23]. Such 3D models have proved to be easily recognisable by users [24, 25], where recognition is assisted by a range of salient cues [26].

![Image of 2D and 3D maps](image.jpg)

Figure 22: A 2D map (left; ©City of Tampere) and a 3D map of the same area on the STREETLIFE Mixed Reality app prototype

In a realistic mobile 3D map, the primary technical challenges are in rendering a potentially very detailed 3D large scale environment with limited resources. Our original research solved this problem by an elaborate optimisation scheme [27]. While the in-core resources of current mobile devices have significantly improved, the issues of scalability and efficient use of battery power remain.

In contrast to augmented reality, 3D maps do not suffer from many of the perceptual issues. Interposition is resolved with the well known Z buffer solution as part of 3D rendering. Registration is resolved automatically as the city model and augmentations maintain the same coordinate system.

Interactivity has long remained a challenge in mobile 3D maps. In an earlier experiment, 3D manoeuvring with simple hardware buttons was found cumbersome [28]. Along with spatial sensing and touch screens, 3D manipulation has now become easier. For example, interacting with a 3D map can follow the same pointing paradigm as the AR with spatial sensors [29]. Here we have found that such a method applies well to on-site browsing, while the common touch screen based pan&zoom paradigm suits off-site tasks.
5.2. First use cases for mixed reality

In Tampere region, public transportation is mostly implemented with buses. The local central hub is located in an open square, Keskustori (central plaza), with over ten bus stops and platforms. The Lissu service presents them as yellow dots on a traditional map interface. For a passenger who only knows that she should catch a bus at Keskustori, the task requires extensive browsing of the platforms. A map interface, when correctly interpreted, would assist in the task. However, on location, there is some additional ambiguity. For example, there are two physical platforms that are unmarked in the map interface. Figure 23 presents the situation. Of the three platforms on that side of the Hämeenkatu street, two are actual bus stops, but share only one unique identifier. Initial qualitative interviews support our observation that the overall task of identifying a bus stop in this environment is at least initially confusing.

To support multimodal routing in Tampere region, our interfaces should be successful in the primary tasks. We define our first two use cases in the context of public transportation to be 1) bus stop identification and 2) bus vehicle identification, and hypothesise that our interfaces should be more efficient in these tasks than an abstract map. After reaching sufficient technical maturity in the interfaces, this will be our first focus of validation.

The primary use cases will be extended validated later, when our other application features and enablers mature (T5.1, T5.2, T5.3) and have been integrated to the advanced graphical interfaces to form a full-fledged app (T5.5).

5.3. Data processing and architecture for mixed reality

The STREETLIFE Mixed Reality application depends on both static and real-time data. These data sets include OpenStreetMap, GTFS and the SIRI public transport tracking data. Deliverable D3.2.1, Mobility Data Integration Models and Techniques (initial), describes these data types and formats. We also need textured 3D city models.
5.3.1. Static data processing

The Mixed Reality application utilises OpenStreetMap data as the fundamental spatial data set for anything that moves. The data itself may not be directly visualised, but for example, real-time tracked entities such as public transportation vehicles are mapped unto it. The OSMSqueeze process (described in Section 2.4) parses the data, filters it and encodes it. Public transportation information, such as bus routes and bus stops, described in General Transit Feed Specification, is parsed and filtered for only relevant data, encoded and stored in the same data containers.

The currently available Tampere 3D model is the Tre-D model from the 3D City Info project [24]. It covers the main city centre, the densest area of traffic in Tampere, including the square, which is Tampere’s main bus hub. The model is provided in the VRML format and consists of 194 buildings and 260 unique facade textures. Textured areas cover mostly building facades around the central square, Keskustori, and around the main road Hämeenkatu. Figure 24 presents the Tre-D model with Details such as bus stops and the bus cover added for STREETLIFE. The bus models are not part of the static model, but present real-time tracked vehicles. Options to acquire larger models are being investigated.
The Tre-D model is pre-processed to optimise it for progressive transmissions and viewing in mobile devices. The pre-process pipeline consists of 5 phases:

1. VRML parsing and reformatting
2. Texture processing
3. Visibility determination
4. Visibility list encoding
5. Packaging data onto cache files

The resulting data set includes binary geometry with bounding boxes, surface normals and dominant colours that are inferred from textures, a set of level-of-detail textures and a compressed visibility list that provides the potentially visible pieces of geometry from any point within a volume surrounding the 3D model.
The OpenStreetMap data parsing, filtering and compression has been implemented in Java together with the FP7 EU project CultAR. GTFS data parsing and integration with OpenStreetMap data has been implemented in Java within STREETLIFE. The 3D model preprocess pipeline is based on AALTO’s earlier work on 3D map data processing, with improvements for STREETLIFE.

5.3.2. Real-time data processing

Real-time bus locations from Tampere are available in the SIRI Lite format encoded in JSON (please refer to D3.2.1 for more details). The feed covers the local area public transportation with approximately 200 vehicles. Query parameters are not supported and instead the interface provides a single file containing vehicle monitoring responses for all vehicles in circulation. The file changes once per second with an additional latency of one second before location observation updates from GPS receivers in the vehicles are centrally received and written into the file.

