simpli·city

The Road User Information System Of The Future

WP2 – Vision and Requirements

D2.3: Requirements Analysis

Deliverable Lead: TEMP

Contributing Partners: TUV, ASC, TIE, TUDA, IBM, FGM, TALK, TEMP, SRM, CRF

Delivery Date: 01/2014

Dissemination Level: Public

Version 2.0

This document lists and provides a high-level description of the requirements of the SIMPLI-CITY project. These requirements have been gathered taking into account the point of view of the different users of the platform: end users and developers. It also presents the requirements of the use cases to be implemented within the project.

SEVENTH FRAMEWORK PROGRAMME

Document Status	
Deliverable Lead	Marc Cardenete, TEMP
Internal Reviewer 1	Sven Abels, ASC
Internal Reviewer 2	Ulrich Lampe and Christian Gottron, TUDA
Type	Deliverable
Work Package	WP2: Vision and Requirements
ID	D2.3: Requirements Analysis
Due Date	31.03.2013
Delivery Date	24.01.2014
Status	Approved

Document History	
Draft Version	V1.0, TEMP, 01.03.2012
Contributions	V1.1, TUV, ASC, TIE, TUDA, IBM, FGM, TALK, TEMP, SRM, CRF, 16.04.2013 V1.2, TUV, ASC, TIE, TUDA, IBM, FGM, TALK, TEMP, SRM, CRF, 23.04.2013 V1.3, TUV, ASC, TIE, TUDA, IBM, FGM, TALK, TEMP, SRM, CRF, 07.05.2013 V1.4, TEMP, 16.05.2013 (First Review Version) V1.5, TEMP, 04.06.2013 (Second Review Version) V1.6, TEMP, 12.06.2013 (approved by the internal reviewers) V1.7, TEMP, 17.01.2014 V1.8, TEMP, 21.01.2014 (First Review of the Second Version) V1.9, TEMP, 23.01.2014 (Second Review of the Second Version)
Final Version	V2.0, TEMP, 24.01.2014 (Second Version approved by the internal reviewers)

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners

Vienna University of Technology (Coordinator), Austria

Ascora GmbH, Germany

TIE Nederland B.V., The Netherlands

Technische Universität Darmstadt, Germany

IBM Research – Ireland

Smarter Cities Technology Centre

Forschungsgesellschaft Mobilität, Austria

Talkamatic AB, Sweden

Atos Worldline, Spain

Centro Ricerche FIAT, Italy

SRM – Reti e Mobilità, Italy

Executive Summary

This document lists and describes the requirements of the SIMPLI-CITY project, which have the objective to identify the user needs and to describe the requisites of the SIMPLI-CITY architecture in order to fulfil that needs. These requirements will serve as a basis for the upcoming tasks of the project.

The document starts by defining the engineering process that has been used within the project in order to elicit and analyse the list of requirements.

Afterwards, the complete list of user requirements is provided. These requirements are the ones expected by SIMPLI-CITY from the user's point of view. SIMPLI-CITY has two main groups of users: end users (or road users) and developers.

The following sections describe in detail each requirement, and define which tasks of the project are involved in its implementation. These requirements are defined depending on its nature: strategic, functional, or related to use cases. The strategic ones refer to high-level goals of the project. The functional ones specify concrete functionalities to be provided by the different components of the SIMPLI-CITY architecture. Finally, the requirements of the use cases refer to functionalities to be provided by the different use case scenarios that will be implemented within the project.

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Table of Contents

1 Introduction

SIMPLI-CITY – The Road User Information System of the Future – is a project funded by the Seventh Framework Programme of the European Commission under Grant Agreement No. 318201. It provides the technological foundation for bringing the “App Revolution” to road users by facilitating data integration, service development, and end user interaction.

This deliverable presents the requirements specification for the project, containing a high level definition of the requirements that are required by the project as a whole. These requirements are extracted from the project vision, market report, and the preliminary definition of the use case scenarios that will be implemented within the project.

1.1 SIMPLI-CITY Project Overview

Analogously to the “App Revolution”, SIMPLI-CITY adds a “software layer” to the hardware-driven “product” mobility. SIMPLI-CITY will take advantage of the great success of mobile apps that are currently being provided for systems such as Android, iOS, or Windows Phone. These apps have created new opportunities and even business models by making it possible for developers to produce new applications on top of the mobile device infrastructure. Many of the most advanced and innovative apps have been developed by players formerly not involved in the mobile software market. Hence, SIMPLI-CITY will support third party developers to efficiently realise and sell their mobility-related service and app ideas by a range of methods and tools, including the Mobility Services and Application Marketplaces.

In order to foster the wide usage of those services, a holistic framework is needed which structures and bundles potential services that could deliver data from various sources to road user information systems. SIMPLI-CITY will provide such a framework by facilitating the following main project results:

- Mobility Service Framework: A next-generation European Wide Service Platform (EWSP) allowing the creation of mobility-related services as well as the creation of corresponding apps. This will enable third party providers to produce a wide range of interoperable, value-added services, and apps for drivers and other road users.
- Mobility-related Data as a Service: The integration of various, heterogeneous data sources like sensors, cooperative systems, telematics, open data repositories, people-centric sensing, and media data streams, which can be modelled, accessed, and integrated in a unified way.
- Personal Mobility Assistant: An end user assistant that allows road users to make use of the information provided by apps and to interact with them in a non-distracting way – based on a speech recognition approach. New apps can be integrated into the Personal Mobility Assistant in order to extend its functionalities for individual needs.

To achieve its goals, SIMPLI-CITY conducts original research and applies technologies from the fields of Ubiquitous Computing, Big Data, Media Streaming, the Semantic Web, the Internet of Things, the Internet of Services, and Human-Computer Interaction. For more information, please refer to the project Website at http://www.simpli-city.eu.

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

1.2 Deliverable Purpose, Scope and Context

The purpose of this deliverable is to enumerate the requirements of the project, including strategic requirements of the project and the functional requirements of the different components of the SIMPLI-CITY platform and of the scenarios to be implemented within the project. These requirements will be used within the further definition and implementation of the different components and use case scenarios of SIMPLI-CITY in the upcoming work packages and tasks of the project.

1.3 Document Status and Target Audience

This document is listed in the Description of Work (DoW) as “public”, as it provides general information about the goals and scope of SIMPLI-CITY and can therefore be used by external parties in order to get according insight into the project activities.

While the document primarily aims the project partners, this public deliverable can also be useful for the wider scientific and industrial community. This includes other publicly funded projects, which may be interested in collaboration activities.

1.4 Abbreviations and Glossary

A definition of common terms and roles related to the realization of SIMPLI-CITY as well as a list of abbreviations is available in the supplementary document “Supplement: Abbreviations and Glossary”, which is provided in addition to this deliverable.

Further information can be found at http://www.simpli-city.eu.

1.5 Document Structure

This deliverable comprises the following sections:

Section 1 provides an introduction for this deliverable, including a general overview of the project, and outlines the purpose, scope, context, status, and target audience of this deliverable.

Section 2 describes the methodology and procedure of the requirements engineering process used within the project.

Section 3 presents the complete list of user requirements expected from the project from the user’s point of view

Section 4 presents the strategic requirements of the project, derived from the user requirements and which refer to the high-level goals of the project.

Section 5 describes the functional requirements of the different components of the SIMPLI-CITY platform.

Section 6 covers the requirements of the use cases scenarios of the project.

Finally, Section 7 concludes the document by describing how the requirements will serve as a basis for the upcoming work of the project.

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

2 Requirements Engineering Process

This section describes the requirements engineering process used in the SIMPLI-CITY project. This process covers all the activities involved in discovering, documenting, and maintaining the set of requirements of the project.

2.1 Requirements Definition

The requirements are captured at different levels of detail, each of them leading to a further refinement of the requirements set. These requirements are differentiated between:

- User requirements, which are the requirements expected from the point of view of the different users of the project: end users and developers.
- Strategic requirements represent business application goals and high-level expectations of the project.
- Functional requirements describe what has to be done to fulfil the goals and expectations of the project. They cover the functionalities to be provided by the different components architecture of the project.

Figure 1 illustrates the requirements engineering process of the SIMPLI-CITY project, and covered by this deliverable.

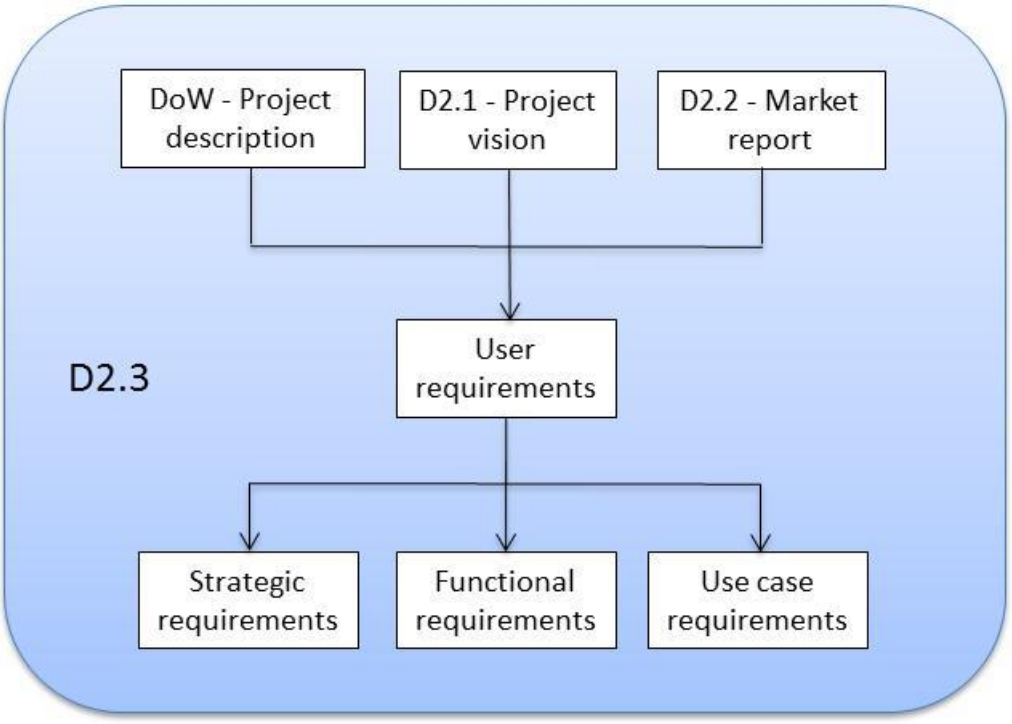

Figure 1: SIMPLI-CITY Requirements Engineering Process

2.2 Methodology

The gathering of requirements is formed by the following processes:

- Elicitation, which refers to the identification and capture of the needs of the different users of the SIMPLI-CITY platform, namely end users and developers.
- Analysis, in order to refine the user needs into formal specifications.

The methodology used within the project is described in the following subsections.

2.2.1 Elicitation

This elicitation process included the generation of a list of user requirements, created from the already established requirements of the project description, the project vision, the market report, and the definition of the use case scenarios.

The elicitation of the user requirements process was carried out in a workshop in one of the Plenary Meetings of the project, held in Barcelona in February 2013. Two different working groups were set and each of them carried out a brainstorming session in order to elicit the requirements of the project, taking into account the point of view of the users of the SIMPLI-CITY project, namely end users and developers. After that, the two lists of requirements were merged, grouped, and discussed by all the partners

The requirements were collected by means of a mind map, which can be observed in the Annex of this deliverable. The complete list of user requirements is provided in section 3 of this deliverable.

2.2.2 Analysis

This process included the processing of the collected data in order to determine which requirements could be considered and included within the scope of the project, and in order to avoid overlapping requirements. This analysis process also permitted to add missing requirements that were extracted by reviewing the previously created documents like the project description or project vision.

Each requirement was defined with a priority, between 1 and 5, meaning high and low priority, respectively. This prioritization was later mapped to the MoSCoW one, described in section 3, which prioritizes the requirements into the following possibilities: Must have, Should Have, Could Have, and Will Not Have For Now; identifying the requirements that will be addressed during the project's lifetime.

The requirements were separated between the ones aimed at end users, the ones aimed at developers, and requirements of the use cases. Some of the requirements were defined as strategic requirements, others as functional requirements, and others as use case requirements. The strategic ones refer to the high-level goals of the project, whereas the functional ones refer to specific functionalities to be provided by the SIMPLI-CITY components. Finally, the use case requirements refer to concrete functionalities to be implemented within the use case scenarios of the project.

Each functional requirement was appointed to a task of the RTD work packages (WP4, WP5 and WP6). The appointments were done taking into account the primary task assignment of each requirement, even though a single requirement may imply work from different tasks of the project, indicated as secondary task assignments.

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

3 User Requirements

This section lists all the requirements expected from SIMPLI-CITY from a user point of view. There are two main users of SIMPLI-CITY: end users and developers. End users are the road users that make use of SIMPLI-CITY during their travel, whereas developers are the ones that develop services and apps that run over the SIMPLI-CITY platform and that are aimed at end users. Thus, the requirements are separated between the ones expected by end users and the ones expected by developers. A third group includes the use cases requirements, that refer to the functionalities expected to be implemented in the use case scenarios of the project; this group of requirements are not included within the first two groups because these requirements refer to functionalities to be provided by the applications developed over the SIMPLI-CITY platform, not by the platform itself. The different requirements are grouped by functionality.

The requirements are prioritized using the *MoSCoW* method into:

- Must Have Requirements: Requirements that need to be met in any case during the project lifetime, as they provide a functionality that is essential for the overall project. These requirements need to be implemented in any way.
- Should Have Requirements: Important requirements which are nevertheless not as crucial for the overall project. Hence, they may be implemented if resources allow.
- Could Have Requirements: Not so important requirements, which nevertheless provide an interesting functionality. If resources allow, they may be implemented, but the chances are rather small.
- Will Not Have For Now Requirements: Requirements that are not relevant in the context of SIMPLI-CITY, but may be relevant in future projects in this research area. These requirements will not be covered in the project..

This section only lists the requirements, which will be further described in the upcoming sections 4 (strategic requirements), 5 (functional requirements), and 6 (use case requirements).

3.1 End Users

Usage types	
U1. Support of car drivers as end users	Priority: Must Have
U2. Support of cyclists as end users	Priority: Should Have
U3. Support of private transport passengers as end users	Priority: Should Have
U4. Support of public transport users as end users	Priority: Could Have
U5. Support of pedestrians as end users	Priority: Could Have
U6. Support of truck drivers as end users	Priority: Could Have
U7. Support for disabled drivers. It should take into account the interaction between the user and the road information system.	Priority: Will Not Have For Now

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

U8. Multiuser support. It should support multiple drivers and different users	Priority: Will Not Have For Now
User goals – Strategic	
U9. Allow users to become green	Priority: Must Have
U10. Reduction of the number of accidents	Priority: Must Have
U11. Provision of useful information to drivers	Priority: Must Have
U12. No distraction of the driver	Priority: Must Have
U13. No SIMPLI-CITY big brother, i.e., no excessive data collection beyond the necessary scope	Priority: Should Have
User goals - Tactical	
U14. Show real time carbon print	Priority: Must Have
U15. Help to balance the traffic in a particular area	Priority: Must Have
U16. Provide ordered driving in the city	Priority: Must Have
U17. Allow local authorities to connect easily with users	Priority: Must Have
User experience - UI and usability of apps	
U18. Reasonable response time	Priority: Should Have
U19. Minimum manual configuration	Priority: Could Have
U20. Multilingual	Priority: Will Not Have For Now
U21. Link voice commands to apps	Priority: Will Not Have For Now
U22. Personalization - Incremental configuration	Priority: Could Have
User experience - Look&Feel of apps	
U23. Unified Look&Feel within the project	Priority: Should Have
U24. Unified Look&Feel for third party developers	Priority: Should Have
U25. Usage of UI guidelines within the project	Priority: Should Have
U26. Intuitive Usability	Priority: Should Have

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

User experience - Behaviour - System reaction to context	
U27. Proactive behaviour	Priority: Must Have
U28. The system learns from feedback	Priority: Will Not Have For Now
U29. Reaction to who is in the car, via simple dialog	Priority: Could Have
U30. Reaction to who is in the car, via sensors	Priority: Should Have
U31. Reaction to time of the day	Priority: Must Have
U32. Reaction to sensors	Priority: Must Have
U33. Reaction to KPIs from the car, like level of fuel and kilometres done	Priority: Must Have
U34. Reaction to history of usage	Priority: Must Have
U35. Reaction to traffic information, like traffic jams, train schedules, road works, accidents, and strikes	Priority: Must Have
User experience - Behaviour - Interaction with the system	
U36. Natural speech recognition	Priority: Must Have
U37. Results oriented instead of service/app recognition oriented	Priority: Could Have
U38. Disambiguation	Priority: Could Have
U39. Provision of a limited number of alternatives	Priority: Could Have
U40. Friendly voice	Priority: Could Have
U41. Input interactions with system via multimodal UI: on screen, voice control, and non-voice control	Priority: Must Have
U42. Output interaction from system through UI	Priority: Must Have
U43. Non-distracting interaction	Priority: Should Have
U44. Voice quality	Priority: Could Have
U45. Automotive quality voice recognition (automotive acoustic models for in car use)	Priority: Should Have
U46. High speech recognition rate	Priority: Should Have
U47. Voice interaction through the car audio system (microphone & loudspeakers) for hands free in car use	Priority: Could Have

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

User experience - App crashing		
U48.	App crashes are minimized	Priority: Should Have
U49.	A crash should not impact other apps	Priority: Should Have
User experience - Offline access		
U50.	Prefetching of media data & offline access	Priority: Must Have
U51.	Avoid the download of data from 3G	Priority: Should Have
U52.	Offline access of data used within apps	Priority: Should Have
U53.	App recommendations	Priority: Could Have
U54.	Expiration of data	Priority: Should Have
User experience - Feedback		
U55.	Rating of apps	Priority: Should Have
U56.	Feedback to developers through the marketplace	Priority: Should Have
U57.	Social network functionality with the objective of spreading the word	Priority: Could Have
U58.	User recommendations	Priority: Could Have
User experience - Generic services		
U59.	User centric data services, e.g., Facebook events, calendar	Priority: Must Have
Marketplace		
U60.	Payment for apps and services – Mock-up	Priority: Could Have
U61.	Payment for apps and services - Fully deployed	Priority: Will Not Have For Now
U62.	Versioning of apps	Priority: Must Have
U63.	Upgrading of apps	Priority: Must Have
U64.	Quality assurance – Mock-up	Priority: Must Have
U65.	Quality assurance - Fully deployed	Priority: Could Have
U66.	Marketplace easy to use	Priority: Must Have
U67.	Apps and services are easy to buy	Priority: Must Have

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

U68.	Discovery of apps and services	Priority: Must Have
U69.	App download into the device	Priority: Must Have
U70.	App installation	Priority: Must Have
U71.	App uninstallation	Priority: Must Have
Environment - Device		
U72.	Android support	Priority: Must Have
U73.	iOS support	Priority: Will Not Have For Now
U74.	Blackberry support	Priority: Will Not Have For Now
U75.	Windows phone support	Priority: Will Not Have For Now
U76.	Tablet support	Priority: Could Have
U77.	The PMA fits into the pocket	Priority: Will Not Have For Now
U78.	The battery lifetime of the PMA is longer than 1 day	Priority: Will Not Have For Now
Environment		
U79.	Support to exchange devices	Priority: Will Not Have For Now
U80.	Profile in the cloud	Priority: Must Have
U81.	Storage of data in the cloud	Priority: Must Have
Environment - Connectivity		
U82.	Gateway to data sources	Priority: Will Not Have For Now
U83.	Interaction with head up display	Priority: Will Not Have For Now
U84.	Interaction with user's devices, e.g. contacts exchange	Priority: Could Have
U85.	Interaction with car sensors, e.g. car sensors	Priority: Must Have
Privacy		
U86.	Transparency	Priority: Must Have

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

U87. Confidentiality. Does not give away data to third parties	Priority: Must Have
U88. Data encryption	Priority: Must Have
U89. Certification. Only certified apps are allowed to access users data	Priority: Must Have
Security	
U90. Availability	Priority: Must Have
U91. Integrity	Priority: Must Have
U92. Secure access to system	Priority: Must Have
U93. Third party access to the system	Priority: Must Have

3.2 Developers

Backwards compatibility	
U100. Backwards compatibility of apps	Priority: Should Have
U101. Backwards compatibility of services	Priority: Could Have
U102. Backwards compatibility of API	Priority: Should Have
Robustness	
U103. Fault tolerance	Priority: Must Have
U104. Stability	Priority: Must Have
Access to components	
U105. Access to the dialog system	Priority: Must Have
U106. Access to cloud services	Priority: Must Have
Access to sensors	
U107. Access to sensors of the vehicle (via car APIs) when the functionality is available. Types: fuel consumption, speed, acceleration	Priority: Should Have
U108. Access to smart device sensors	Priority: Must Have
U109. Access to remote sensors, e.g. traffic sensors	Priority: Must Have

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Access to car components	
U110. Remote control of car components when the functionality is available, e.g. air conditioning, heating, battery charge timing	Priority: Will Not Have For Now
U111. Provision of web site of the car	Priority: Could Have
Data sources	
U112. Support of open data	Priority: Must Have
U113. Handling of multimedia data	Priority: Must Have
Data sources - Data services	
U114. Configuration of the frequency of update of the data from data sources	Priority: Must Have
U115. Transformation support	Priority: Must Have
U116. Unified data model	Priority: Must Have
U117. Data filtering	Priority: Must Have
U118. Data correlation	Priority: Must Have
U119. Data summarisation	Priority: Must Have
U120. Handle of data streams	Priority: Must Have
Data sources - Storage type	
U121. Local storage of data	Priority: Should Have
U122. Storage of data in the cloud: binary, semantic and structured	Priority: Must Have
System scalability	
U123. Scalability of the cloud infrastructure	Priority: Should Have
U124. Scalability of the service platform	Priority: Should Have
General standard approach	
U125. Open interfaces	Priority: Should Have
U126. Openness of the system	Priority: Should Have
U127. Extensibility	Priority: Should Have

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Business model - Finance – Apps	
U128. The system allows app developers to make money	Priority: Should Have
U129. The system allows the consortium to make money from apps	Priority: Should Have
U130. Apps are free but supported, i.e. red hat model	Priority: Could Have
U131. Free and premium versions of apps	Priority: Could Have
U132. Guarantee and minimum service: life time, support time	Priority: Could Have
Business model - Finance - Services	
U133. The system allows service developers to make money	Priority: Could Have
U134. The system allows the consortium to make money from services	Priority: Could Have
U135. Services are free but supported, i.e. red hat model	Priority: Could Have
U136. Free and premium versions of services	Priority: Could Have
U137. Guarantee and minimum service: life time, support time	Priority: Could Have
Business model - Sales	
U138. Provision of apps statistics	Priority: Must Have
U139. Provision of services statistics	Priority: Must Have
U140. Promotional aspects	Priority: Could Have
U141. Permit to spread the word	Priority: Could Have
App marketplace	
U142. Quality/checking certification	Priority: Must Have
U143. App publication	Priority: Must Have
U144. App versioning	Priority: Must Have
U145. App removal	Priority: Must Have
U146. Call for developers	Priority: Will Not Have For Now
U147. Easy to publish	Priority: Must Have

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Service marketplace	
U148. Service registration	Priority: Must Have
U149. Service extension and modification	Priority: Must Have
U150. Service management	Priority: Must Have
U151. Service versioning	Priority: Must Have
Service Level Agreement	
U152. SLA support	Priority: Must Have
U153. Usage of an official SLA standard	Priority: Must Have
U154. Simple SLA description standard	Priority: Must Have
Developer studio - SDK - API	
U155. Provision of JAVA API	Priority: Must Have
U156. Provision of C/C++ API	Priority: Will Not Have For Now
U157. Standard programming interface	Priority: Must Have
U158. Easy access to API through the developer studio	Priority: Must Have
Developer studio - SDK - Use case driven development	
U159. Support for dialogue	Priority: Must Have
U160. Support for apps	Priority: Must Have
U161. Support for data services	Priority: Must Have
U162. Support for backend services	Priority: Must Have
Developer studio - SDK	
U163. Easy to use environment	Priority: Should Have
U164. Low investment required	Priority: Should Have
U165. User friendly developer studio interface	Priority: Could Have
U166. Identification of the developer / signature	Priority: Must Have
U167. Hot updates	Priority: Will Not Have For Now

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Developer studio - SDK – Documentation	
U168. Provision of source code examples	Priority: Must Have
U169. Provision of UI templates	Priority: Must Have
U170. Provision of best practices	Priority: Must Have
U171. Provision of tutorials	Priority: Must Have
U172. Provision of guidelines	Priority: Must Have
U173. Provision of examples	Priority: Must Have
Developer studio - Mind-set	
U174. Permit to develop an app in less than a day	Priority: Should Have
U175. Hide complexity from developer	Priority: Should Have
U176. Developer should not need to be skilled on spoken dialogue design	Priority: Will Not Have For Now
Developer studio - Simulator	
U177. Provision of a PMA emulator	Priority: Will Not Have For Now
Developer studio - Functionality description	
U178. Definition of minimum hardware requirements of apps	Priority: Must Have
U179. Description of services: semantic description, keywords, phrasing, ontology	Priority: Must Have
Developer studio – Debugging	
U180. Easy debugging of services	Priority: Could Have
U181. App and service crash reports	Priority: Could Have
Developer studio - Statistics	
U182. Provision of statistics, e.g. usage, traffic	Priority: Must Have
U183. Provision of bug reports	Priority: Could Have
U184. Provision of crash reports	Priority: Could Have
Development - Interaction between apps and services	
U185. Standardised messages between apps	Priority: Must Have

U186. Registry of installed SIMPLI-CITY apps	Priority: Must Have
U187. Composition of apps	Priority: Will Not Have For Now
U188. Composition of services	Priority: Should Have
Development - Interface	
U189. Unified interface for accessing sensors	Priority: Must Have
U190. Unified interface for accessing user centric data	Priority: Must Have
U191. Simulation of sensors	Priority: Must Have
Development	
U192. Service deployment	Priority: Must Have
U193. Exchange of information from apps to server	Priority: Must Have
U194. Exchange of information from server to apps	Priority: Must Have
U195. Proactive user notifications	Priority: Must Have
U196. Prioritization of notifications	Priority: Must Have
U197. Get support from community	Priority: Will Not Have For Now
Cross-cutting concerns	
U198. Small number of device to support	Priority: Must Have
U199. Compliance to regulations	Priority: Should Have

3.3 Use Cases

Generic	
U201. Provision of personalized info, e.g. travel, costs	Priority: Must Have
IBM scenario	
U202. Diagnosis of abnormal traffic condition in real-time	Priority: Must Have
U203. Prediction of abnormal traffic condition	Priority: Must Have
U204. Support for querying diagnosis historic	Priority: Must Have

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

U205. Support for querying impact factor on traffic condition	Priority: Must Have
SRM scenario	
U206. Optimization of financial resources, e.g. avoiding tolls, cheaper parking	Priority: Must Have
U207. Support suggestions for road/trip optimization if conditions change	Priority: Must Have
U208. Possibility to continue a previously started trip planning on the same device or different device	Priority: Must Have
Tempos 21 scenario	
U209. Provision of real-time information about the current route	Priority: Must Have
U210. Reproduction of multimedia information	Priority: Must Have
U211. Reproduction of streaming audio	Priority: Should Have
U212. Notification to end user about the proximity of Points of Interest	Priority: Must Have
U213. Social network integration	Priority: Must Have
CRF scenario	
U214. Reporting to the end user about eco-driving information	Priority: Must Have
U215. Vehicle information available to the system	Priority: Must Have
U216. Provision of real time feedback to the user in order to improve his/her performance	Priority: Must Have
U217. Access to a journey-related eco-driving data using the specific web portal	Priority: Must Have
U218. Comparing (eco-)performances of different drivers	Priority: Must Have
U219. Reward drivers through the eco-driving contest	Priority: Should Have

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

4 Strategic Requirements

The following tables give an overview and short discussion of the strategic requirements as listed in Section 3.

The strategic requirements are drafting the greater strategic goal of SIMPLI-CITY, either in ways of the general target group itself but also high level goals like safety, privacy and environmental and traffic planning issues.

Name: Support of different end-users	**U1: Support of car drivers as end users** **U2: Support of cyclists as end users** **U3: Support of private transport passengers as end users** **U4: Support of public transport users as end users** **U5: Support of pedestrians as end users** **U6: Support of truck drivers as end users** **U7: Support for disabled drivers. It should take into account the interaction between the user and the road information system**	**Priority: Must Have** **Should Have (U2, U3)** **Could Have (U4, U5, U6)** **Will Not Have For Now (U7)**
Analysis and Comments: Car users are the main target group of SIMPLI-CITY, therefore they are given the highest priority. But in order the contribute to greening of transportation and therefore CO2 reduction and improved usage of more environmentally sound modes of transportation, users of other transport modes like cyclists, public transport users and pedestrians are seen as target groups as well. These target groups need to be taken into account also in situations, where car users may possibly change transport mode, e.g. if the car trip is obstructed by congestion or when access restrictions are in place. However, transport modes other than the car need better data sources: public transport needs real time data, which is not available in sound enough quality everywhere, and also routing information for pedestrian routing needs to be much more detailed than routing information for car routing. The priority assigned to the other modes is lower, because the whole functionality will not be available from the beginning but the mechanisms will be incorporated in the system.		

Name: Multiuser support	**U8: Multiuser support. It should support multiple drivers and different users**	**Priority: Will Not Have For Now**
Analysis and Comments: The SIMPLI-CITY platform will support multiple users along with their specific basic data and user preferences.		

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Green users	U9: Allow users to become green	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY will interrelate to the carbon footprint of the trips in several ways: • Direct interface to the in-car diagnostics (e.g. “you have enough time to reach your meeting, drive 20 km/h slower to save X of CO_2”) • (Re)routing that bypasses congestion • (Re)routing to other, more environmentally friendly modes		

Name: Reduction of accidents	U10: Reduction of the number of accidents	Priority: Must Have
Analysis and Comments: Due to real time information the system will influence the safety of driving positively by e.g. warning of hazardous road conditions or avoiding dangerous areas. The user interfacing will be designed to be non-distracting and requiring a minimum of interaction so that the driver keeps his concentration.		

Name: Useful information to drivers	U11: Provision of useful information to drivers	Priority: Must Have
Analysis and Comments: The SIMPLI-CITY apps will aim to give useful information to the user that is either requested by her/him (e.g. “Where is the next parking lot?”) or implicated by the specific situation (e.g. “if you want to be at your meeting in time, you should change to the train at the station XY because the road ahead is blocked”). However, the SIMPLI-CITY apps are not aiming to give information consisting of things like advertising (e.g. “turn left because there is a 10% off sale at XY”) etc., or information not relevant in the specific context (e.g. information about traffic incident in another area).		

Name: No distraction of the driver	U12: No distraction of the driver	Priority: Must Have
Analysis and Comments: As the main target group of SIMPLI-CITY are car drivers, safety is the crucial concern. In order to keep the distraction of the driver at a minimum a natural speech interface is the main way of interaction with the system when in movement. This will ensure a low distraction. Basic configuration of the system will use a full size interface, but basic configuration is supposed to be done in advance of the trip.		

Name: No SIMPLI-CITY big brother	U13: No SIMPLI-CITY big brother, i.e., no excessive data collection beyond the necessary scope	Priority: Should Have
Analysis and Comments: In most cases the SIMPLI-CITY apps need to know user preferences and geo-location of the user. This, per-se leads to an amount of data that can lead to the creation of user patterns and personal data. All (pseudo) anonymisation does not seem to be sufficient where trip data and movement patterns come into play, because these data can easily be interpolated and lead to the address of home and work of the user. But this is the case of most of the applications in place nowadays. SIMPLI-CITY will therefore aim to protect the data in the core system, so that the risk of unwanted third party		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

access and privacy breach is minimised. Moreover, only the minimum required data will be collected.

Name: Real time carbon print	U14: Show real time carbon print	Priority: Must Have
Analysis and Comments: The calculation and display of the real time carbon print gives a direct feedback to the driver in real time. This is made possible with the direct interfacing to in-car diagnostics. For other modes generic calculation methods will be used.		

Name: Traffic control	U15: Help to balance the traffic in a particular area	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY will allow to balance the traffic in certain areas (e.g. by specific re-routing of traffic flows).		

Name: Ordered driving	U16: Provide ordered driving in the city	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY will take into account access and other traffic restrictions, and will provide the driver with the information needed to behave in line with the traffic laws and depending on the current traffic of the road.		

Name: Allow local authorities to connect easily with users	U17: Allow local authorities to connect easily with users	Priority: Must Have
Analysis and Comments: Local authorities need to be provided with an interface to the system in order to give the users information regarding access restrictions, (temporal) speed limits or general traffic incidents. Road users need to access this information provided by the local authorities through the SIMPLI-CITY platform.		

Name: Gateway to data sources	U82: Gateway to data sources	Priority: Will Not Have For Now
Analysis and Comments: The SIMPLI-CITY platform should act as a gateway to external data sources, i.e. it should permit a user to access external data sources through the platform and consume these data sources within end-user applications.		

5 Functional Requirements

In this section, the collected requirements as listed in Section 3 will be discussed in more details, including a brief analysis of each requirement, which serves as a foundation for the forthcoming SIMPLI-CITY deliverables Global Architecture Definition (D3.1), and most importantly, the Functional Specification (D3.2.1) and Technical Specification (D3.2.2).

In the following subsections, the analysis and description of closely related requirements may be combined, if this is reasonable. Requirements are either linked to specific RTD work packages (WP4, WP5, or WP6). Within the single work packages, they are further linked to specific tasks which will play a role in the realization of the requirement. In many cases, more than one task will be involved in the realization of a requirement; the involved tasks may be from the same work package but may also be from different work packages. In this analysis, each requirement is therefore linked to one primary task, i.e., the task that will have the biggest efforts in the requirement's implementation. Furthermore, one or more *secondary* task assignments may be added if applicable, i.e., tasks that also have efforts in the requirement's implementation besides the primary task assignment.