The SIRI GPS position updates are raw data, with variation in accuracy. Direct mapping to nearest OSM street segments did not yield satisfactory visual quality due to warping and jumping of the positions, especially at crossings. We therefore map these updates to the OSM data using pre-generated GTFS based bus routes. Currently, most of the vehicles can be reliably mapped to actual locations along the route data. Unfortunately, there may be multiple routes for each route number (depending on time of day and other factors), and the SIRI feed does not fully specify the route of each vehicle. However, the feed contains information fields intended for local people, describing areas that are along the route. This information does not match with actual start and end stop information, and a separate inference phase is required. Implementation of this phase and improvement of its accuracy is ongoing. This detail is dependent on the SIRI data provider and is not an inherent problem of the format itself.

5.3.3. Architecture of the AALTO Platform for advanced graphical applications

Figure 25 presents AALTO’s research architecture for the advanced graphical applications. Static data sets are parsed and optimised into binary form and placed available in the main Back End server. Real-time data in the SIRI format is parsed and mapped onto pre-defined bus routes along OSM roads in the Data Correction stage. This stage also creates and maintains a virtual dynamic entity in an internal Server Adapter for each individual vehicle in the SIRI feed, where position is part of the entity state. These entity state updates are then forwarded to the Context Service using local binary sockets.
Context Service is the main operational Back End component, which manages both static and real-time data, holding a dynamic World State in efficient memory structures. It handles all communications to mobile and desktop clients using a tokenised binary XML protocol, supporting request/response and publish/subscribe paradigms. It can progressively transmit textured 3D models and update the localised real-time state of the mobile clients, pushing updates based on Interest Management.

Interest Management optimises transmissions of real-time data based on inferred interests of each mobile client. Currently, this component uses a straightforward spatial data culling method. Later, the component will utilise potential visibility, road network topology, general Euclidean vicinity and explicit lists such as named friends to form a real-time data propagation table for each mobile client.

5.4. Current implementation status of 3D Maps

At the Back End side of the STREETLIFE Mixed Reality App, we have a functional static data processing pipeline consisting of the OSM and GTFS optimisation pipeline, and the 3D map data optimisation pipeline. We have a functional real-time process for handling public transportation tracking of Tampere via SIRI interface. These data sets are fed to Context Service, which is implemented with C++ and running on a Linux server.

At the Front End side, we have an Android implementation of a 3D map of Tampere, supporting real-time tracked public transportation visualisation. Our client device is ASUS T1701T tablet, based on the Tegra 4 chipset, with a 10” touch screen and a 2560x1600 screen resolution. The app runs at 30fps, portraying tracked 3D buses within the Tampere 3D model.

Figure 26 presents a screenshot of the 3D map, pointing out a particular bus stop, our first use case. As the first attempt at guiding users’ attention, we are using simple arrows hanging
above the target. This visualisation may change. Figure 27 presents a screenshot of our second case, bus tracking, and an on-site photo of the app.

The 3D user interface is a 3D navigating interface. One can orientate in the model either by using touch screen margins, or in an embodied manner, orienting the device similar to one holding a video camera. Translation is implemented with panning on the touch screen. Navigation is possible in the entire 3D space of Tampere city centre, from ground level to sky. Collision avoidance keeps the viewpoint outside of the buildings.

Figure 26: 3D map pointing out a bus stop
5.5. Current implementation status of Augmented Reality

The STREETLIFE Mixed Reality app supports augmented reality (AR) on the same client as the 3D map. One of our goals here is to discover the feasibility of currently available technology in sensor based tracking. In our special case, both the client and target may be mobile, causing further temporal requirements than in a common situation where targets are static. Our implementation therefore relies in the same real-time tracking information as the 3D map, together with spatial sensing of the device, combining its internal GPS and orientation sensor data.
Currently, we support two augmentation targets: 1) bus stops and 2) buses. The use cases here are spotting the correct bus stop and identifying the appropriate bus from any direction. Figure 28 presents a case for a proximal bus stop. As the platform is visible, it is decently clear what the target is\(^9\). In Figure 29, we present two cases of a more distant bus stop. In the upper figure, the platform is visible, although it is oriented 90° to the camera, causing difficulties in identifying it. In the lower figure, a bus occludes the view, and identification becomes nearly impossible, unless one already knows of the existence and location of the platform. Basically one would easily assess that the arrow is pointing at the bus (see Figure 30, where this assessment would be correct). These cases are static. With accurate tracking of the mobile client, motion parallax can assist in assessing the distance to the arrow.

Figure 30 presents a situation, where a bus has stopped and both bus tracking and mobile client tracking match quite well, making bus identification easy. Note that from this direction, no physical bus number is visible and such an interface could be of benefit. Figure 31 presents a worst case situation, with a range of tracking failures. The bus in the centre is not tracked at all. The next bus, 12, is tracked, but with a latency and orientation error. Two buses, 2P and 28, have already left but their tracking has not caught up. Several distant buses are tracked (5, 12, 13) but occluded. On the right, two buses (28 and 12) are on their stops to the right, but occluded and with dislocated annotations.

\(^9\) The screenshots are from a field experiment pilot version of the app. “thumb up” and “thumb down” markers are used by subjects to indicate their opinion of the representation – can they identify the target or not.
Figure 29: Pointing out a distant bus stop, when it is visible (upper) and occluded (lower)
Figure 30: Augmented reality for bus labelling. Bus number 21 has been overlaid onto the video feed based on real time vehicle tracking and spatial sensing of the viewing device.

Figure 31: A worst case scenario for sensor based augmented reality. Due to occlusions, misalignments and latencies in tracking, augmented labels are more confusing than helpful.
5.6. Implementation and validation plan

Our current advanced graphical application is a technical demonstrator of 3D maps and augmented reality with full data processing and transmission pipelines. These two interfaces are combined in a single mixed reality app on an Android client. At this time, no higher level application features are present. Also, there are several detail level issues that need to be solved, as well as improvements on 3D manoeuvring. Visualisation will also be subject to change, with a range of variations. These issues will be researched simultaneously with the first TRE Pilot phase within WP5.

Higher level features will be implemented by the end of the second project year, simultaneously increasing the level of testing, from perceptual to usability and ability to leverage greener travel behaviour.

6. MOBILE APP DEVELOPMENT (T5.5)

STREETLIFE mobile apps provide the fundamental tools and user interfaces for on-the-fly mobility management and assistance. Integrated back end services feed the mobile apps with full support for personalised, intermodal journey planning and on-the-fly situational awareness, where the app can alert the user on accidents with automatic re-routing suggestions. The app will also keep the users aware of their current level of ‘greenness’, guiding users constantly toward energy efficient mobility.

STREETLIFE apps will be developed on commonly available platforms such as Android phones and tablets. They combine features from tasks T5.1-5.4 into actual mobile apps, also utilising components emerging from T3.2, T3.3, T3.4 and T3.5. Depending on available data on each site (Tampere, Rovereto, Berlin) and other site specific issues, these applications are localised to some extent.

6.1. Tampere apps

There will be two branches of apps in Tampere, CGI’s industry app and AALTO’s research app.

6.1.1. CGI’s Tampere Pilot app

The Tampere pilot travel assistance and multimodal app builds on the investment the City of Tampere which has already set-up their Intelligent Transport Systems. The Tampere STREETLIFE pilot will utilise the Tampere region passenger information system and journey planner (RTPIS) and enhance it with the integration of real-time data feeds in AALTO’s Tampere Research App.

AALTO’s STREETLIFE app is a research prototype, which will be validated as part of WP5. This validation is incremental and will happen in parallel with CGIs Tampere Pilot. The focus of development is not in traditional user interface features, which are well developed in other apps, but in assessing the potential of advanced graphical interfaces.
AALTO’s mobile app will be built on top of the mixed reality app, adding features and components from Tasks 5.1, 5.2 and 5.3 as they mature. By Month 18, we expect multimodal routing and basic Virtual Mobility technologies to be integrated. By Month 24, we expect higher level user interface features to be present, supporting these enabling technologies.

Section 5 provides the description of AALTO’s current app. The basic use cases\(^{10}\) AGI-3D and AGI-AR are now supported for limited experiments. For 3D maps, requirements NAV-3D-R-1 – 6 have been implemented. R-7, attention management, is partially ready with simple markers. Specifications NAV-3D-S-1, S-2, S-5 and S-6 have been followed and realised with minimal deviations (i.e. we have obtained Tegra 4 based tablets, which only support Wi-Fi but not 3G/4G networking). Specifications NAV-3D-S-3 and S-4 have been followed with full data processing pipeline in place, although the multimodal routing engine (S-4) has not yet been ported to Android. The quality of most requirements needs to be assessed and in some cases improved.

3D Interaction feature specifications (NAV-3DUI-S-*) have been followed. Implementation ideas NAV-3DUI-S-2, S-6 and S-7 are functional.

For augmented reality, requirement NAV-AR-R-2 (annotating) has been met. Specifications NAV-AR-S-1 (bus route and bus stop data) and S-2 (SIRI tracking) have been followed and are ready. S-4 and S-5 (suitable mobile platform) has been acquired and related software implemented as planned. S-3 and S-6, routing and user interface for routing, will be implemented as task T5.1 yields suitable software.

AALTO’s Tampere research app follows the general requirement ideas APP-1 – 4.

The success on providing sufficient navigational support for requirements NAV-GEN-*, NAV-KNOW-*, NAV-UI-*, NAV-OBJ-* and NAV-ANN-* needs to be assessed during the 1st and 2nd Pilot phases.

6.2. Berlin apps

The use cases defined for the Berlin pilot depend on functionalities, requiring the development of a new mobile app. The so far developed Android prototype is available for members of the STREETLIFE consortium and for participants of the evaluations. The Berlin STREETLIFE app is a multi-modal trip-planning app aiming at integrating transport-related data and services available in Berlin and promoting sustainable as well as energy efficient mobility.

Besides the core feature of multi-modal trip-planning, special requirements of the bicyclists community (i.e. safe bike routing, consideration of weather and route characteristics), gamification aspects and advanced mobility features (i.e. crowd sourced traffic messaging, route monitoring) have to be considered. During the development phase of the first iteration of the project a subset of these features has been implemented.

\(^{10}\) Please refer to APPENDIX B: Requirements and implementation ideas from D5.1 for the use cases and requirements. AGI = Advanced Graphical Interfaces, NAV = Navigation.
So far the app allows planning multi-modal personalised trips, adjusting individual user profiles as well as, selecting a trip for monitoring and trip history logging. Specifically, the following features have been developed:

- **Trip, profile and preferences configuration**: to enable the input of information that is relevant for the intermodal journey planner (IJP) a user interface has been developed. The user is able to state origin, destination and, travel time. Further editable profiles and indication of preferences (i.e. duration, distance, monetary costs, emission) is supported (Figure 32, top left & bottom right).