5.1 Generic

Name: User centric data services	**U59: User centric data services**	**Priority: Must Have**
Analysis and Comments: Services must be able to integrate user-related information in order to personalize its content depending on this information. User-related information may include among others the user profile, Facebook profile, calendar events and data from smartphone sensors.		
Task Assignment: • T4.4: Must provide the basics for contextualizing information using end-user information Secondary Task Assignment(s): • T5.2: Must integrate user-related information into the services outcome • T5.1: Must permit to configure the integration of user-related data into services		

Name: PMA size	**U77: The PMA fits into the pocket**	**Priority: Will Not Have For Now**
Analysis and Comments: The size of the PMA should be small; ideally it should fit in a pocket. Thus, the technologies used to develop the PMA have to permit to execute the PMA in a small device.		
Primary Task Assignment: • T9.1: The exploitation plan has to evaluate mobile devices in which the PMA can be executed Secondary Task Assignment: • None		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: PMA battery lifetime	U78: The battery lifetime of the PMA is longer than 1 day	Priority: Will Not Have For Now
Analysis and Comments: The PMA and the applications developed for the SIMPLI-CITY platform should be developed taking into account that they consume few resources of the device. And thus allowing that the lifetime of the battery has a long duration (more than one day).		
Primary Task Assignment: • T6.3: The PMA should optimize the use of resources of the device Secondary Task Assignment: • T7.2, T8.2: Apps should be written in a way that they don't compute unnecessarily		

5.2 Related to T3.3 Holistic Security and Privacy Concept

Name: Secure Transmissions	U90: Availability U91: Integrity U92: Secure access to system U93: Third party access to the system	Priority: Must Have
Analysis and Comments: Transmissions between the Personal Mobility Assistant (PMA) and the service backend, but also between the service backend and data sources should be secure. Therefore, SIMPLI-CITY needs to provide mechanisms to make transmissions secure in terms of confidentiality, integrity, availability, and secure access to the system in general. Especially, third parties should get a secure access to the system.		
Primary Task Assignment: • T3.3: Must provide security concepts which is to be followed by all components Secondary Task Assignment: • T4.2: Must be able to accept incoming secure connections • T4.3: Must permit for a secure communication between sensors and database backend, where applicable and technically possible • T5.3: Must allow that secure communication is established to it (by the PMA). Must also be able to establish secure communication to data sources • T6.3: Must be able to establish secure communication to backend services • T6.4: The PMA part must always communicate over secure connection		

Name: Compliance	U199: Compliance to regulation	Priority: Should Have
Analysis and Comments: Data could be subject to licences. It is important for SIMPLI-CITY to make sure that data which is processed and stored in the Cloud-based Information Infrastructure is compliant with regulations and licensing rules.		
Primary Task Assignment: • T3.3: Must ensure data can be stored in the cloud under the licence contract		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment: • None

5.3 Related to WP4 Mobility-related Data as a Service

5.3.1 Related to T4.1 Privacy-Aware Data Modelling and Access Framework

Name: Data Privacy-Protection	U86: Transparency U87: Confidentiality. Does not give away data to third parties U88: Data encryption U89: Certification. Only certified apps are allowed to access users data	Priority: Must Have
Analysis and Comments: In case of sensitive data, privacy-protection techniques are used to make sure that sensitive information will not be leaked. This ensures data transparency, data confidentiality through data encryption and certification for apps.		
Primary Task Assignment: • T4.1: Data privacy aspects must be integrated into the common data representation layer Secondary Task Assignment: • T3.3: Must provide the basic concepts for data privacy and data protection • T4.3: Sensor interfaces must ensure privacy through appropriate technical mechanisms • T4.4: User-centric and Open Data access must be privacy-aware • T5.3: Must support the usage of data access control during service execution		

Name: Data Transformation, Filtering and Correlation	U115: Transformation support U117: Data filtering U118: Data correlation	Priority: Must Have
Analysis and Comments: SIMPLI-CITY must provide basic functionalities for correlating data from sensors, so basic operations can be easily achieved on multiple sources of information. Examples of transformation are: splitting, merging, filtering, and aggregation of information in a context of heterogeneous data (both format and up-to-date).		
Primary Task Assignment: • T4.1: Must support basic correlation (cross and auto-correlation) and basic integration of data, i.e. data splitting, merging, filtering, and aggregation Secondary Task Assignment: • T4.3: Must provide basic functionalities for data aggregation and composition • T4.4: Must allow contextualization of information		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Data Representation	U116: Unified data model	Priority: Must Have
Analysis and Comments: SIMPLI-CITY needs to provide the means to represent heterogeneous mobility-related data in a ready-to-be-integrated framework (e.g. through spatial, temporal and dynamic representation – mainly for chronological tracking), following semantic representation standards. Since space and time are both important features in city- and mobility-related data, SIMPLI-CITY should strongly focus on the spatiotemporal representation of the data model. In order to allow service developers to easily access data coming from different sources, all data sources should be integratable into software services using a similar approach and similar API controls, following the Mobility Data as a Service approach. Apart from the unified access model, this means that for different data type wrappers need to be provided, so that at the end of the data there is one unified access model for sensors as well as other data sources.		
Primary Task Assignment: • T4.1: SIMPLI-CITY must use the common representation layer for basic integration and allow to access data using a unified access model Secondary Task Assignment: • T4.2: Must provide and store data in the appropriate representation format and must also store the common representation layer itself • T4.3: The Sensor Abstraction and Interoperability Layer must provide according wrappers for data sources • T4.4: Must use the common representation layer for contextualization • T5.1: The Service Development API must take into account the unified data access model		

Name: Data Summarisation	U119: Data summarisation	Priority: Must Have
Analysis and Comments: As sensors generate a large amount of data, the SIMPLI-CITY Cloud Information Infrastructure cannot store all data. Thus, techniques for summarizing and keeping the most relevant data are required.		
Primary Task Assignment: • T4.1: Must provide the mechanisms for summarizing sensor data Secondary Task Assignment: • T4.2: Must provide the basics for storing summarized data • T4.3: Must provide basic functionalities for data aggregation and composition		

5.3.2 Related to T4.2 Cloud-based Information Infrastructure

Name: User Profile Storage	U80: Profile in the cloud	Priority: Must Have
Analysis and Comments: SIMPLI-CITY will support personalization of services and apps. As such, personal profiles need to be either stored or always provided by some particular source (e.g., the PMA). The Cloud-based information storage will be fully content agnostic and can therefore be used for storing user profiles. However, since personal information is very sensitive in terms of privacy, the cloud storage has to provide a way of storing data in an isolated way so that it can't be accessed by non-authorized components. This may be achieved by either providing isolated data storage spaces or by using encryption to ensure that data can		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

only be decrypted by a certain application. Hence, this requirement is closely related to requirements U85-U87.
Primary Task Assignment: • T4.2: The Cloud-based Information Infrastructure should provide a way for securely storing information and for encrypting data or for using other techniques to provide data isolation for protecting sensitive information Secondary Task Assignment: • T3.3: Needs to define basic requirements for data privacy • T4.1: Data privacy aspects need to be regarded within the SIMPLI-CITY data model • T6.3: Needs to be able to store the profile in the cloud

Name: Cloud-based Data Storage	U81: Storage of data in the cloud U122: Storage of data in the cloud: binary, semantic and structured	Priority: Must Have
Analysis and Comments: A scalable approach is required for storing data in the cloud. This storage should be able to store different types of data. For the purpose of the project, a binary storage, a semantic storage and a structured (NoSQL-like) storage are foreseen, which will cover all different data types regarded within SIMPLI-CITY. The cloud storage will provide basic CRUD (Create, Read, Update and Delete) functionality for all those storage types as well as an extended interface for type-specific requests.		
Primary Task Assignment: • T4.2: Must provide the basics for creating, requesting, updating, and deleting collected data Secondary Task Assignment: • None		

Name: Local Storage	U121: Local Storage of data	Priority: Should Have
Analysis and Comments: The PMA should provide a local storage facility as part of T4.2, allowing apps to store information based on key-value pairs.		
Primary Task Assignment: • T4.2: Needs to provide a local storage facility the PMA and apps can make use of to store app- or user-related data. Secondary Task Assignment: • T6.3: Should provide a wrapper for easily accessing the local storage		

Name: Scalability	U123: Scalability of the cloud infrastructure	Priority: Should Have
Analysis and Comments: One of the main advantages of cloud-based data storage is its scalability. This allows developers to store information without having to worry about data distribution or bandwidth aspects. As such, the SIMPLI-CITY data storage will support scalable data storage and will automatically distribute the load among different servers if needed.		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

It should, however, be noted that data that is stored in the cloud storage can only be transferred to the services/apps as fast as the connection is. This means that large amounts of data should not be stored or requested if it is avoidable, since data bandwidth for mobile devices is limited and may be expensive for the end user.
Primary Task Assignment: • T4.2: Must ensure that data storage may be distributed among different servers if needed Secondary Task Assignment: • None

5.3.3 Related to T4.3 Sensor Abstraction and Interoperability Interfaces

Name: Access to Vehicle Data	U85: Interaction with car sensors U107: Access to sensors of the vehicle	Priority: Must Have Should Have (U107)
Analysis and Comments: For apps and backend services based on vehicle data in an intuitive way, SIMPLI-CITY needs to provide unified access to sensor readings of sensors in the car, and to allow, e.g., to get information about fuel levels, fuel consumption, actual speed, in- and outside temperature or on-board voltage. For this, the concept of Mobility-related Data as a Service will be applied within the project. This information can be obtained by means of standardised interfaces to the vehicle systems. Note that this functionality will only be available when the car provides access to the sensors.		
Primary Task Assignment: • T4.3: Must allow access to sensor readings of the vehicle Secondary Task Assignment: • T4.1: Most provide a unified model for data description and access		

Name: Access to vehicular HMI	U83: Interaction with head up display	Priority: Will Not Have For Now
Analysis and Comments: Modern cars have several built-in Human-Machine Interface (HMI) systems for displaying information to the driver, e.g., a display in the head unit, a head up display or the displays of the multimedia system. SIMPLI-CITY needs to provide a unified interface to enable the Application Runtime Environment to transfer information to the vehicular to display on built-in user interfaces. This can be done by interfacing available data buses of vehicles, but needs the availability of respective interfaces in the proprietary systems of the car manufacturers.		
Primary Task Assignment: • T4.3: Must allow sending information to the vehicle's user interfaces		

Name: Access to Device Data	U108: Access to smart device sensors	Priority: Must Have
Analysis and Comments: Many apps and services make use of data coming from the smart device the PMA is running on, e.g., temperature, acceleration, GPS, or luminance data. Hence, SIMPLI-CITY needs to provide the means to easily configure these sensors; furthermore, hardware constraints of the mobile device		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

have to be taken into account.
Primary Task Assignment: • T4.3: Must allow access to sensor readings of the smart device the PMA is running on Secondary Task Assignment: • T4.1: Most provide a unified model for data description and access

Name: Access to Remote Sensors	U109: Access to remote sensors	Priority: Must Have
Analysis and Comments: SIMPLI-CITY-enabled apps and services should be provided with the possibility to get data from various sensor sources, most importantly traffic sensors, but also other kinds of sensors. For this, the concept of Mobility-related Data as a Service will be applied within the project. In order to unify access to different sensor types, a unified data description and access model is needed.		
Primary Task Assignment: • T4.3: Must allow access to sensor readings of external sensors in the environment Secondary Task Assignment: • T4.1: Most provide a unified model for data description and access		

Name: Remote Control	U110: Remote control of car components, e.g. air conditioning, heating, battery charge timing	Priority: Will Not Have For Now
Analysis and Comments: Some functionality to increase the users comfort or benefit needs the remote control of some car components. For example this can be the remote activation of the cars heating system before the user arrives at the car or the remote controlled charging of the batteries of electric cars depending on the current energy prices. Developers should be provided with a unified way to send predefined command to the vehicle. This command flow can be achieved by means of car APIs of vehicles. Note that this functionality will only be available when the car provides remote control fucntionality.		
Primary Task Assignment: • T4.3: Must allow to send commands to the vehicle Secondary Task Assignment: • T4.1: Must provide a unified model for command description		

Name: Data Source Update Frequency	U114: Configuration of the frequency of update of the data from data sources	Priority: Must Have
Analysis and Comments: Data stream can be with high throughput and processing all the data is out of scope of the project. In order to manage such streams, their access should be configurable e.g., through their frequency of update		
Primary Task Assignment: • T4.3: Must interface the sensor and the stream		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment:

- T4.1: Data model must support the description of the frequency of update
- T4.2: Must interpret the frequency of update to access the stream
- T5.1: Must allow service developers to define the needed update frequency for their backend services

Name: Data Streams	U120: Handle data streams	Priority: Must Have
Analysis and Comments: The SIMPLI-CITY infrastructure will need to be able to handle data streams, coming from sensors (e.g. temperature or fine dust sensors). In particular, manipulation and transformation of data streams using a unified data model will be handled by SIMPLI-CITY.		
Primary Task Assignment: • T4.3: Must provide basic functionalities for interfacing data streams Secondary Task Assignment: • T4.1: Must provide the basics for handling data streams • T4.2: Must provide the basics for playing data streams		

Name: Sensor Simulation	U191: Simulation of sensors	Priority: Must Have
Analysis and Comments: In order to simplify the development process with respect to the integration of sensor data, i.e., enable developers to create services without requiring access to actual data sources, SIMPLI-CITY should provide a basic sensor simulation. Simulated sensors constitute a simplified version of an actual sensor; specifically, they provide data in an identical format as the actual sensor, but do not necessarily provide fully accurate or realistic readings. Hence, historic or fictive data may be used as the basis of the sensor simulation.		
Primary Task Assignment: • T4.3: Must provide simulated sensors and according services Secondary Task Assignment: • T5.1: The API must permit for access to simulated sensors in addition to actual sensors		

Name: Unified Sensor Interface	U189: Unified interface for accessing sensors	Priority: Must Have
Analysis and Comments: There are various types of sensors which could provide valuable information for SIMPLI-CITY-enabled apps and services. However, even if the sensor types and technologies are heterogeneous, service and app developers should be provided with a unified way to access them.		
Primary Task Assignment: • T4.3: Must allow to access sensor readings by the use of unified interfaces and must provide a common representation for sensor readings		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment: • T4.1: Most provide a unified model for data description and access

5.3.4 Related to T4.4 User-Centric Data and Open Data Management

Name: Support of open data	U112: Support of open data	Priority: Must Have
Analysis and Comments: Basic open data access following the MDaaS paradigm. More importantly by linking data from sensors and open data sources, SIMPLI-CITY-enabled apps and backend services can be provided with contextualized information. Hence, basic functionalities for contextualising information from sensors using open data sources are needed.		
Primary Task Assignment: • T4.4: Must provide the basics for contextualizing information using open data related information Secondary Task Assignment: • T4.3: Must provide an interface for interaction with T4.4		

Name: User-based Sensor Data Contextualization	U190: Unified interface for accessing user centric data	Priority: Must Have
Analysis and Comments: Basic user-centric data access following the MDaaS paradigm. More importantly, analogue to the last requirement, contextualizing sensor data using user-centric data can provide valuable input for SIMPLI-CITY enabled apps and backend services.		
Primary Task Assignment: • T4.4: Must provide the basics for contextualizing information using end-user information Secondary Task Assignment: • T4.3: Must provide an interface for interaction with T4.4		

5.3.5 Related to T4.5 Media Data Streams and Data Prefetching

Name: Offline Access	U50: Prefetching of media data & Offline access	Priority: Must Have
Analysis and Comments: SIMPLI-CITY will allow to prefetch and media data if this is reasonable and will provide the user with a better experience. This is the case if – without prefetching – media playback would be interrupted. For instance, this is the case if the user will enter an area with bad connectivity (e.g., a tunnel), be abroad and does not want to pay roaming fees, or has already used his/her bandwidth cap, but may use his/her home WiFi before starting his journey. In all these cases, a service needs to recognize that there is a need for media data prefetching based on personal information, context data, and app data. The prefetching of non-media data will be regarded separately in U52.		
Primary Task Assignment: • T4.5: Should provide a local buffer for media data. Media data should be automatically downloaded and buffered if needed Secondary Task Assignment: • T4.2: Should support the streaming of data from cloud based data sources		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

- T4.4: Should support the streaming of data from user-centric data sources
- T5.2: Must provide context-based prefetching decisions

Name: Bandwidth Optimization	U51: Avoid the download of data from 3G	Priority: Should Have
Analysis and Comments: Mobile web connections get cheaper and cheaper making the internet almost omnipresent. However, as of today, people usually still pay on a per-use (e.g., by paying each Megabyte that is consumed) or they use flat rate options with limitations (e.g., two Gigabytes per month). As such, the amount of data that is consumed when being on the road should be limited. Additionally mobile web connections tend to be relatively slow which enforces SIMPLI-CITY to optimize the bandwidth, too. Therefore, SIMPLI-CITY should avoid the consumption of mass data via cellular (3G/4G) connections. For this purpose, media data should be provided in an adoptable way based on the current connection of the user		
Primary Task Assignment: • T4.5: Should provide a server side component to deliver media data in different qualities based on the current user connections. Should also allow users to use the PMA to prefetch media data before leaving the local broadband connection (e.g., before starting a road trip). Secondary Task Assignment: • None		

Name: Data Prefetching	U52: Offline access of data used within apps	Priority: Should Have
Analysis and Comments: SIMPLI-CITY should act as an intelligent data prefetcher which allows users to instantly access data when it is needed. This means that SIMPLI-CITY should automatically prefetch information from data services before they are needed whenever possible. For example, SIMPLI-CITY may automatically load a list of free parking lots when entering Barcelona city. Prefetching of data should be accessible by apps when if the PMA loses its connection (i.e., offline data access). The prefetching of media data is a separate requirement; nevertheless, work on the realization of these two requirements will naturally be very closely related.		
Primary Task Assignment: • T4.5: Should automatically detect data that might be needed and should automatically prefetch it Secondary Task Assignment: • T4.2: Should support the streaming of data from cloud based data sources • T4.4: Should support the streaming of data from user-centric data sources • T5.2: Must provide context-based prefetching decisions		

Name: Multimedia Support	U113: Handling of multimedia data	Priority: Must Have
Analysis and Comments: SIMPLI-CITY will build the next generation of road user information systems and will provide users with wide possibilities for accessing services and data while they are on the road. Data may be information that is read to the user using the Text-to-Speech features of WP6. However, it may also be multimedia information such as an app that acts as a tourist guide with pictures, videos or music elements inside. This will require the support of the PMA to handle those media formats and to manage them in a non-		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

distracting way. Additionally, users may want to access their personal music data source, e.g. fore listening to music while they are driving. This will also require the capability of SIMPLI-CITY to access and handle multimedia data.
Primary Task Assignment: • T4.5: Should provide handlers for popular media formats including music, video and image formats Secondary Task Assignment: • T4.2: Should support the streaming of data from cloud-based data sources • T4.4: Should support the streaming of data from user-centric data sources