- **Itinerary search**: the IJP of the BER partner VMZ has been integrated via an interface provided by the City Intelligence Platform (CIP) of SIEMENS.

- **Itinerary selection**: the proposed itineraries are listed within the app (Figure 32, top middle), the user can view details for each of the trips, using a list view or a map view, and a companion mode can be activated for a selected trip (Figure 32, top right).

- **Green leaves calculation**: in preparation for the integration of extended gamification features green leaves are calculated from the Carbon emission information for each itinerary. The number of earned green leaves has been integrated in the graphical user interface, as a measure of “green mobility” (Figure 32, top right & bottom left).

- **Google profile login**: in order to make the user identifiable for the STREETLIFE system the user can login using a Google profile. A login is required to enable the companion mode. In order to pseudonymise the user information the Google ID is hashed before it is sent to the STREETLIFE system (Figure 32, bottom middle).

- **Server profile creation**: if user information of an unknown user is sent to the STREETLIFE system, a new server sided user profile is created.

- **End user license agreement**: according to German law the user has to accept the end user license agreement, before the app can be used.

- **Trip history logging**: if a trip is selected for monitoring (companion mode), information concerning this itinerary is sent to the STREETLIFE system. The information will be used for advanced mobility features, and for evaluation purposes.
6.3. Rovereto apps

The use cases defined for the Rovereto pilot in the first project iteration required the development of three new mobile apps, to cover new sustainable mobility services provided by the city (i.e. eMotion bike sharing service) and advanced features provided by the STREETLIFE framework (i.e., real-time mobility data management, mobility management dashboard, gamification), and a significant extension of the existing trip planning mobile app, to guide users toward energy efficient mobility.

6.3.1. ViaggiaRovereto – extended.

**ViaggiaRovereto** is a multi-modal trip-planning and travel assistance app freely available on Google Play aiming at integrating all transport-related data and services available in Rovereto and at promoting sustainable and energy efficient mobility behaviors.
VaggiaRovereto mobile app allows to: plan multimodal personalised trips (by foot, car, public transport or with shared mobility services such as car or bike sharing); monitor user’s usual routes and receive notifications in real time of any delays or problems; consult up-to-date information on public transport timetables, urban viability and parking availability; report inconveniences and problems encountered during the trips by foot, car, public transport or with shared mobility services.

Within STREETLIFE, the app has been significantly extended to promote sustainable transport solutions and to increase the user awareness on the real value of its transport choices (i.e. in term of cost, time, and carbon footprint).

In particular, the following features have been developed (see Figure 33):

- **Sustainable trip recommendations**: when planning a trip, the app suggests the user the best trip solutions with respect to the urban mobility policies defined by the municipality. These trip solutions are highlighted and shown before the itinerary solutions identified by the trip planner according to the user preferences (e.g. transport mode, minimum time, minimum changes). This functionality exploits the new features of the *Intermodal Journey Planner (IJP)* developed by FBK within T5.1 (see Section Error! Reference source not found.).: the possibility of defining mobility policies and of injecting them in the IJP to suggest sustainable itineraries.

- **Real value of a car trip**: when presenting a trip itinerary by car the app shows 1) a tentative amount of time for finding an on-street parking in that area; 2) the parking cost per hour. This information should make “park and ride“ solutions more attractive, since the parking cost is typically lower and the time for the “ride“ is most of the time similar to the time spent searching for an on-street parking. This functionality relies on historic data about on-street parking occupancy that is collected through the “Parking availability“ mobile app (WP5) and aggregated through the *Data Management Rovereto sub-system* built on top of FI-WARE technology (WP3).
- **Trip monitoring and re-planning**: the app allows to 1) save trip itineraries, 2) receive notifications about inconveniences and problems related to the trip, and 3) re-plan the trip whenever necessary. Trip notifications are related to real-time mobility data and in the moment include real-time parking availability, bus and train delays, and eMotion bike availabilities. Real-time data are handled by the Data Management Rovereto subsystem built on top of FI-WARE technology (WP3).

![Image of ViaggiaRovereto mobile app](image.png)

**Figure 34: ViaggiaRovereto mobile app - STREETLIFE features**

### 6.3.2. Bike Sharing app

This mobile app allows to know, in real time, the information of the different bike sharing services available in the city. The application supports both standard bike-sharing services that use docking stations, as well as free city bikes (bikes that can be picked up by any user without requiring a registration) tracked via GPS devices.
Figure 35: Bike Sharing mobile app

For traditional bike sharing services the mobile app allows to inspect, via map or in a list view, the number of bikes and docking points at each station. It is also possible to add favorite stations for a quick search. Enabling the usage of GPS on the smartphone, it is possible to get the list of stations in close proximity to the user.

Free city bikes can be seen on the city map or in a list view.

For both kind of bike sharing services, it is possible to i) get the directions to the bike pick up point (exploiting ViaggiaRovereto mobile app), ii) signal problems related to the bike/station or the service, and iii) inspect the notifications sent by other users.

The app is used in Rovereto for the eMotion bike-sharing service and in Saarbrucken University Campus for the free bike sharing service.