Name: App Recommendations	U53: App Recommendations	Priority: Could Have
Analysis and Comments: As described in U52, SIMPLI-CITY should act as an intelligent data prefetcher. This prefetched data should then be connected to the correct app and users should be informed. For example, the PMA may inform the user by stating “Dear user, many people are using the ‘free parking spaces’ app when entering Barcelona. Shall I launch it for you now?”		
Primary Task Assignment: • T4.5: Should automatically detect apps that are connected to prefetched data and should recommend those apps to the user in case that they are installed on the PMA Secondary Task Assignment: • None		

Name: Expiration of Data	U54: Expiration of Data	Priority: Should Have
Analysis and Comments: The PMA is limited in terms of storage space. As such, SIMPLI-CITY should handle this space wisely and only buffer content that is actually needed. Therefore, data should expire when being too old and when not being used for a certain amount of time in order to free up space. Additionally, some data might have an explicit “expiration date”. For example, information about free parking lots is only useful if it is not older than, e.g., 20 minutes. Afterwards, the data can be considered to be obsolete and may automatically be deleted from the PMA.		
Primary Task Assignment: • T4.5: Should support the tagging of data with expiration dates and should automatically free up space if needed Secondary Task Assignment: • None		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

5.4 Related to WP5 Mobility Services Framework

5.4.1 Related to T5.1 Service Development API

Name: Data and backend services	U161: Support for data services U162: Support for backend services	Priority: Must Have
Analysis and Comments: The Service Development API should allow the creation and configuration of data and backend services. Data services refer to the access to external data sources, provided by WP4 components as MDaaS-enabled data sources. The developer should be able to define the different configuration parameters of these data sources. On the other hand, backend services provide more complex functionality, like the combination of different data services (composition) or the inclusion of context-related information.		
Primary Task Assignment: • T5.1: Must provide the means to create and configure data and backend services Secondary Task Assignment: • T5.3: Must permit to register a created data and backend service		

Name: Service Management	U149: Service extension and modification U150: Service management	Priority: Must Have
Analysis and Comments: The Service Development API should permit to manage and modify a previously created data service or backend service, including the edition and modification of the description of the service and the configuration parameters. It should also be possible to delete created services from the registry and the marketplace. The actual storage of service versioning information is covered by U151.		
Primary Task Assignment: • T5.1: Must provide the means to manage and modify previously created services Secondary Task Assignment: • T5.3: Must support modification of services and service metadata, including deletion of services		

Name: Developer Studio	U163: Easy to use environment U164: Low investment required U165: User-friendly developer studio interface	Priority: Should Have Could Have (U165)
Analysis and Comments: The Service Development API should provide a user-friendly web interface allowing service developers to create and register their own services. This web interface should permit to introduce information about the backend services to be created like the name and description, and configure data services with parameters like the update frequency. Last but not least, the interface should not make it necessary to invest heavily to make use of it. This means that standard technologies should be applied and the software itself should be free of charge.		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Primary Task Assignment:

- T5.1: Must provide the means to create services in a user-friendly, intuitive way

Secondary Task Assignment:

- None

Name: Service Developer Signature	U166: Identification of the developer/signature	Priority: Must Have
Analysis and Comments: The Service Development API should store the identification information of the service developer that created a service and provides this information to other users.		
Primary Task Assignment: • T5.1: Must identify the developer of a service Secondary Task Assignment: • T5.3: The Service Registry must permit to store and provide information about the developer of a service		

Name: Service Updates	U167: Service hot updates	Priority: Will Not Have For Now
Analysis and Comments: The Service Development API should support hot-deployments of a service. This facilitates that no interruption of the service is needed during the modification of its parameters. This allows the apps that use the service to operate normally and to not be affected by these modifications.		
Primary Task Assignment: • T5.1: Must provide the means to update services in real time Secondary Task Assignment: • T5.3: The Service Runtime Environment must provide the means to allow for service hot updates during service runtime • T5.3: The Service Registry must provide the means to allow to register service hot updates		

Name: Service Development Kit Documentation	U168: Provision of source code examples U169: Provision of UI templates U170: Provision of best practices U171: Provision of tutorials U172: Provision of guidelines U173: Provision of examples	Priority: Must Have
Analysis and Comments: The Service Development API should support service developers during the service creation process through according documentation and examples. This support should include the provision of source code examples, user interface templates, general guidelines, tutorials and examples of the overall process of creation of services, including the configuration of data services.		

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Furthermore, a unified service model including standard methods and a metadata model needs to be drafted.
Primary Task Assignment: • T5.1: Must provide resources to support developers during the creation of services Secondary Task Assignment: • None

Name: Service Description	U179: Service Metadata	Priority: Must Have
Analysis and Comments: The created services should be described using metadata. For data services and location-aware backend services, one important metadata parameter is the actual location of the underlying data sources. Data services should also be categorised using a basic data type taxonomy which indicates the content of the provided data in a semantic way. In general, metadata allows categorizing services, and to search and find services; this is an important feature if it is foreseen that services should be reused by app and service developers. This can be either achieved through some (ontology-/taxonomy-based) semantic description, keywords, phrasings, or a combination of them.		
Primary Task Assignment: • T5.1: Must provide the means to describe services using metadata Secondary Task Assignment: • T5.3: Must provide the means to search for and provide services based on metadata; this includes the possibility to store the metadata in the Service Registry		

Name: Service Debugging	U180: Easy debugging of services U181: Service crash reports U183: Provision of bug reports U184: Provision of crash reports	Priority: Could Have
Analysis and Comments: The Service Development API should provide debugging functionalities that allow developers to debug services and to be informed about the correct operation or about any problem occurred during the process. This includes information about the availability of used data services (i.e., are the data sources alive) and the provision of bug reports and crash statistics/reports to the service developer.		
Primary Task Assignment: • T5.1: Must provide debugging functionalities and according information useful to the service developer Secondary Task Assignment: • T5.3: The Service Registry should provide status information (i.e., information about the availability) of data services by regularly checking this		

Name: Service compositions	U188: Composition of services	Priority: Should Have
Analysis and Comments: Service compositions combine different (project-internal and -external) services with each other in order to provide added functionality. Instead of hard-coding this inside an end user app, modelling and execution of service compositions should be explicitly provided SIMPLI-CITY services framework. This way, the backend service can cater for the orchestration of different services and the PMA functionality stays relatively lightweight.		
Primary Task Assignment: • T5.1: The definition of the API must allow service compositions Secondary Task Assignment: • T5.3: The SRE must be able to invoke (data) services from within other services		

Name: Community Support	U197: Community support for service developers	Priority: Will Not Have For Now
Analysis and Comments: The Service Development API should provide the means to create a community of developers. This community should allow the different developers to interact between them and to ask and answer questions regarding the creation of services.		
Primary Task Assignment: • T5.1: Must provide the means to create a community of developers Secondary Task Assignment: • None		

5.4.2 Related to T5.2 Context-based Service Personalisation

Name: Proactive Notifications	U27: Proactive behaviour U195: Proactive user notifications	Priority: Must Have
Analysis and Comments: Apart from reacting to particular user requests, services (and apps) should also be able to proactively provide information to the user, e.g., in case there is an important update or if the user has predefined that he/she wants to be reminded of ecological driving behaviour. For this, SIMPLI-CITY needs to calculate a value indicating the importance of a particular piece of information. In case a particular threshold is met, information should be forwarded to the user. Factors that need to be taken into account are the importance and value of the information.		
Primary Task Assignment: • T5.2: Must recognize which information is relevant to the user to a particular degree Secondary Task Assignment: • T5.3: Needs to enable services to forward information to the end user app • T6.3: Must also be able to display notifications when they have been received (via T6.2).		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Prioritization of Notifications	U196: Prioritization of notifications	Priority: Must Have
Analysis and Comments: Not all possible notifications from backend services (and therefore PMA apps) should be sent to the end user in all situations. This could lead to the distraction of the end user or to the point where notifications are ignored pretty much all the time. Instated, notifications should be prioritized with regard to their importance to the driver in a particular context or situation. This prioritization should be based on the value indicating the importance of a particular piece of information in a certain context, as it is foreseen to achieve requirements U27 and U195.		
Primary Task Assignment: • T5.2: Must recognize which information is relevant to the user to a particular degree Secondary Task Assignment: • T5.3: Needs to enable services to forward information to the end user app • T6.3: Must provide information about the user status, i.e., if she is currently distracted. Must also be able to display notifications when they have been received (via T6.2).		

5.4.3 Related to T5.3 Service Runtime Environment or the Service Registry

5.4.3.1 Primarily Related to the Service Runtime Environment

Name: Fault Tolerance	U103: Fault tolerance	Priority: Must Have
Analysis and Comments: Service execution should be robust, i.e., dependable with respect to external faults, e.g., loss of communication, missing responses by data sources, etc. Fault tolerance is the ability to detect errors and recover from them. It does not take care of fault prevention, i.e., fault tolerance mechanisms are only applied when a fault has already occurred. In SIMPLI-CITY, error detection will be based on monitoring capabilities as well as handling of error codes (e.g., a data source could not answer and a service would therefore get a HTTP status 503 – Service unavailable). Recovery will have to provide both error handling, i.e., eliminate the error from the system state, and fault handling, i.e., prevent faults from being experienced again. However, it is not possible to eliminate faults in any case; for this, the fault has to occur within the control sphere of SIMPLI-CITY. Faults occurring, e.g., at a particular data source cannot be resolved.		
Primary Task Assignment: • T5.3: Must provide general fault tolerance mechanisms for services Secondary Task Assignment: • T6.3: Must provide the possibility for apps to request a particular service several times		

Name: Stability	U104: Stability	Priority: Must Have
Analysis and Comments: Fault tolerance (U103) is needed when a fault has already occurred. However, it should be the aim of SIMPLI-CITY to prevent faults to occur from the very beginning. For service executions, this means that stability should be facilitated with regard to communication channels and general application of stability mechanisms during the development of the Service Runtime Environment.		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Primary Task Assignment:

- T5.3: Must include general policies and best practices to achieve the stability of the system

Secondary Task Assignment:

- None

Name: Integration of Data from the Cloud	U106: Access to cloud services	Priority: Must Have
Analysis and Comments: The Cloud-based data storage allows to access arbitrary data sources. Furthermore, it provides functionalities for data analysis, filtering, etc. (see requirements U117-U119). Both data source access as well as the data services should be available to backend services.		

Primary Task Assignment:

- T5.3: Must be able to access Cloud-based data services

Secondary Task Assignment:

- T5.1: Must be able to provide the means to integrate Cloud services into backend services
- T4.2: Must provide the basic Cloud services

Name: Scalability	U124: Scalability of the service platform	Priority: Should Have
Analysis and Comments: It should be made sure that the Service Runtime Environment is able to scale, even if a huge number of services are invoked at the same time. Hence, the service platform should be able to allocate new resources to a service as well as the platform itself if necessary.		

Primary Task Assignment:

- T5.3: Must be able to scale in order to serve many service requests at the same time

Secondary Task Assignment:

- None

Name: General standard approaches	U125: Open interfaces U126: Openness of the system U127: Extensibility	Priority: Should Have
Analysis and Comments: As promised in the Description of Work (DoW), the SIMPLI-CITY Mobility Services Framework is applicable in different domains and easily extendable and customizable. For this, the framework should not rely on specific technologies, should provide open interfaces to all of its components, should be open to all interested parties, and provide the means to extend the single components as well as the overall framework.		

Primary Task Assignment:

- T5.3: Must provide the means to extend and access the service platform and all of its components

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment: • T5.1: Should provide access to all components of the services framework

Name: Service Deployment	U192: Service deployment	Priority: Must Have
Analysis and Comments: Backend services provide the necessary functionality needed for the end user apps running in the PMA. Backend services may be either project-external, i.e., provided by third parties outside the SIMPLI-CITY domain, or project-internal, i.e., running inside the service backend. The latter makes it necessary to provide a Service Runtime Environment. Project-external services make it necessary to provide proxy functionalities within the Runtime Environment, e.g., for service monitoring. In any case, the Service Runtime Environment has to cater for the binding of services. Whenever a service invocation is triggered (either by another service or by an end user app), the Service Runtime Environment gets information that is sufficient to identify the particular backend service that should be invoked. There is only one backend service that meets the identification information (tight coupling). Nevertheless, the Service Runtime Environment needs to look up the service information (based on the identifier) in order to find the information necessary to invoke the backend service, i.e., information about the service interfaces, operations, and its endpoints.		
Primary Task Assignment: • T5.3: Must include a Service Execution Framework; for this, an existing solution like Apache Tomcat may be reused Secondary Task Assignment: • T6.3: Must be able to invoke services in the backend		

5.4.3.2 Primarily Related to the Service Registry

Name: Service Registration	U148: Service registration	Priority: Must Have
Analysis and Comments: Services and apps need to be registered for different reasons: First, in order to resolve service invocations from the PMA. In SIMPLI-CITY, end user apps will make use of tightly-bound services, i.e., the service to be invoked is known beforehand. Second, apps which should be installed in a PMA need to be stored somewhere along with the respective metadata. Third, project-internal services also need to be stored somewhere, such that in the case of service failure, another service instance can be easily deployed. Fourth, the services and applications should be discoverable by the end user in the market in which they are exposed/published.		
Primary Task Assignment: • T5.3: Must provide app/service *registry* functionalities, i.e., the possibility to store app and service metadata (including Service Level Objectives and Agreements) and make use of it. Must also provide app/service *repository* functionalities, i.e., the possibility to store and retrieve service objects (e.g., as WAR archives) and apps. Secondary Task Assignment: • T5.1: Must provide service submission functionalities • T6.4: Must provide app submission functionalities		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Service Versioning	U101: Backwards compatibility of services U151: Service versioning	Priority: Must Have Could Have (U101)
Analysis and Comments: Service developers may develop new functionalities and new versions of their provided services (see U149). Hence, the service registry should provide the means to update the version of a service to the app developers who make use of them. The developers that have the service already integrated should be notified of the availability of the new version of the service and be able to use both versions of these service, thus realizing backwards compatibility of services through the provisioning of different versions, if necessary. This makes it also necessary to store information about the compatibility of different service versions in the Service Registry. SIMPLI-CITY will distinguish two types of versions: A sales technical version which will be used by the registry and the service runtime environment for technical integration and a sales version, which will be used by the service market. The sales version may be based on marketing decisions and may cover multiple technical versions. For example, a sales version might be “Car Sharing Service 2014” while the technical version may be “Build 2014.103”.		
Primary Task Assignment: • T5.3: Must provide app/service *registry* functionalities, i.e., the possibility to store app and service metadata (including Service Level Objectives and Agreements) and make use of it. Must also provide app/service *repository* functionalities, i.e., the possibility to store and retrieve service objects (e.g., as WAR archives) and apps. Secondary Task Assignment: • T5.4: The marketplace should provide versioning information to the service developer • T5.1: Should integrate the versioning information into the API		

Name: SLA Support	U152: SLA support	Priority: Must Have
Analysis and Comments: Service Level Objectives (SLOs) are defined by service requesters – in the case of SIMPLI-CITY, software developers making use of services in other services or in apps. By offering particular Quality of Service (QoS) levels, service providers and consumers agree on a Service Level Agreement (SLA). SLAs are helpful when estimating whether the performance of a service is sufficient and therefore needed in fault tolerance (see U103). SIMPLI-CITY should, in any case, provide the means to define and store SLAs. Moreover, the SLA objectives of each service should be monitored in order to check if the SLA obligations are violated.		
Primary Task Assignment: • T5.3: Service Registry must be able to store SLA information. The framework needs to be able to monitor the (quantifiable) parameters defined in an SLA and start according countermeasures, if necessary. Secondary Task Assignment: • T5.1: Should integrate the possibility to define SLAs • T5.4: The marketplace should provide SLA information to the service requester		

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: SLA Format	U153: Usage of an official SLA standard U154: Simple SLA description standard	Priority: Must Have
Analysis and Comments: SLAs and SLOs need to be described in an intuitive format and/or a format that has been standardised. In SIMPLI-CITY, the usage of a lightweight SLA description standard will be preferred over the usage of an official standard.		
Primary Task Assignment: • T5.3: Will define a particular, lightweight SLA format Secondary Task Assignment: • None		

5.4.4 Related to T5.4 Mobility Service and Application Marketplaces

Name: App Ratings	U55: Rating of apps	Priority: Should Have
Analysis and Comments: Users should be able to provide feedback to an app by rating apps and possibly even commenting them. Consumers may be able to rate services and to provide feedback in terms of comments. This will allow other developers to judge the quality and usefulness of a service based on other users. App marketplaces such as the Apple AppStore or the Google Play Market have shown that this can lead to valuable information and will also give an indicator about the reliability of a service. Feedback should be accompanied with data about the rating, e.g. by specifying the version that is the base for the rating.		
Primary Task Assignment: • T5.4: Should be able to provide a rating UI for consumers (e.g., based on a Likert scale) Secondary Task Assignment: • None		

Name: User Feedback	U56: Feedback to developers through the marketplace	Priority: Should Have
Analysis and Comments: Feedback of users is really important for the developers on their day by day tasks. The feedback should inform about app crashes and improvements. A mechanism for sending/receiving the feedback should be provided. The easiest way for the users to send the feedback is by using the mechanisms already established by other Apps Market such as iTunes or Google Play which is the usage of the marketplaces as channel. In other words, the feedback of the users should be accessible to the app developers via the marketplace.		
Primary Task Assignment: • T5.4: Should be able to provide a UI for developers allowing them to view details about feedback Secondary Task Assignment: • None		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Promotion through Spreading the Work	U57: Social network functionality with the objective of spreading the word U141: Permit to spread the word	Priority: Could Have
Analysis and Comments: Users should be able to recommend apps to friends by using a recommendation form, which will notify friends. Users should be able to recommend apps to friends via social networks such as Facebook. For this purpose, the marketplace will provide social networking buttons allowing users to post feedback to their timeline.		
Primary Task Assignment: • T5.4: Should be able to provide a feature that allows customers to notify other users or friends about an application and it should also provide an optional Facebook posting and maybe a support for additional social networks such as Google+. Secondary Task Assignment: • None		

Name: User recommendations	U58: User recommendations	Priority: Could Have
Analysis and Comments: The marketplace should give users recommendations about apps based on already used apps. For example, if a user has installed a parking app, the marketplace might recommend to also installing an app for public transportation because many users install those two apps together.		
Primary Task Assignment: • T5.4: Should be able to recommend apps to users and services to developers. Secondary Task Assignment: • None		

Name: Payment	U60: Payment for apps and services – Mock-up U61: Payment for apps and services – Fully deployed	Priority: Could Have Will Not Have For Now (U61)
Analysis and Comments: Related to the requirements U128 to U133, payment methods for the purchase of the apps and services should be developed and established. Consequently, a wide range of payment methods (PayPal, debit cards, credit cards) should be supported. Please note, however, that the prototype implementation of SIMPLI-CITY may not support those features or may only support mock-ups of the payment process due to the RTD nature of the project.		
Primary Task Assignment: • T5.4: Should be able to handle payment information and to allow/disallow the service usage based on the payment status Secondary Task Assignment: • None		

Name: Commercialization	U128-U132: Business models for apps U133-U137: Business models for services	Priority: Could Have Should Have (U128,129)
Analysis and Comments: A key issue of SIMPLI-CITY is to allow developers to create new apps and services. For achieving this, developers will need a motivation for creating apps/services in terms of allowing a commercialization of their work. The consortium is convinced that this may lead to a vital ecosystem as it is seen in modern app stores such as the Apple AppStore or the Google Play market. Therefore, SIMPLI-CITY should support various business models including paid apps/services as well as freemium approaches.		
Primary Task Assignment: • T5.4: Should be able to handle payment information and to allow/disallow the service usage based on the payment status • T6.3: Should prevent the download and installation of apps which are not available in the current license Secondary Task Assignment: • None		

Name: App Versioning	U62: Versioning of apps U63: Upgrading of apps U144: App versioning	Priority: Must Have
Analysis and Comments: Due to the lifecycle of an application, it is required to allow developers to submit new versions of an app to SIMPLI-CITY. Versions may be targeting sales/marketing information (e.g. Version 2013) or developers in the registry (e.g. Build 2034). Users should be able to see and upgrade their apps.		
Primary Task Assignment: • T5.4: Should be able to show sales version information to consumers Secondary Task Assignment: • T6.3: Should download and install updates to the device • T6.4: Should allow developers to specify both versions (sales and build)		

Name: Quality Assurance	U64: Quality assurance – Mock-up U65: Quality assurance – Fully deployed U142: Quality/checking certification	Priority: Must Have Could Have (U65)
Analysis and Comments: Developers should be able to submit their apps and services and to register them. Submitting may include a quality check as it is found in popular app markets such as the Apple App Store. Apps and services that do not pass this quality check should be rejected and a notification with an explanation should be sent to the developer. During the review process, apps and services should not be listed in the market.		
Primary Task Assignment: • T5.4: Should be able to provide a submission information or a review team and should allow the review team to approve or decline a submission		

Secondary Task Assignment: • None

Name: Marketplace Usage	U66: Marketplace easy to use U67: Apps and services are easy to buy U68: Discovery of apps and services	Priority: Must Have
Analysis and Comments: The marketplace will only be successful if it' is easy to use. This encompasses several aspects such as the discovery of apps, their installation, and the provision of an easy-to-use search functionality.		
Primary Task Assignment: • T5.4: Should be able to be used without technical expert knowledge Secondary Task Assignment: • None		

Name: App Management	U69:App download to the device U70: App installation U71: App uninstallation	Priority: Must Have
Analysis and Comments: Apps should be easily installable on and removable from the PMA. This includes the download of the app from the market as well as the local installation and technical preparation.		
Primary Task Assignment: • T5.4: Should be able to allow users the installation and uninstallation of apps including the download process Secondary Task Assignment: • T6.3: Should be able to handle the technical installation/setup based on T5.4		

Name: App and Service Statistics	U138: Provision of app statistics U139: Provision of service statistics U182: Provision of statistics, e.g. usage, traffic	Priority: Must Have
Analysis and Comments: Developers should be supported with statistics about their service and apps usage which covers, e.g., the number of installations as well as average values for the customer feedback (rating) and may also include statistics about the crashes or sales related data. Crash reports for services are part of T5.1 (see U180, U181, U183, U184).		
Primary Task Assignment: • T5.4: Should allow developers to view statistics in a console Secondary Task Assignment: • T6.4: Should be able to link apps to the market of T5.4 • T5.3: Should store information about app usage etc.		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Promotion	U140: Promotional aspects	Priority: Could Have
Analysis and Comments: SIMPLI-CITY should allow developers to promote their apps and services with special offers. Those promotions may fall into three categories: Firstly, cross-selling offers, which would allow users that buy product A to get a discount for product B if they purchase both together. Secondly, time restricted offers to, e.g., offer a cheaper purchase price for one week and thirdly, offers based on the purchase amount to offer 1000 versions of a product in the market with a discount.		
Primary Task Assignment: • T5.4: Should allow developers to define cross-selling offers and offers based on timely restrictions Secondary Task Assignment: • None		

Name: App Publication	U143: App publication U145: App removal U147: Easy to publish	Priority: Must Have
Analysis and Comments: Apps should be easily submittable. Developers should be able to manage their apps via a web interface. This includes the possibility to remove existing apps from the marketplace.		
Primary Task Assignment: • T5.4: Should be able to be used without technical expert knowledge even for developers Secondary Task Assignment: • T6.4: Should be able to prepare the submission of apps by developers		

Name: Call for Tenders	U146: Call for developers	Priority: Will Not Have For Now
Analysis and Comments: In many cases, users may have very promising ideas for new apps and services. As such, SIMPLI-CITY will provide them with a forum to describe their ideas and to search for developers. This process may include a pure description of ideas allowing developers to develop services and apps whenever they want. Additionally, users may offer funding for the development work in which case developers may express their interest in a specific development job.		
Primary Task Assignment: • T5.4: Should be able to allow users to enter new ideas for apps and services. Developers should be able to bid on those ideas for either developing them for free or for a small charge. Users may then select one or more developers to develop the app / service. Secondary Task Assignment: • None		

5.5 Related to WP6 Personal Mobility Assistant

5.5.1 Related to T6.1 Dialogue Interface Prototype and T6.2 Voice-based Multimodal User Interface

Name: Response Time	U18: Reasonable response time	Priority: Should Have
Analysis and Comments: Reasonable response time means that the system responds to user requests in reasonable time. For T6.2, this means that given that the response time should be kept at such a level that the whole system is perceived as responsive given that the applications and services also are responsive. The Apple Siri response times could serve as a benchmark in case of good car connectivity.		
Primary Task Assignment: • T6.1&T6.2: Must make sure that the cloud solutions in the UI layer are fast enough, and that, when services, applications and UI components are slow, feedback is given to the user that something still happens. Secondary Task Assignment: • None		

Name: PMA Configuration	U19: Minimum manual configuration U22: Personalization – Incremental configuration	Priority: Could Have
Analysis and Comments: The system should be able to use with a minimum of configuration, e.g., following an “One-click configuration” approach. One solution could be a configuration app or similar approach. The system should support an incremental way of configuration. The system should be usable without any configuration, but the user experience should be improved when the user enters more information.		
Primary Task Assignment: • T6.1: Must build a configuration app, and/or mechanism to extract configuration information from data submitted to applications Secondary Task Assignment: • T6.2: Make potential config application multimodal		

Name: Polyglot PMA	U20: Multilingual	Priority: Could Have
Analysis and Comments: The system should be easily extended with new languages. The default language will be (British) English.		
Primary Task Assignment: • T6.1 & T6.2: Make multilingual support in TDM easily accessible Secondary Task Assignment: • T6.4: The development environment must be capable of handling strings and grammar components for different languages		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Voice Commands	U21: Link voice commands to apps	Priority: Will Not Have For Now
Analysis and Comments: The user should have the option to link specific voice commands to certain applications. This is preferably done as a traditional configuration.		
Primary Task Assignment: • T6.2: Must create a GUI-based mode/an application/etc. to create such links. Secondary Task Assignment: • None		

Name: Speech Recognition	U36: Natural speech recognition	Priority: Must Have
Analysis and Comments: In SIMPLI-CITY, the user does not need to learn a limited number of predefined commands in order to use the PMA, but can use a relatively large set of expressions to communicate to the system.		
Primary Task Assignment: • T6.1: Support for robust parsing of input from dictation ASRs is needed Secondary Task Assignment: • None		

Name: Interaction with the PMA	U37: Result oriented instead of service/app recognition oriented U38: Disambiguation U39: Provision of a limited number of alternatives	Priority: Could Have
Analysis and Comments: In the case of ambiguity (when an utterance is associated with more than one meaningful interpretation by the system), or when a number of apps can handle the same type of tasks, there should be a disambiguation process built into the system. This process should lead to a system state where only one interpretation of the utterance remains. The user should not need to be aware of exactly which apps can help him or her to fulfil the request. The disambiguation should work according to the following prioritization: 1. The system should use relevant context information (when available) to choose the appropriate app/interpretation without asking the user. 2. When the system cannot decide without the help of the user, the user should be prompted to make the decision. 3. The user should be prompted to select from a very short list of relevant apps/interpretations.		
Primary Task Assignment: • T6.1: Must provide according mechanisms to resolve ambiguity Secondary Task Assignment • T6.2: Must create filtering technique for list items 1 and 3.		

Name: PMA Voice	U40: Friendly voice U44: Voice quality	Priority: Could Have
Analysis and Comments: The PMA should have a high quality voice, which is also friendly-sounding. SIMPLI-CITY will use the voices that are available as a default on the respective platforms. If the default voices are deemed to be of too low quality or too unfriendly, some alternative solution needs to be found.		
Primary Task Assignment: • T6.1: Must evaluate default voices from existing software frameworks Secondary Task Assignment • None		

Name: Input interactions via multimodal UI	U41: Input interactions with system via multimodal UI: on screen, voice control, and non-voice control	Priority: Must Have
Analysis and Comments: Input must be collected from the user. The output of the ASR component must be parsed into dialogue moves that can be used as input to the DME (Dialogue Move Engine) of the PMA. The interpretation of a particular utterance is dependent on the current dialogue context. The GUI of the system, displayed on the screen, must be built in such a way that it is easy to interpret the user intention behind tapping on and in other ways manipulating different parts of it.		
Primary Task Assignment: • T6.1: ASR input and interpretation in the light of the current context. Secondary Task Assignment: • T6.2: GUI design and interpretation.		

Name: Output interaction through UI	U42: Output interaction from system through UI	Priority: Must Have
Analysis and Comments: Output from the system needs to be displayed on the screen as well as spoken to the user in a coordinated manner. The Text To Speech (TTS) unit of the system and the GUI will need to communicate in order to achieve this. The DME will select dialogue moves to be realised. The dialogue moves will be realised in different ways by generation components, and the realisation will need to be coordinated by some coordination component. If the generation components and the GUI/TTS modules are run on different machines, there is a need for a communication protocol for communicating the realisation information and the coordination information between the realisation components.		
Primary Task Assignment: • T6.1: Generation and realisation of spoken utterances. Secondary Task Assignment: • T6.2: Generation and realisation of GUI output, coordination of GUI and TTS output.		

Name: Distraction by the PMA	U43: Non-distracting interaction	Priority: Should Have
Analysis and Comments: The interaction with the PMA should be non-distracting, meaning that the user's workload should be kept below a certain level, that a user should not be disturbed when in stress etc. This requirement is partially related to requirement U196.		
Primary Task Assignment: • T6.1 & T6.2: Should implement findings from research on non-distracting user interaction both regarding voice interaction and multi-modal interaction Secondary Task Assignment • None		

Name: Voice recognition	U45: Automotive quality voice recognition (automotive acoustic models for in car use)	Priority: Should Have
Analysis and Comments: The speech recognition system to be used in car, also if used through a mobile phone or other portable device, should be robust to the noise typical of the automotive environment. It should thus be optimized for the automotive acoustic environment; this characteristic is given by the acoustic model with which the engine is trained. Normally a voice recognition engine for mobile phones or other device has not this feature and thus doesn't have good recognition performances (in term of accuracy and recognition rate) when used in a driving car.		
Primary Task Assignment: • T6.1 The voice recognition should be suitable for the automotive environment		

Name: High speech recognition rate	U46: High speech recognition rate	Priority: Should Have
Analysis and Comments: The SIMPLI-CITY system should have high accuracy and speed recognizing voices. Different target value are given for different conditions of use (noise scenarios), complexity of the task, kind of commands, complexity of the grammars, etc.		
Primary Task Assignment: • T6.1 The system should have high performance in terms of word error rate (WER).		

Name: Voice interaction through the car audio system	U47: Voice interaction through the car audio system (microphone & loudspeakers) for hands free in car use	Priority: Could Have
Analysis and Comments: The SIMPLI-CITY system should be able to recognize voice thought the in car automotive microphone and play audio from the car loudspeakers.		
Primary Task Assignment: • T6.2 The voice-based user interface should use the in-car audio components (microphone and speakers) for the voice interaction.		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

5.5.2 Related to T6.3 Mobile Application Runtime Environment

Name: App Crashing	U48: App crashes are minimized U49: A crash should not impact other apps U103: Fault tolerance U104: Stability	Priority: Must Have Should Have (U48, U49)
Analysis and Comments: For a good user experience, the SIMPLI-CITY PMA and its apps should be stable and should not crash. However, since SIMPLI-CITY will allow authors to write third party apps, the stability of an app cannot be directly influenced by the SIMPLI-CITY team. Therefore, SIMPLI-CITY should ensure that apps are isolated up to a certain extend and that a crash of an app does not crash the full SIMPLI-CITY system.		
Primary Task Assignment: • T6.3: The application runtime environment should ensure that app crashes do not have a major impact into the overall system Secondary Task Assignment(s): • None		

Name: Device	U72: Android support U73: iOS support U74: Blackberry support U75: Windows phone support U76: Tablet support	Priority: Must Have Could Have (U76) Will Not Have For Now (U73, U76 U75)
Analysis and Comments: In an ideal situation, SIMPLI-CITY should support multiple device platforms (by allowing the PMA to run on it). This will allow SIMPLI-CITY to be usable within various environments. However, since the project time is limited, the project team will most likely only implement an Android version.		
Primary Task Assignment: • T6.3: The application runtime environment should be implemented in different OS environments Secondary Task Assignment(s): • T6.1, T6.2, T6.4: Should support the same OS environments as T6.3		

Name: Device Exchange	U79: Support to exchange devices	Priority: Will Not Have For Now
Analysis and Comments: SIMPLI-CITY should allow users to use their account without any reconfiguration work on multiple devices. As such, the user should be able to switch from one device to another and to take his/her settings and apps with him/her.		
Primary Task Assignment: • T6.3: The application runtime environment should use the cloud storage to store all user and app settings and should automatically restore all data on new devices		

Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment(s): • None

Name: App Compatibility	U100: Backwards compatibility of apps	Priority: Should Have
Analysis and Comments: Apps developed under an old version of SIMPLI-CITY should run on PMA even if the PMA is based on a newer version of SIMPLI-CITY		
Primary Task Assignment: • T6.3: The application runtime environment should be able to launch/integrate apps that are based on an older SIMPLI-CITY version Secondary Task Assignment(s): • T6.4: Should allow developers to state the minimum and maximum version of SIMPLI-CITY for each app		

Name: Inter-apps Communication	U185: Standardized messages between apps	Priority: Must Have
Analysis and Comments: SIMPLI-CITY should allow apps to exchange messages with each other. For this, a standardized message exchange interface is needed.		
Primary Task Assignment: • T6.3: The application runtime environment should be able to receive messages from an app and to forward it to another app. Furthermore, an exchange format for messages needs to be defined. Secondary Task Assignment(s): • None		

Name: Local App Registry	U186: Registry of installed SIMPLI-CITY apps	Priority: Must Have
Analysis and Comments: SIMPLI-CITY should allow apps to query the list of installed apps and their versions.		
Primary Task Assignment: • T6.3: The application runtime environment should be able to deliver a list of installed apps to other apps in the system Secondary Task Assignment(s): • None		

Name: Information Exchange Between Server and Apps	U193: Exchange of information from apps to server U194: Exchange of information from server to apps	Priority: Must Have
Analysis and Comments: SIMPLI-CITY will have to allow the communication between the device and the server side. This concerns both ways: From apps to services and from services to apps. Apps need to be able to invoke services to query information on a one-time or on a regular base (pull). Optionally, services may actively inform apps about events (push). SIMPLI-CITY will support both ways. The communication from apps to the server will be handled by a pull mechanism, which will allow apps to invoke services by making calls to the SIMPLI-CITY server side. Those calls will technically be handled by the application runtime environment and routed via the service runtime environment. Calls may return values such as a response string. The communication from server to the apps will be performed with push notifications. Services may use this way of communication for informing users of an app about events, e.g., a traffic jam. Technically, the application runtime environment will provide a special push notification service which services may use to send those push notifications via a simple call.		
Primary Task Assignment: • T6.3: The application runtime environment should provide an API method for apps to invoke services from the Service Runtime Environment. Furthermore, it should provide API methods to receive messages from the server. Secondary Task Assignment(s): • T5.3: The Service Environment should provide interfaces to be used by T6.3 for the communication with the services • T6.3: Should provide API methods to send data from the server to the apps		