6.3.3. Parking availability mobile app

This mobile app allows collecting data about parking availability for on-street parking and parking lots. The functionalities currently provided are the following (see Figure 36: Parking availability mobile appFigure 36 for some screenshots):

- View and select parking lots/streets from city map or list;
- Search a parking lot/street through free text;
- For each parking lot, specify the number of free parking spaces on a certain time/date and of unusable parking spaces (e.g. due to road works);
- For each street segment, specify the number of free parking spaces on a certain time/date (differentiated by parking type: free parking, parking in parking meter, disc parking) and of unusable parking spaces (e.g. due to road works);
- For each parking lot/street segment, inspect the list of previous notifications.
Figure 36: Parking availability mobile app
The app is used in Rovereto by traffic aids to collect data that is used to obtain real-time information about parking lot occupancy (for free parking lots not equipped with access control systems) and to derive statistical information about on-street parking availability. The collected data is aggregated through the Data Management Rovereto sub-system built on top of FI-WARE technology (WP3), and exploited by ViaggiaRovereto mobile app to show real-time parking lots occupancy and estimated time to find an on-street parking.

7. CONCLUSION AND NEXT STEPS

Work in WP5 is progressing as planned. Significant effort has been put for speedy development of Gamification Engine (T5.3) and Advanced Graphical Interfaces (T5.4). The former is already in a mature state, configurable via a user interface. The latter, a mixed reality app, based on work on T5.4 and T3.2, has been implemented on an Android device with server-side static and real-time data processing and transmission pipelines. Understanding of personalised routing and travel assistance has been improved and our routing cases analysed by developing a conceptual description. Fundamental work for creating a lightweight Open Source multi-modal routing solution with real time support based on Open Data is well underway. High-level STREETLIFE app interfaces have been designed and existing ones extended with support for greener behaviour, and new apps such as a mobile parking availability app created.

The requirements and implementation ideas of D5.1 will be followed in the next period, supported by on-site validation, experimentation and piloting.
APPENDIX A: LITERATURE


APPENDIX B: REQUIREMENTS AND IMPLEMENTATION IDEAS FROM D5.1

Mobile Navigation Aids -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-GEN-1</td>
<td>MUST</td>
<td>Navigational support for initial orientation</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-GEN-2</td>
<td>MUST</td>
<td>Navigational support for manoeuvring</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-GEN-3</td>
<td>MUST</td>
<td>Navigational support for maintaining orientation</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-GEN-4</td>
<td>MUST</td>
<td>Navigational support for recognising the target</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>

Navigational Knowledge Presentation -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-KNOW-1</td>
<td>MUST</td>
<td>Maintain and present landmark knowledge in navigation</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-KNOW-2</td>
<td>MUST</td>
<td>Maintain and present route knowledge in navigation</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-KNOW-3</td>
<td>MUST</td>
<td>Maintain and present configurational knowledge in navigation</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>

Navigation Interface -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-UI-1</td>
<td>SHOULD</td>
<td>Minimise cognitive load (as defined by working memory load, amount or duration of cognitive task processing, or complexity of mental computations)</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-UI-2</td>
<td>SHOULD</td>
<td>Minimise motor effort and procedural complexity</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-UI-3</td>
<td>SHOULD</td>
<td>Minimise use of time</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>

Navigation Knowledge Guidelines -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-KNOW-4</td>
<td>SHOULD</td>
<td>Maximise information that helps orientation</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-KNOW-5</td>
<td>SHOULD</td>
<td>Maximise information that helps performing the current task</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>
### Navigation Objectives – Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-OBJ-1</td>
<td>SHOULD</td>
<td>User is able to find and travel through all places of interest</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-OBJ-2</td>
<td>SHOULD</td>
<td>Uses does not get lost</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-OBJ-3</td>
<td>SHOULD</td>
<td>User is able to re-visit places with less effort</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-OBJ-4</td>
<td>SHOULD</td>
<td>User feels familiar with the space</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>

### Navigation Annotation – Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-ANN-1</td>
<td>SHOULD</td>
<td>Virtual signage should be placed at decision points</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-ANN-2</td>
<td>SHOULD</td>
<td>Virtual signage should be legible</td>
<td>NAV-GEN</td>
</tr>
<tr>
<td>NAV-ANN-3</td>
<td>SHOULD</td>
<td>Virtual signage should utilise the presence of landmarks</td>
<td>NAV-GEN</td>
</tr>
</tbody>
</table>

### Routing – Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER-RI-1</td>
<td>MUST</td>
<td>an app MUST be able to support multi-modal route planning</td>
<td>BER-PTP/1</td>
</tr>
<tr>
<td>BER-RI-9</td>
<td>SHOULD</td>
<td>The computation of actual necessary mode of transport SHOULD consider the user preferences. These user preferences MAY be predefined or given by the user directly. For example a user MAY be able to define not to use the bike on rainy days.</td>
<td>BER-PTP/3</td>
</tr>
<tr>
<td>ROV-RI-43</td>
<td>MUST</td>
<td>The STREETLIFE routing service MUST be able to provide the multi-modal P+R route even if the user has not original specified public transport or alternative transport means among the preferences for the trip</td>
<td>ROV-PR/1</td>
</tr>
</tbody>
</table>
ROV-RI-44 | MUST | The STREETLIFE system MUST record the choice taken by the user about the trip with destination in the city center | ROV-PR/1