5.5.3 Related to T6.4 Application Design Studio

Name: Look & Feel of Apps	U23: Unified look & feel within the project U24: Unified look & feel for 3rd party developers U25: Usage of UI guidelines within the project U26: Intuitive usability	Priority: Should Have
Analysis and Comments: The Application Design Studio should provide guidelines and templates allowing developers to create apps with a similar look & feel for creating a good user experience. It should also provide a document for defining usability guidelines for developers		
Primary Task Assignment: • T6.4: Should provide documents and templates for usability and a unified look&feel of apps Secondary Task Assignment: • None		

Name: API Compatibility	U102: Backwards compatibility of API	Priority: Should Have
Analysis and Comments: Apps developed under an old version of SIMPLI-CITY should be compatible with new versions of the SIMPLI-CITY components.		
Primary Task Assignment: • T6.4: The application studio should support different API versions Secondary Task Assignment(s): • All tasks of WP4, 5 and 6: Each component should provide an API that is backwards compatible		

Name: API Provision	U105. Access to the dialog system U106. Access to cloud services U159. Support for dialogue U160. Support for apps U161. Support for data services U162. Support for backend services	Priority: Must Have
Analysis and Comments: The Application Design Studio does not provide any interfaces to all the components (e.g., cloud storage, services, sensors) as the components will offer them in their interfaces. However, being a unique starting point for app developers, the Studio will ship those interfaces as part of the studio installation. This means that for all important components, app developers will either be able to directly make use of the component interface in their developments or they will have access to an interface document describing how they can access this component.		
Primary Task Assignment: • T6.4: Should ship the APIs of all components needed for app developers Secondary Task Assignment: • None		

Name: App Developer Studio APIs	U155: Provision of Java API U156: Provision of C/C++ API U157: Standard programming interface U158: Easy access to API through the developer studio	Priority: Must Have Will Not Have For Now (U156)
Analysis and Comments: SIMPLI-CITY should allow developers to create their apps in different development languages including Java and C/C++. It should equip developers with an easy to use API to develop new apps.		
Primary Task Assignment: • T6.4: The application design studio should provide easy access to all APIs of WP4-6 in an intuitive environment.		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment(s): • None

Name: General App Developer Studio Requirements	U174:Permit to develop an app in less than a day U175: Hide complexity from the developer	Priority: Should Have
Analysis and Comments: SIMPLI-CITY should provide developers with a user-friendly environment to create new apps and to prepare the submission of apps. This App Developer Studio should allow developers to create manifests for their developments with a simple UI editor that hides the complexity of defining rights and controls as much as possible. Skilled developers should be able to use the studio to create apps within less than a day.		
Primary Task Assignment: • T6.3: The application design studio should provide an intuitive environment for developing apps, e.g. based on existing environments such as Eclipse Secondary Task Assignment(s): • None		

Name App Developer Signature	U166: Identification of the developer/signature	Priority: Must Have
Analysis and Comments: The App Development API should store the identification information of the app developer that created an app and provide this information to the end users.		
Primary Task Assignment: • T6.4: Must identify the developer of a service Secondary Task Assignment: • None		

Name: App Development Studio Documentation	U168: Provision of source code examples U169: Provision of UI templates U170: Provision of best practices U171: Provision of tutorials U172: Provision of guidelines U173: Provision of examples	Priority: Must Have
Analysis and Comments: The App Development Studio should support app developers during the app creation process through according documentation and examples. This support should include the provision of source code examples, user interface templates, general guidelines, tutorials and examples of the overall process of creation of app.		
Primary Task Assignment: • T6.4: Must provide resources to support developers during the creation of apps		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Secondary Task Assignment: • None

Name: PMA Emulator	U177: Provision of a PMA emulator	Priority: Will Not Have For Now
Analysis and Comments: SIMPLI-CITY should allow developers to test their apps within the studio without the need to test them on a real device. For this, a PMA emulator is needed.		
Primary Task Assignment: • T6.3: The Application Design Studio should provide a PMA emulator Secondary Task Assignment(s): • None		

Name: Functional Description of Apps	U178: Definition of minimum hardware requirements of apps	Priority: Must Have
Analysis and Comments: SIMPLI-CITY should allow developers to describe their app developments in a manifest stating various properties such as the system requirements, categories, etc.		
Primary Task Assignment: • T6.3: The application design studio should provide a manifest editor Secondary Task Assignment(s): • T5.3: Should support the manifest for services within the registry		

Name: App Debugging	U180: Easy debugging of services U181: App crash reports U182: Provision of statistics, e.g. usage, traffic U183: Provision of bug reports U184: Provision of crash reports	Priority: Must Have Could Have (U180, U181, U183, U184)
Analysis and Comments: The Application Design Studio should provide debugging functionalities that allow developers to debug apps and to be informed about the correct operation or about any problem occurred during the process. This includes the provision of bug reports and crash statistics/reports to the app developer.		
Primary Task Assignment: • T6.4: Should provide app debugging information in a similar vein as the Google Developer Console for the App Market, which allows a unified access to statistics and bug reports on a per-app base Secondary Task Assignment: • T6.3: Should collect bug and statistic information and deliver it to the cloud • T5.3: Should collect bug and statistic information and deliver it to T6.3 via a method		

http://www.simpli-city.eu/ Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: App Composition	U187: Composition of apps	Priority: Will Not Have For Now
Analysis and Comments: SIMPLI-CITY should provide a mashup editor to create mashups of apps and to allow a graphical way of combining apps.		
Primary Task Assignment: • T6.4: Should provide an app mashup editor Secondary Task Assignment(s): • T6.3: Needs to support the execution of app mashups and the exchange of data between apps		

6 Use Case Requirements

This section provides the description of the use case requirements as listed in section 3. These requirements are grouped depending on the use case scenario of the project that they belong to. The description of the different scenarios (namely the IBM scenario, the SRM scenario, the Tempos 21 scenario, and the CRF scenario) can be found in the SIMPLI-CITY project vision deliverable (D2.1). Moreover, there are some generic requirements, which apply to different use cases.

6.1.1 Generic

Name: Learning from feedback	U28: The system learns from feedback	Priority: Will Not Have For Now
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to adapt their output depending on past interactions between the user and the apps. The logic of this learning process should be implemented in the apps.		
Primary Task Assignment: T7.1, T8.1: The definition of the use cases should be able to adapt the output of an app depending on past interactions Secondary Task Assignment(s): • None		

Name: User recognition via voice dialog	U29: Reaction to who is in the car, via simple dialog	Priority: Could Have
Analysis and Comments: The Personal Mobility Assistant should be able to recognize who is in the car via a voice-based user interface. Once recognized the vehicle can setup the user preference of the current driver.		
Primary Task Assignment: • T7.1, T8.1: The definition of the use cases should take into account the reaction to who is in the car via a simple dialog Secondary Task Assignment(s): • T6.1 The user interface should recognize the driver via simple dialog		

Name: User recognition via sensors	U30: Reaction to who is in the car, via sensors	Priority: Should Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to react depending on who is in the car. The user should be able to be recognized by means of the sensors of the vehicle, e.g. it is possible to recognize who is driving depending on the key that opened the car.		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Primary Task Assignment:

- T7.1, T8.1: The definition of the use cases should take into account the reaction to who is in the car via sensors

Secondary Task Assignment(s):

- T4.3 The vehicle sensors should be able to recognize the users

Name: Reaction to time	U31: Reaction to time of the day	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to suggest alternative solutions e.g. change routes based on the time arrival and traffic information. The driver should be also informed for example if it is better to change the order of the scheduled appointment according to the information coming from the cloud.		

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account the reaction to time

Secondary Task Assignment(s):

- T4.1: Must support basic integration of data
- T4.3: Must allow access to external data, e.g. time and traffic
- T4.4: Must provide the basics for contextualizing information using end-user information

Name: Reaction to sensors	U32: Reaction to sensors	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to react to the information provided by the sensors. These sensors may include car sensors and external sensors. The SIMPLI-CITY framework should be able acquire, manage and store information coming from the sensor in a very fast way e.g. the temperature value for a specific sensor.		

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account the reaction to sensors

Secondary Task Assignment(s):

- T4.3: Must allow access to data from sensors
- T4.2: Must store historical information

Name: Reaction to car KPIs	U33: Reaction to KPIs from the car, like level of fuel and kilometres done	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to suggest to the driver to perform some action depending on the information coming from the vehicle KPIs e.g. the driver should be informed to stop to the upcoming gas station because with the current fuel autonomy he/she cannot reach the next petrol station.		

Primary Task Assignment:

- T8.1: The definition of the use cases should take into account the diagnosis of traffic condition in real-time

Secondary Task Assignment(s):

- T4.3: Must allow access to KPIs from the car

Name: Reaction to history of usage	U34: Reaction to history of usage	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to suggest to the driver how to improve his/her eco-driving performance on the basis of the history of the usage of the driver.		

Primary Task Assignment:

- T8.1: The definition of the use cases should take into account the diagnosis of traffic condition in real-time

Secondary Task Assignment(s):

- T4.2: Must store historical information
- T4.4: Must provide the basics for contextualizing information using end-user information

Name: Reaction to traffic information	U35: Reaction to traffic information, like traffic jams, train schedules, road works, accidents, and strikes	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to suggest to the driver how to reach his/her destination basing on the information coming from the cloud. The driver will be informed to change, for example, the means of transport, because he/she may not able to reach the destination by car in time due to an accident, but will arrive in time if he/she takes the next train.		

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account the diagnosis of traffic condition in real-time

Secondary Task Assignment(s):

- T4.2: The cloud-based infrastructure should provide information about traffic jams, train schedule, roadworks, accidents and strikes
- T4.4: Data about traffic jams, train schedules, roadworks, accident statistics, or strikes, could also be directly obtained from Open Government Data sources on the Web.

Name: Provision of personalized info	U201: Provision of personalized info, e.g. travel, costs	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should be able to provide personalized info to the users. This means that the output of the apps should vary depending on the user and his/her context.		

Primary Task Assignment:

- T7.1, T8.1: The definition of the use cases should take into account the provision of personalized info

Secondary Task Assignment:

- T5.2: Must recognize which information is relevant to the user to a particular degree

6.1.2 IBM Scenario

Name: Traffic condition diagnosis	U202: Diagnosis of abnormal traffic condition in real-time	Priority: Must Have
Analysis and Comments: The SIMPLI-CITY users should have the possibility to have real-time explanation of abnormal traffic conditions. In other words, reasoning mechanisms should be supported by SIMPLI-CITY for identifying the nature and causes of abnormal traffic conditions such as traffic congestion.		
Primary Task Assignment: • T7.1: The definition of the use cases should take into account the diagnosis of traffic condition in real-time Secondary Task Assignment(s): • T4.1: Must provide a unified data model for combining and correlating heterogeneous data sources • T4.2: Must provide mechanisms for storing relevant data • T4.3: Must provide unified access to traffic-related sensors • T4.4: Must provide open data contextualisation of sensor data for accurate diagnosis		

Name: Traffic condition prediction	U203: Prediction of abnormal traffic condition	Priority: Must Have
Analysis and Comments: Good road navigation systems should be able to anticipate critical situations in road traffic, e.g., congestions, major delays, or strong perturbations, so relevant and accurate solutions are available in real-time. In such a way, car drivers would have the possibility to anticipate critical road traffic situations and reach any part of any large city without unexpected conditions. SIMPLI-CITY will provide such a navigation system where road traffic conditions will be predicted so road users are satisfied with the road infrastructure of the city.		
Primary Task Assignment: • T7.1: The definition of the use cases should take into account the prediction of traffic condition Secondary Task Assignment(s): • T4.1: Must provide a unified data model for combining and correlating heterogeneous data sources • T4.2: Must provide mechanisms for storing relevant data • T4.3: Must provide unified access to traffic-related sensors • T4.4: Must provide open data contextualisation of sensor data for accurate prediction		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Querying diagnosis historic	U204: Support for querying diagnosis historic	Priority: Must Have
Analysis and Comments: In the context of the search of historical road traffic diagnosis the SIMPLI-CITY system should offer functionalities for querying historic diagnosis results through the following parameters: (1) the type of traffic anomaly is interested in e.g., delayed buses, congested buses, congested roads, (2) a boundary box or specific location where the system should track for anomalies and diagnosis results, (3) a time window, (4) some keywords for refining the search. The SIMPLI-CITY system should display information and statistics related to the query results using some spatial and temporal visualization features.		
Primary Task Assignment: • T7.1: The definition of the use cases should take into account the query of traffic anomalies diagnosis Secondary Task Assignment(s): • T4.1: Must provide a unified data model for combining and correlating heterogeneous data sources • T4.2: Must provide mechanisms for storing relevant data • T4.3: Must provide unified access to traffic-related sensors • T4.4: Must provide open data contextualisation of sensor data for structured search		

Name: Querying impact factor	U205: Support for querying impact factor on traffic condition	Priority: Must Have
Analysis and Comments: Traffic conditions are impacted by various exogenous events. SIMPLI-CITY should provide support for end-users to obtain information about their impact factor and their severity (e.g. to show the estimated amount of additional time caused by a traffic jam delay).		
Primary Task Assignment: • T7.1: The definition of the use cases should take into account the query of impact factor on traffic condition Secondary Task Assignment(s): • T4.1: Must provide a unified data model for combining and correlating heterogeneous data sources • T4.2: Must provide mechanisms for storing relevant data • T4.3: Must provide unified access to traffic-related sensors • T4.4: Must provide open data contextualisation of sensor data for obtaining insight on impact factor on traffic condition		

6.1.3 SRM Scenario

Name: Optimization of financial resources	U206: Optimization of financial resources	Priority: Must Have
Analysis and Comments: The Personal Mobility Assistant should be able to optimize the route in order to reduce as much as possible costs for the users, taking into account parking fees per hours, tolls to access specific areas of the city, traffic congestion affecting travelling time and consequently fuel consumption, etc.		

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account optimization of financial resources

Secondary Task Assignment(s):

- T4.4: Information about costs-generators (parks, bus tickets, tolls,…) has to be known

Name: Road / trip suggestions	U207: Support suggestions for road/trip optimization if conditions change	Priority: Must Have

Analysis and Comments: The Personal Mobility Assistant should be able to calculate the optimized route depending on real-time conditions, in particular on one side it should take into account all access restrictions in force in the city (as they change depending on the day and on the time) and calculate route in order to avoid forbidden areas, on the other side it should be able to recalculate the route when conditions change (traffic congestion, no more parking availability, etc.) and communicate to the user the new route.

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account to support suggestions for road/trip optimization if conditions change

Secondary Task Assignment(s):

- T4.4: Real-time information are the trigger of rerouting process
- T5.3: Must provide alert services functionalities (see requirements U27, U195)
- T6.1: PMA should give information to the users about the new route to follow

Name: Trip planning continuation	U208: Possibility to continue a previously started trip planning on the same device or different device	Priority: Must Have

Analysis and Comments: The Personal Mobility Assistant could be used by users on different devices, such as smartphones, tablet or on-board car-computers, so it could be possible to switch from one device to another if conditions or mean of transportation changes, i.e. from car-computer to smartphone when a user leaves the car in a car park and takes the bus to complete his/her route. This Requirement is related to U79 Support to exchange devices

Primary Task Assignment:

- T7.1: The definition of the use cases should take into account the possibility to continue a previously started trip planning on the same device or different device

Secondary Task Assignment(s):

- T6.3: The application runtime environment should use the cloud storage to store all user and app settings and should automatically restore all data on new devices

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

6.1.4 Tempos 21 Scenario

Name: Current route information	U209: Provision of real-time information about the current route	Priority: Must Have
Analysis and Comments: Applications related with navigation should provide real-time information about the current route of the user. This information may include Points of Interest, parking slots, traffic information, alternative routes, or alternative public transport, depending on the needs of the application.		
Primary Task Assignment: • T7.1, T8.1: The definition of the use cases should take into account the provision of real-time information of the current route Secondary Task Assignment(s): • T5.2: Must provide information to users according to their context • T5.3: Must be able to push updates to the PMA • T6.3: Must be able to notify the application about real-time information of the route		

Name: Multimedia reproduction	U210: Reproduction of multimedia information	Priority: Must Have
Analysis and Comments: Applications developed for the SIMPLI-CITY platform should be able to reproduce multimedia, including audio and video. They should support the standard multimedia formats.		
Primary Task Assignment: • T8.1: The definition of the use cases should take into account the reproduction of multimedia information when needed Secondary Task Assignment(s): • T4.5: Must be able to provide media streams		

Name: Streaming audio reproduction	U211: Reproduction of streaming audio	Priority: Should Have
Analysis and Comments: Applications developed for the SIMPLI-CITY platform should be able to reproduce streaming audio from different data sources. They should support standard streaming audio formats.		
Primary Task Assignment: • T8.1: The definition of the use cases should take into account the reproduction of streaming audio when needed Secondary Task Assignment(s): • T4.5: Must be able to provide media streams		

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Notification of Points of Interest	U212: Notification to end user about the proximity of Points of Interest	Priority: Must Have
Analysis and Comments: Applications developed for the SIMPLI-CITY platform should be able to notify end users about the proximity of Points of Interest. This notification should be proactive by the platform, meaning that the server should be able to detect the proximity of the user to the Point of Interest, and then notify the application of this proximity so that the application can inform the user. This requirement is a sub-requirement of U27.		
Task Assignment: • T8.1: The definition of the use cases should take into account proactive notifications to end users Secondary Task Assignment(s): • T5.2: Must detect the proximity of the user to a Point of Interest • T5.3: Needs to enable services to forward information to the end user app • T6.3: Must also be able to display notifications when they are received (via T6.2).		

Name: Social network integration	U213: Social network integration	Priority: Must Have
Analysis and Comments: Applications developed for the SIMPLI-CITY platform should be able integrate with social networks such as Facebook or twitter, in order that users can publish information in these social networks and share information with their contacts through the SIMPLI-CITY applications.		
Task Assignment: • T8.1: The definition of the use cases should take into account the integration with social networks Secondary Task Assignment(s): • None		

6.1.5 CRF Scenario

Name: Eco-driving information reporting	U214: Reporting to the end user about eco-driving information	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should provide information to the end user about the user's eco-driving style in an intuitive and aggregate form. Information about the performance of the vehicle in terms of acceleration, speed, brake, gear shift will be provided to the end user.		
Task Assignment: • T8.1: The definition of the use cases should take into account the reporting to the end user about eco-driving information Secondary Task Assignment(s): • None		

http://www.simpli-city.eu/ | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Name: Vehicle information	U215: Vehicle information available to the system	Priority: Must Have
Analysis and Comments: The vehicle should provide to the SIMPLI-CITY system information about the vehicle's parameters. Information about the total odometer, instantaneous vehicle speed, GPS longitude and latitude should be available to the SIMPLI-CITY system.		
Task Assignment: • T8.1: The definition of the use cases should take into account to provide the vehicle information to the system Secondary Task Assignment(s): • T4.3 The vehicle sensor data should be available for the mobility services and user application		

Name: Real time feedback	U216: Provision of real time feedback to the user in order to improve his performance	Priority: Must Have
Analysis and Comments: In order to improve the user experience, the apps developed for SIMPLI-CITY should provide responsive feedbacks. The driver should be able to take fast decision if needed to change route or have real time feedback about the current eco-driving style.		
Task Assignment: • T8.1: The definition of the use cases should take into account the provision of real time feedback to the user Secondary Task Assignment(s): • None		

Name: Eco-driving journey	U217: Access to a journey-related eco-driving data using the specific web portal U111: Provision of web site of the car	Priority: Must Have Could Have (U111)
Analysis and Comments: The SIMPLI-CITY system should provide to the end user information about the journey- related eco-driving information using a specific web portal. This web portal will be published on the Internet and publicly accessible by the driver and the other passengers of the vehicle. The driver can receive suggestions about his performance on specific route that should be displayed in a map on the specific web page.		
Task Assignment: T8.1: The definition of the use cases should take into account the provision of a journey-related eco-driving data using the specific web portal Secondary Task Assignment(s): • T8.2: It should provide a web portal access where journey-related eco-driving data is available		

Name: Eco-performance comparison	U218: Comparing (eco-)performances of different drivers	Priority: Must Have
Analysis and Comments: The apps developed for SIMPLI-CITY should compare the individual (eco-) performance against the performances of other drivers with comparable parameters, e.g., similar traffic situation, similar cars, and of course similar journeys.		
Task Assignment: • T8.1: The definition of the use cases should take into account the comparison of performance of different drivers Secondary Task Assignment(s): • T8.2: It should provide a service that analyses the eco-driving data of each SIMPLI-CITY driver.		

Name: Eco-driving contest	U219: Reward drivers through the eco-driving contest	Priority: Should Have
Analysis and Comments: The apps developed for SIMPLI-CITY should compare drivers using similar vehicles on similar routes and should award the winners on a regular basis. The best performing drivers should earn rewards from the SIMPLI-CITY system, e.g.. by Facebook “liking” or giving a Google “+1” or receive some award points that the driver can convert to discount on the next parking ticket or a free ticket for the bus.		
Task Assignment: • T8.1: The definition of the use cases should take into account to reward drivers through the eco-driving contest Secondary Task Assignment(s): • T8.2: It should provide a way to reward the best eco-driver in the SIMPLI-CITY “Community” web portal		

7 Conclusions

Requirements definition and analysis means an important step forward in the SIMPLI-CITY project definition process as it provides the functionality and properties which have to be taken into account in the functional and technical specification process which will be addressed in the following steps of the project.

The first step of the requirements engineering process included the definition of the requirements expected by the users of the SIMPLI-CITY platform, namely end users and developers. From these list of user requirements, the strategic, functional and use case requirements were extracted and analysed.

Strategic requirements refer to the high-level goals and expectations of the project. Functional requirements capture the intended behaviour of the SIMPLI-CITY platform for supporting the goals and expectations of the project, and specify the functionality that should be provided by each component of the platform. Finally, use case requirements define the functionality to be provided by the use case scenarios that will be developed within the project.

This deliverable is the basis for the upcoming RTD work and relating deliverables of Work Package 3, like the architecture definition and the functional and technical specifications (deliverables D3.1, D3.2.1 and D3.2.2).

http://www.simpli-city.eu/	Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201

Annex 1: User Requirements Mind Map

This Annex includes the mind map used in order to collect the list of user requirements of the SIMPLI-CITY project, as described in Section 2.2.