ROV-RI-45 | MUST | The STREETLIFE system MUST be able to send notifications about events that are relevant to the car driver’s trip | ROV-PR/2

TRE-RI-2 | MUST | Take real-time data into account in journey planning | TRE-2/1

TRE-RI-3 | MUST | Remember user preferences in journey planning | TRE-2/1, TRE-4/

TRE-RI-4 | SHOULD | User location awareness in journey planning (suggest current location as the route start point) | TRE-2/1, TRE-4/1

TRE-RI-6 | MUST | Park & ride journey planning: citizens can plan a journey which combines private and public transportation. | TRE-4/1

TRE-RI-9 | MUST | Publish an Open API for journey planner, to boost development of new third party applications. | TRE-2/1

### Personalisation -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER-RI-9</td>
<td>SHOULD</td>
<td>The computation of actual necessary mode of transport SHOULD consider the user preferences. These user preferences MAY be predefined or given by the user directly. For example a user MAY be able to define not to use the bike on rainy days.</td>
<td>BER-PTP/3</td>
</tr>
<tr>
<td>BER-RI-11</td>
<td>MAY</td>
<td>A user MAY create and edit several profiles each with a different set of preferences. For example predefined preference templates like stay dry, low (to zero) carbon footprint, fastest connection, low cost and safety</td>
<td>BER-PTP/5</td>
</tr>
<tr>
<td>BER-RI-12</td>
<td>MAY</td>
<td>A user MAY be able to save the preferences under different names.</td>
<td>BER-PTP/5</td>
</tr>
<tr>
<td>BER-RI-13</td>
<td>MAY</td>
<td>The STREETLIFE system MAY be able to reason and self-learning from previous user-behaviour and preferences when suggesting route trips.</td>
<td>BER-PTP/5</td>
</tr>
<tr>
<td>BER-RI-14</td>
<td>SHOULD</td>
<td>The STREETLIFE system SHOULD have a centralised Identity Management System for end-user.</td>
<td>BER-PTP/6</td>
</tr>
<tr>
<td>BER-RI-15</td>
<td>MUST</td>
<td>A corresponding App and a STREETLIFE connected website MUST share the same Identity Management System.</td>
<td>BER-PTP/6</td>
</tr>
<tr>
<td>BER-RI-23</td>
<td>SHOULD</td>
<td>The user SHOULD be able to add other STREETLIFE user as friends to his profile.</td>
<td>BER-CPI/2</td>
</tr>
<tr>
<td>BER-RI-24</td>
<td>SHOULD</td>
<td>A STREETLIFE App SHOULD act in behalf of the user’s STREETLIFE system identity.</td>
<td>BER-CPI/4</td>
</tr>
<tr>
<td>ROV-RI-9</td>
<td>MUST</td>
<td>be able to register a user with her personal profile to the car pooling service</td>
<td>ROV-CP/1</td>
</tr>
<tr>
<td>ROV-RI-10</td>
<td>MUST</td>
<td>be able to define contact preferences for the car pooling service</td>
<td>ROV-CP/1</td>
</tr>
<tr>
<td>ROV-RI-11</td>
<td>MUST</td>
<td>be able to specify the itinerary for her ride request</td>
<td>ROV-CP/1</td>
</tr>
<tr>
<td>ROV-RI-16</td>
<td>SHOULD</td>
<td>interface with the user’s calendar for such reminders</td>
<td>ROV-CP/1</td>
</tr>
<tr>
<td>ROV-RI-22</td>
<td>MUST</td>
<td>be able to mark ride requests with other (personal) tags</td>
<td>ROV-CP/2</td>
</tr>
<tr>
<td>ROV-RI-27</td>
<td>MUST</td>
<td>be able to input into the system the features needed for identification</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-28</td>
<td>SHOULD</td>
<td>be able to input into the system his/her recurrent starting place of a trip</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-29</td>
<td>SHOULD</td>
<td>be able to input into the system his/her recurrent ending place of a trip</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-30</td>
<td>SHOULD</td>
<td>be able to input into the system his/her recurrent time of a trip</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-31</td>
<td>SHOULD</td>
<td>be able to give inputs to the system about his/her workplace and/or other recurrent destinations</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-32</td>
<td>SHOULD</td>
<td>be able to input infos to the system about his/her common routes</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-33</td>
<td>SHOULD</td>
<td>be able to mark a trip into the system as a very frequent one</td>
<td>ROV-CP/5</td>
</tr>
<tr>
<td>ROV-RI-34</td>
<td>MUST</td>
<td>be able to read into the system the updated standings of the gamification system</td>
<td>ROV-CP/6</td>
</tr>
<tr>
<td>ROV-RI-35</td>
<td>MUST</td>
<td>be able to know the basic personal informations of the users in order to contact them when the game end</td>
<td>ROV-CP/6</td>
</tr>
<tr>
<td>ROV-RI-44</td>
<td>MUST</td>
<td>The STREETLIFE system MUST record the choice taken by the user about the trip with destination in the city center</td>
<td>ROV-PR/1</td>
</tr>
<tr>
<td>ROV-RI-45</td>
<td>MUST</td>
<td>The STREETLIFE system MUST be able to send notifications about events that are relevant to the car driver’s trip</td>
<td>ROV-PR/2</td>
</tr>
</tbody>
</table>
TRE-RI-3  MUST  Remember user preferences in journey planning  TRE-2/1, TRE-4/1
TRE-RI-4  SHOULD  User location awareness in journey planning (suggest current location as the route start point)  TRE-2/1, TRE-4/1

**Virtual mobility – 2D Image Based Guidance -- Requirements**

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-IMG-R-1  MUST</td>
<td>Annotated georeferenced and oriented photos of public transportation stops and of recognisable landmarks in the vicinity</td>
<td>TRE-IMG</td>
<td></td>
</tr>
</tbody>
</table>

**Virtual mobility – 2D Guidance -- Implementation Ideas**

<table>
<thead>
<tr>
<th>Impl.Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-IMG-S-1  MUST</td>
<td>A routing capability that is associated with a visualisation database</td>
<td>TRE-IMG</td>
<td></td>
</tr>
<tr>
<td>NAV-IMG-S-2  MUST</td>
<td>GPS and orientation tagged photo database of transportation stops and recognisable landmarks supporting JPG and PNG formats</td>
<td>TRE-IMG</td>
<td></td>
</tr>
<tr>
<td>NAV-IMG-S-3  MUST</td>
<td>Annotations with associated contextual data (bus stop numbers etc) extracted from for example GTFS or Kalkati.net data</td>
<td>TRE-IMG</td>
<td></td>
</tr>
<tr>
<td>NAV-IMG-S-4  MAY</td>
<td>Potentially crowd-sourcing content and annotation tools for image database creation</td>
<td>TRE-IMG</td>
<td></td>
</tr>
<tr>
<td>NAV-IMG-S-4  MUST</td>
<td>A user interface binding journey planning functionality to photo visualisation using geotagging or road segment identifiers for association</td>
<td>TRE-IMG</td>
<td></td>
</tr>
</tbody>
</table>

**360° Image Based Guidance – Requirement Ideas**

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-360-R-1  MUST</td>
<td>Ability to view and zoom annotated 360° photos</td>
<td>TRE-360</td>
<td></td>
</tr>
</tbody>
</table>
### 360° Image Based Guidance – Implementation Ideas

<table>
<thead>
<tr>
<th>Impl. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-360-S-1</td>
<td>MUST</td>
<td>360° photo viewer within the mobile application, OpenGL ES and C/C++ based implementation for AR/3D compatibility</td>
<td>TRE-360</td>
</tr>
<tr>
<td>NAV-360-S-2</td>
<td>MUST</td>
<td>360° photo database of key navigational points: bus stops, landmarks; geotagged or associated to road segment identifiers</td>
<td>TRE-360</td>
</tr>
<tr>
<td>NAV-360-S-3</td>
<td>MUST</td>
<td>A user interface binding journey planning functionality to 360° photo visualisation; Java UI feasible for Android for AR/3D compatibility</td>
<td>TRE-360</td>
</tr>
</tbody>
</table>

### Gamification -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAM-R-1</td>
<td>MUST</td>
<td>Store and display gamer profile of the user</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-2</td>
<td>MUST</td>
<td>Allow end user to join with her profile and leave one (or multiple) mobility games</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-3</td>
<td>MUST</td>
<td>Associate a number of green leaves point to specific user actions and choices within the end user application</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-4</td>
<td>MUST</td>
<td>Present to the number of points associated to the various actions and choices that are available to the user, based on the current usage context of the end user application</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-5</td>
<td>MUST</td>
<td>Update green leaves point earned by the player, based on actions taken by the player within the end user application</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-5</td>
<td>MUST</td>
<td>Update badge collection of end user based on end user achievements</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-6</td>
<td>SHOULD</td>
<td>Assign awards (prizes) to end user based on end user achievements</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-7</td>
<td>MUST</td>
<td>Show the leader board of the game(s) in which the user participates and the ranking of the user in it</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-8</td>
<td>SHOULD</td>
<td>Show a filtered leader board that includes only a subset of players indicated by this player</td>
<td>GAM-PBA</td>
</tr>
<tr>
<td>GAM-R-9</td>
<td>MUST</td>
<td>Log all game-related user activity for history and audit purposes</td>
<td>GAM-PBA</td>
</tr>
</tbody>
</table>
### 3D Virtual Environments – Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-3D-R-1</td>
<td>MUST</td>
<td>A sufficiently detailed, realistic 3D model of target area</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-R-2</td>
<td>MUST</td>
<td>Real-time rendered visualisation of the 3D city model on mobile devices (20fps or better)</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-R-3</td>
<td>MUST</td>
<td>Free viewpoint for the virtual environment (i.e., not bound to ground level)</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-R-4</td>
<td>MUST</td>
<td>Visualisation of public transportation with realistic vehicles in their current position in the 3D city model</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-R-5</td>
<td>MUST</td>
<td>Gesture and touch screen based interaction for the 3D view</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-R-6</td>
<td>MUST</td>
<td>User positioning</td>
<td>TRE-3D</td>
</tr>
</tbody>
</table>
### 3D Virtual Environments – Implementation Ideas