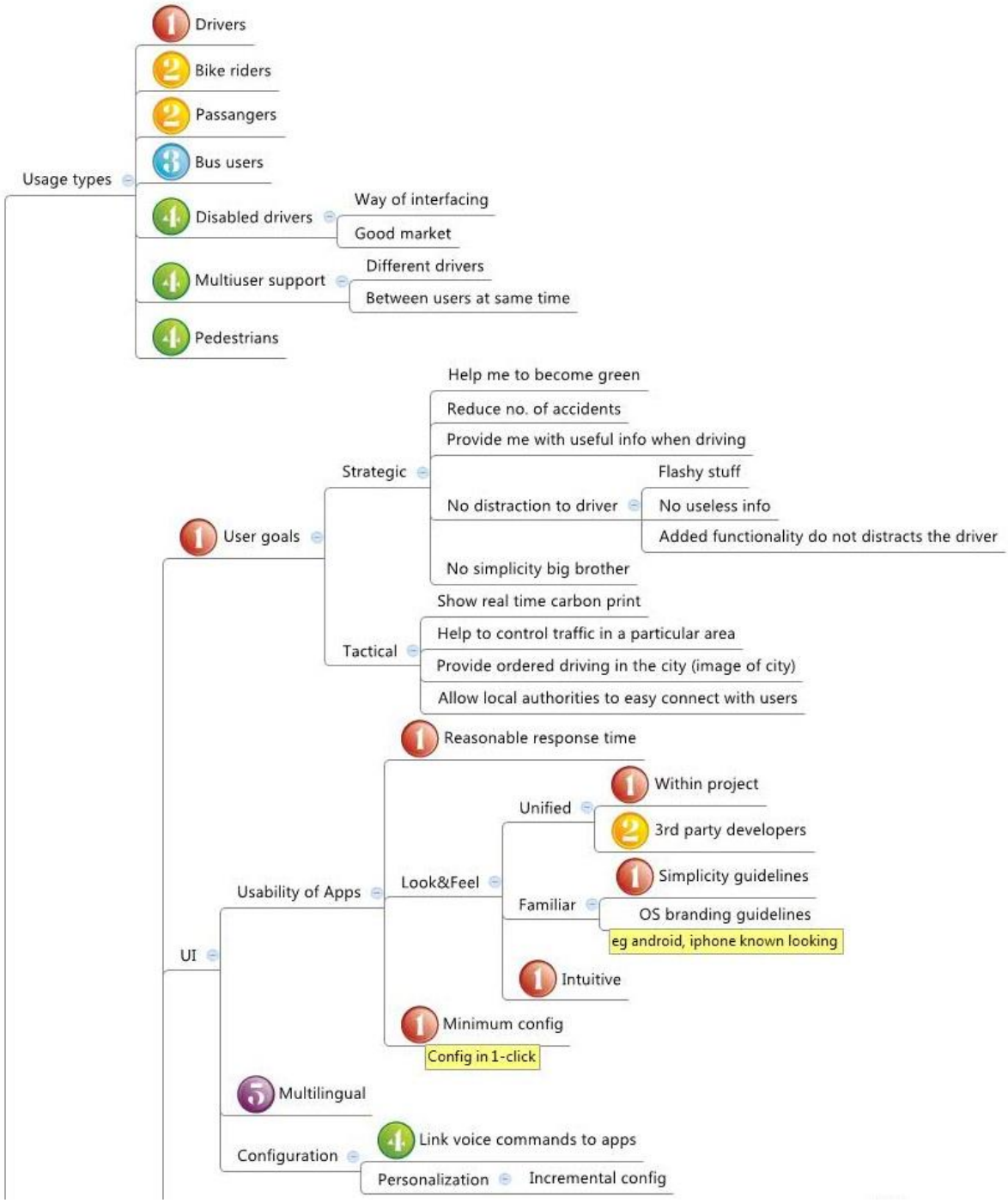

Figure 2: Requirements Mind Map – End Users 1

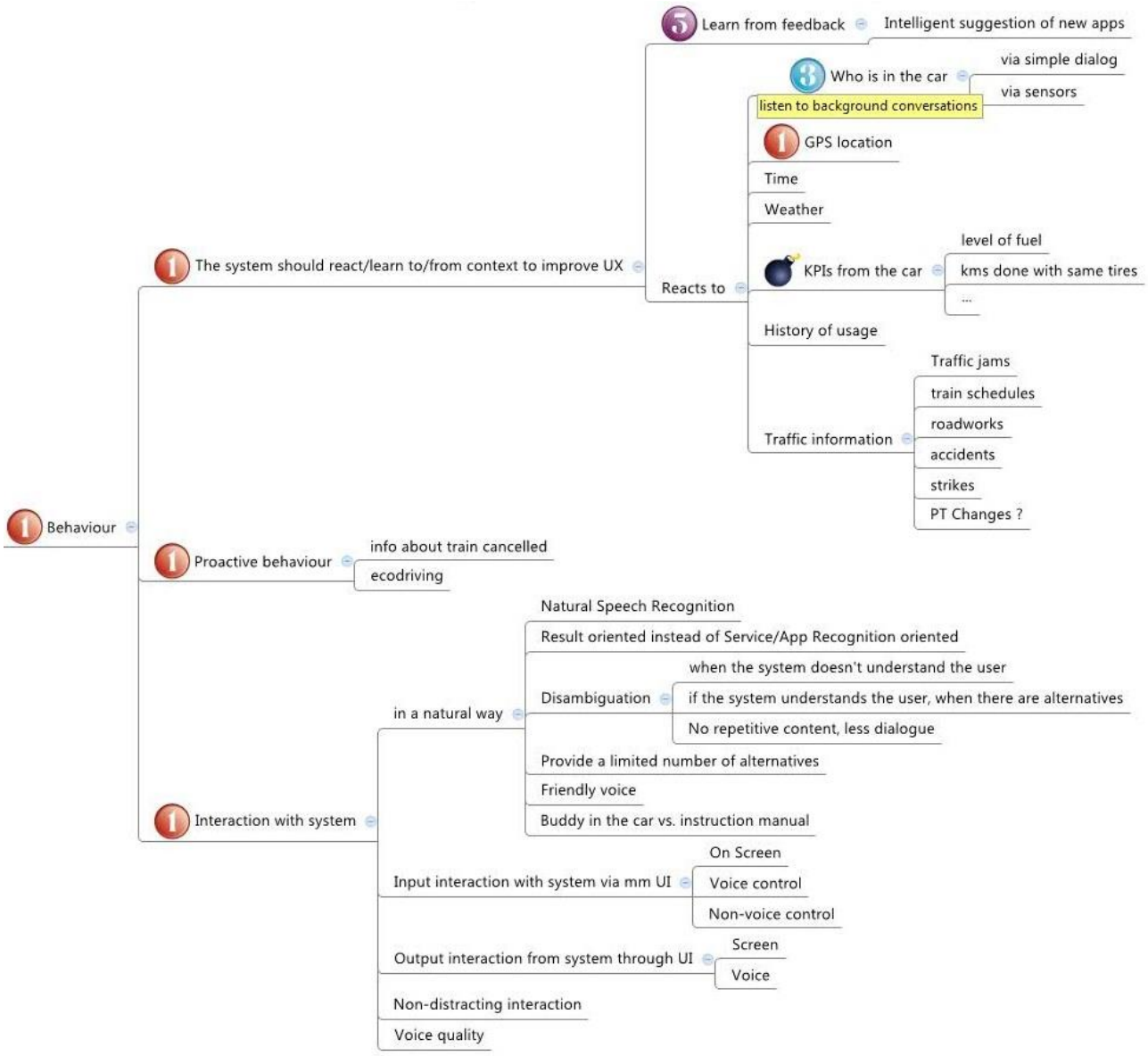

Figure 3: Requirements Mind Map – End Users 2

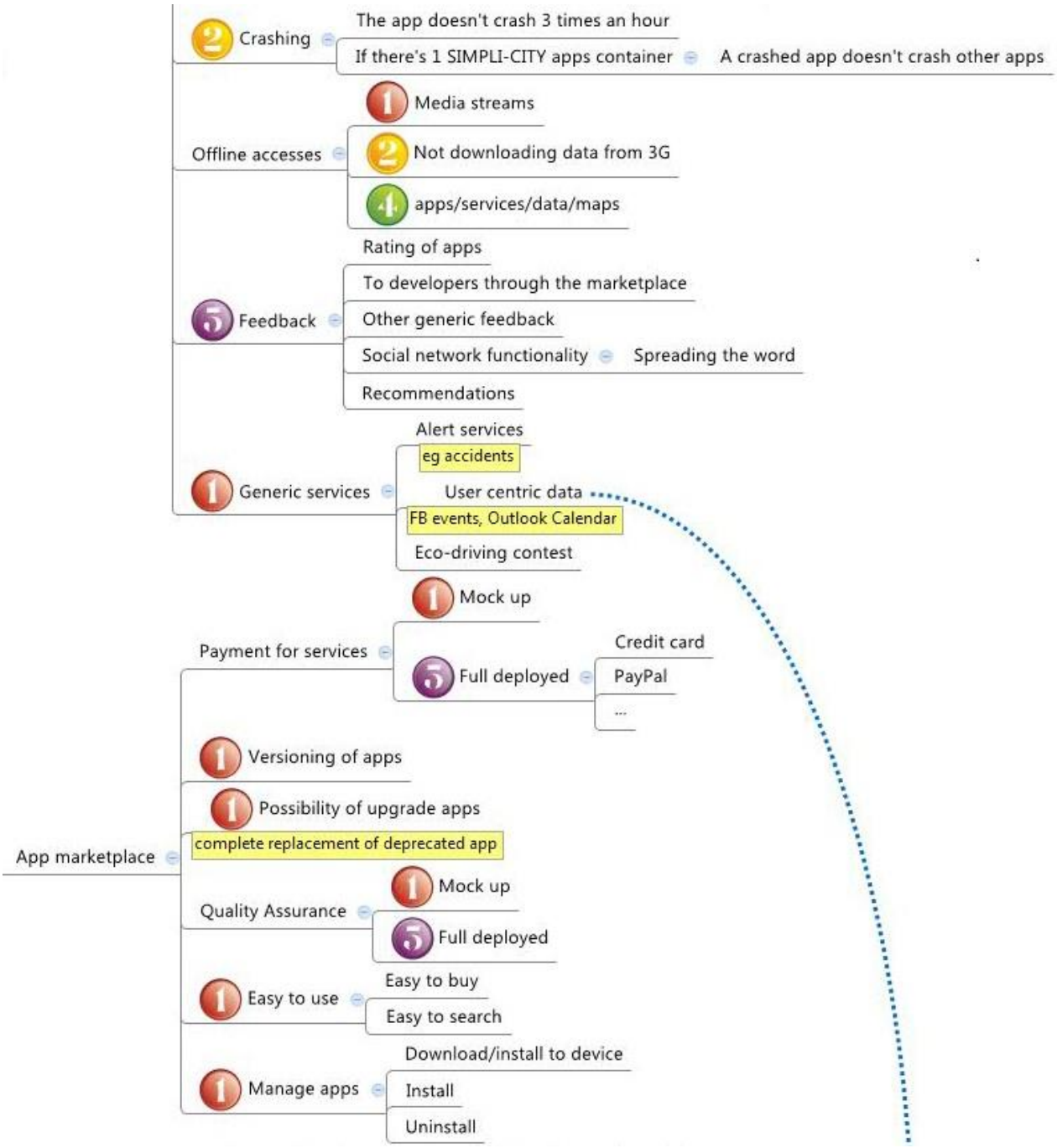

Figure 4: Requirements Mind Map – End Users 3

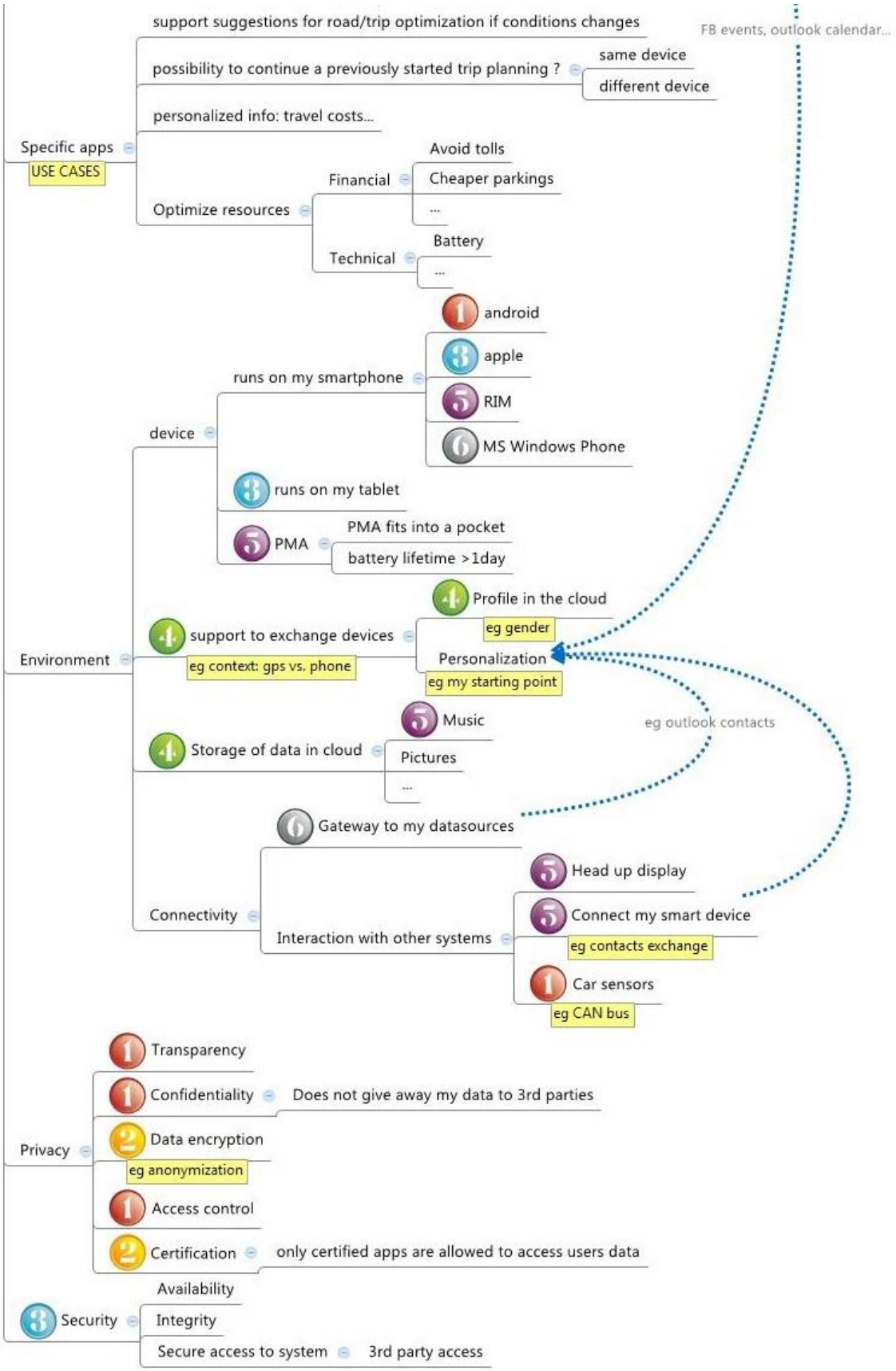

Figure 5: Requirements Mind Map – End Users 4

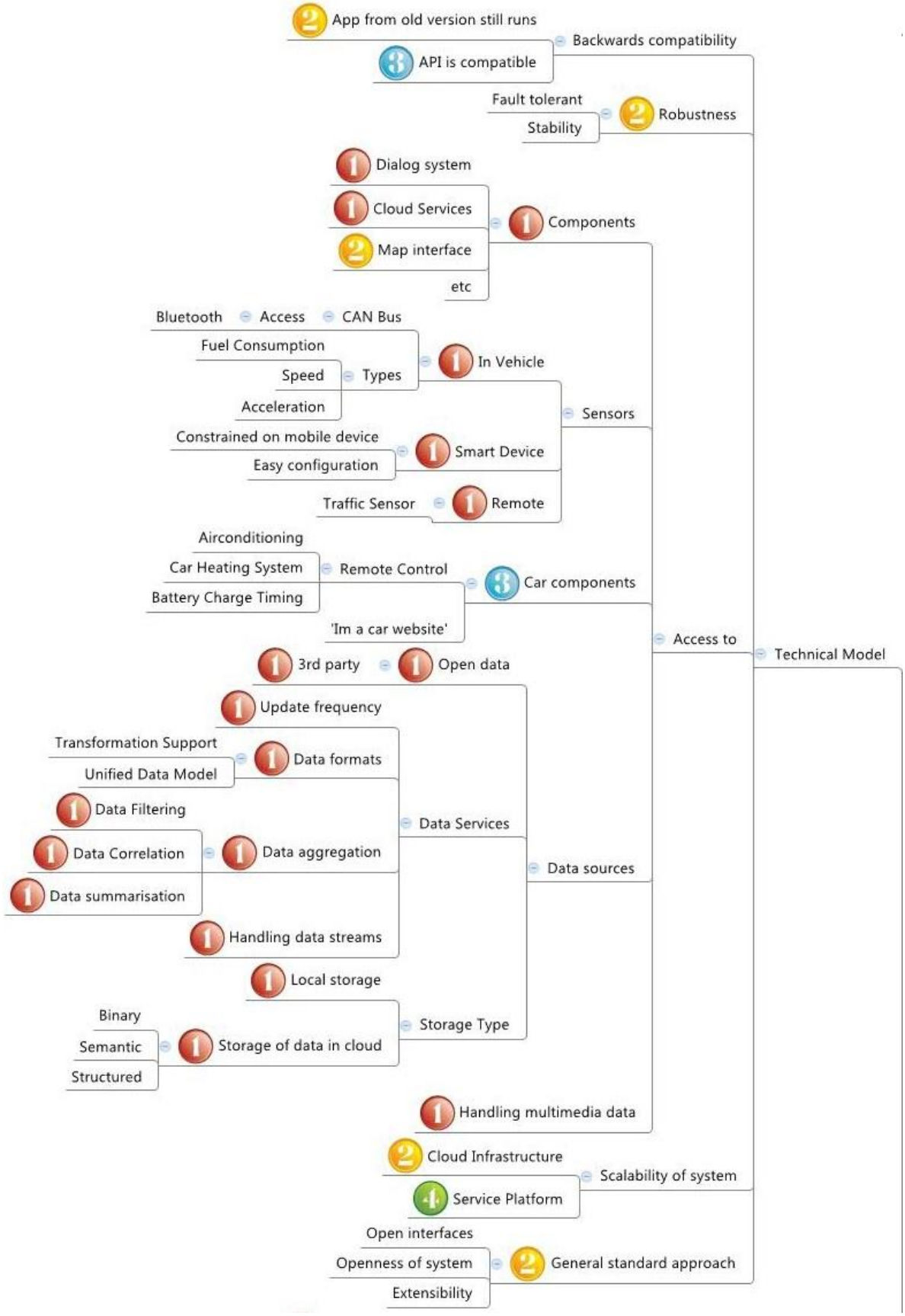

Figure 6: Requirements Mind Map – Developers 1

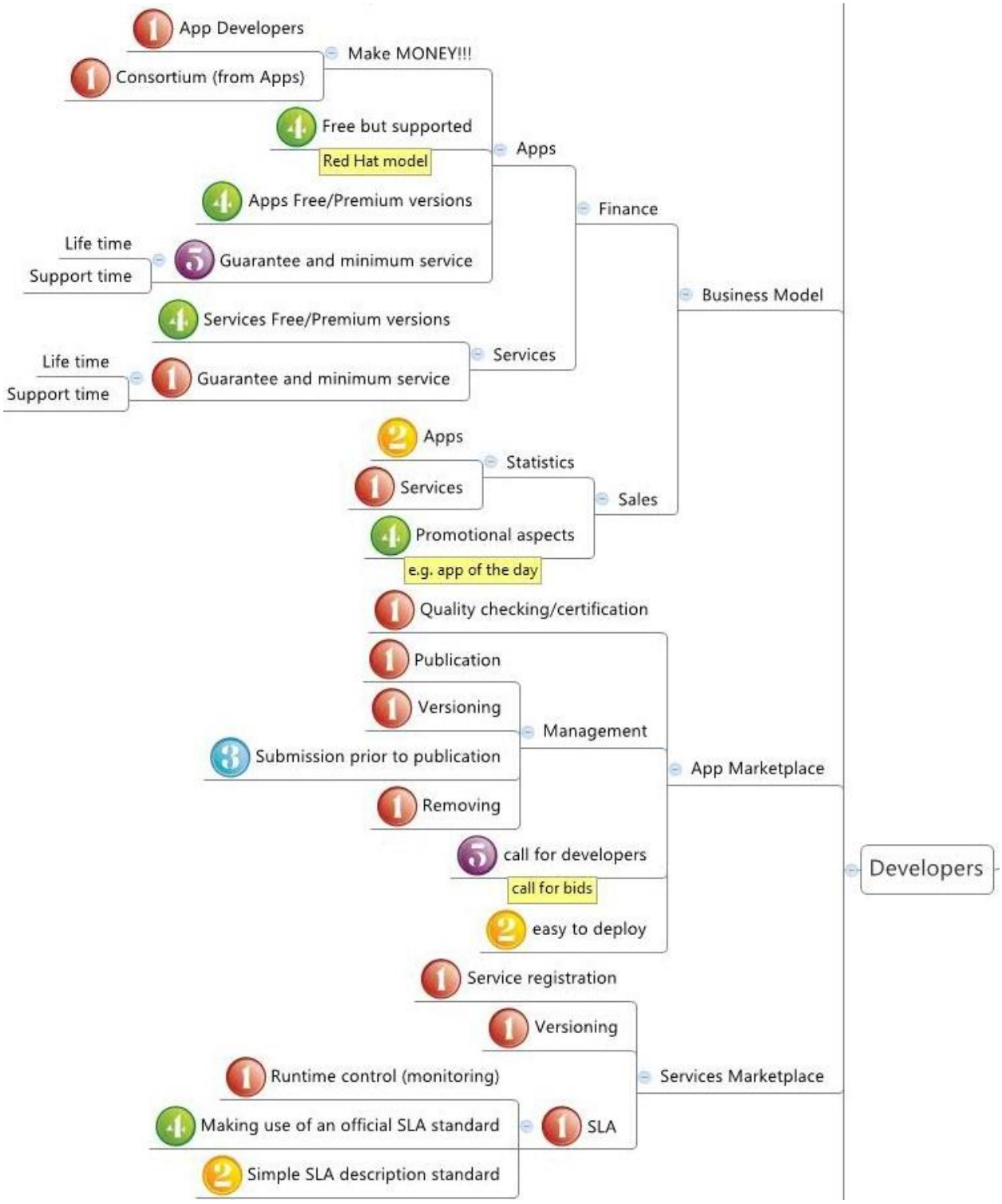

Figure 7: Requirements Mind Map – Developers 2

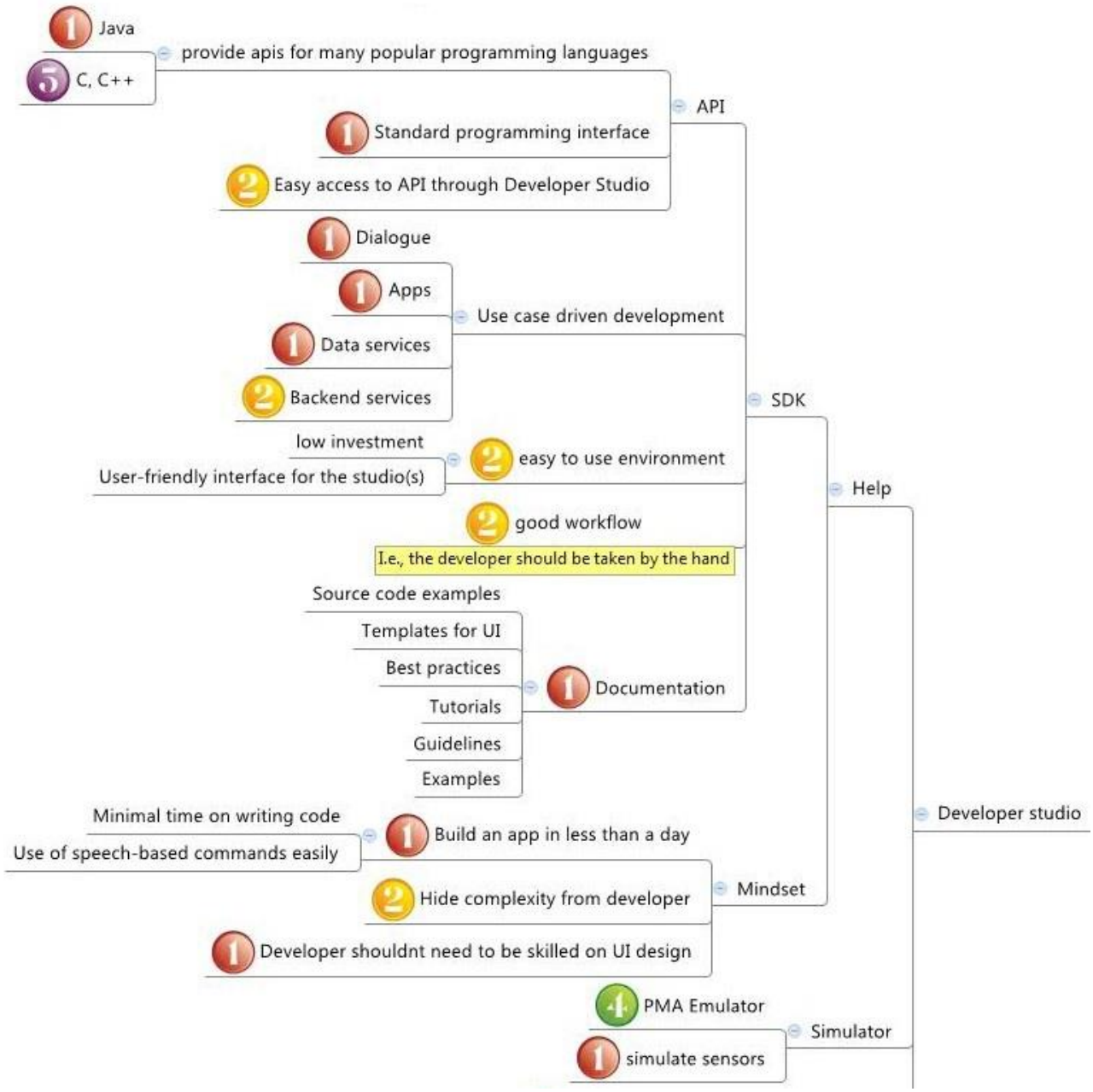

Figure 8: Requirements Mind Map – Developers 3

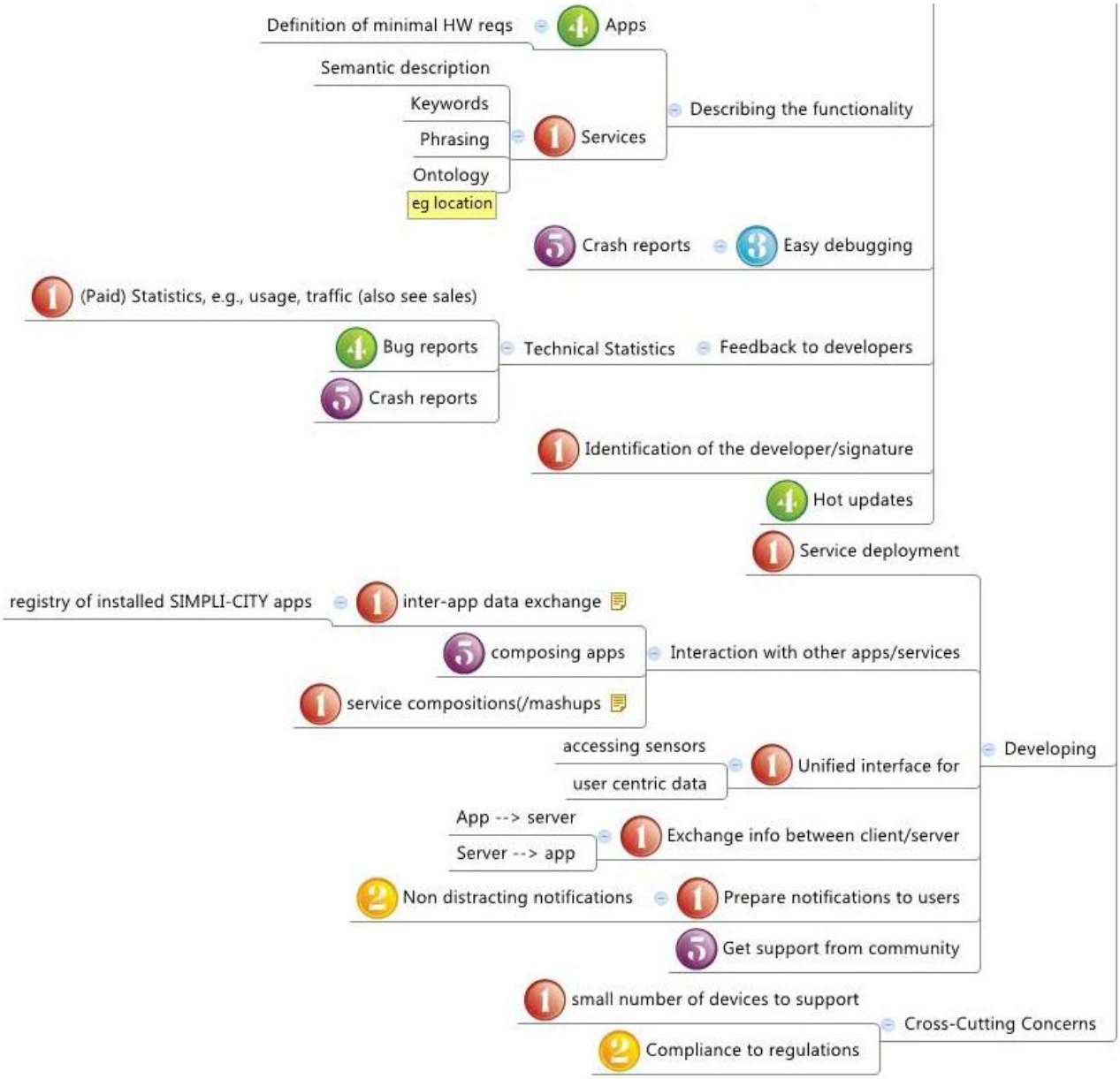

Figure 9: Requirements Mind Map – Developers 4