<table>
<thead>
<tr>
<th>Impl. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-3D-R-7</td>
<td>MUST</td>
<td>Attention management and visual emphasis for essential navigational entities: bus stops, buses</td>
<td>TRE-3D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAV-3D-S-1</td>
<td>SHOULD</td>
<td>SIRI public transportation real time tracking network service</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-2</td>
<td>MUST</td>
<td>Realistic 3D model of Tampere in VRML format with JPEG or PNG texture files</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-3</td>
<td>MUST</td>
<td>Bus route and bus stop data (GTFS, Kalkati.net)</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-4</td>
<td>MUST</td>
<td>Routing engine (integrating GTFS/Kalkati.net and OpenStreetMap)</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-5</td>
<td>SHOULD</td>
<td>3D engine capable of viewing large scale 3D urban environments and large numbers of dynamic content and entities. Implementation with C/C++ and OpenGL ES to allow case specific optimisations.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-6</td>
<td>MUST</td>
<td>Mobile platform capable of real time 3D graphics and networking; NVidia Tegra 3 or newer for CPU/GPU with OpenGL ES/OpenGL support; 3G/4G or Wi-Fi with TCP/IP using sockets.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-7</td>
<td>MUST</td>
<td>Mobile platform with spatial sensing capability (GPS/RTK; internal or external inertia sensor, magnetometer and accelometer). Android SensorManager.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3D-S-7</td>
<td>MUST</td>
<td>A user interface binding journey planning functionality to dynamic 3D visualisation. Java UI on Android.</td>
<td>TRE-3D</td>
</tr>
</tbody>
</table>
### 3D Interaction – Implementation Ideas

<table>
<thead>
<tr>
<th>Impl. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-3DUI-S-1</td>
<td>SHOULD</td>
<td>Tracks - Minimise micro-manoeuvering. Restrict navigation space to a street network. Provide street names.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-2</td>
<td>SHOULD</td>
<td>Orientation value - Maximise view’s orientation value. Orient downward when elevating, and upward when descending; when in tracks mode at street level, translate away from the opposing façade for a better view.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-3</td>
<td>SHOULD</td>
<td>Speed adjustment - Match motion with user’s needs. Adjust speed automatically and smoothly based on elevation, being slower at street level and faster at sky.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-4</td>
<td>SHOULD</td>
<td>View landmark - Orientation aid. Trigger an animated view transition to present a landmark and the user’s current position.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-5</td>
<td>SHOULD</td>
<td>Change viewpoint - Minimise micro-manoeuvering. Scripted view level transition to three predetermined view levels (street level, rooftop level, sky) with automatic orientation.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-6</td>
<td>SHOULD</td>
<td>Markers - The main signage visualisation method. Decrease cognitive load, enable targeted search instead of primed search. Marker arrows point at, for example, the start point and the target, releasing users from remembering the exact positions.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-7</td>
<td>SHOULD</td>
<td>Fly-to-target - A scripted action to fly to a target. If fast transition is needed, a smooth but fast fly is less disorienting than teleportation, and demands less from the user than manual manoeuvering. Can be triggered with a double-click.</td>
<td>TRE-3D</td>
</tr>
<tr>
<td>NAV-3DUI-S-7</td>
<td>SHOULD</td>
<td>Orbit mode - Aid for recognising a target. View is locked towards a point, and controls are mapped to a cylindrical coordinate system, orbiting around the point.</td>
<td>TRE-3D</td>
</tr>
</tbody>
</table>

### Augmented Reality -- Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-AR-R-1</td>
<td>MUST</td>
<td>Capability for public transportation routing</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-R-2</td>
<td>MUST</td>
<td>Capability to mark navigational entities and journey instructions on the camera view of a mobile device</td>
<td>TRE-AR</td>
</tr>
</tbody>
</table>
Augmented Reality – Implementation Ideas

<table>
<thead>
<tr>
<th>Impl. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV-AR-S-1</td>
<td>MUST</td>
<td>Bus route and bus stop data (GTFS, Kalkati.net)</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-S-2</td>
<td>MUST</td>
<td>SIRI public transportation real time tracking network service</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-S-3</td>
<td>MUST</td>
<td>Routing engine with real time tracking support (integrating OpenStreetMap and GTFS/Kalkati.net data)</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-S-4</td>
<td>MUST</td>
<td>Mobile device with capability for augmented reality with networking connection, Android Camera API, 3G/4G or Wi-Fi with TCP/IP connectivity using or Java.net.socket.</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-S-5</td>
<td>MUST</td>
<td>Mobile platform with spatial sensing capability (GPS/RTK, internal or external inertia sensor, magnetometer and accelerometer). Android SensorManager.</td>
<td>TRE-AR</td>
</tr>
<tr>
<td>NAV-AR-S-6</td>
<td>MUST</td>
<td>A user interface binding journey planning functionality to augmented reality. Java UI.Fragments on Android.</td>
<td>TRE-AR</td>
</tr>
</tbody>
</table>

Mobile Apps – Requirement Ideas

<table>
<thead>
<tr>
<th>Req. Idea</th>
<th>Category</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP-1</td>
<td>MUST</td>
<td>Have support on a commonly available platform</td>
<td>ALL</td>
</tr>
<tr>
<td>APP-2</td>
<td>MUST</td>
<td>Support at least one STREETLIFE Use Case or component as a standalone app</td>
<td>ANY</td>
</tr>
<tr>
<td>APP-3</td>
<td>MUST</td>
<td>Be possible to be validated via either benchmarking or user experience</td>
<td>ALL</td>
</tr>
<tr>
<td>APP-4</td>
<td>SHOULD</td>
<td>Follow the general STREETLIFE goal of leveraging green means of travel</td>
<td>ANY</td>
</tr>
</tbody>
</table>