simpli·city

The Road User Information System Of The Future

WP2 – Vision and Requirements

D2.1: Project Vision Consensus Document

Deliverable Lead: TU Vienna

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

Delivery Date: 12/2012

Dissemination Level: Public

Version 1.70

This document will act as a guideline along the project SIMPLI-CITY and will be used by all partners to stay focused on the main ideas and goals of the project even in complex and technical phases of the project. This Project Vision Consensus Document provides information about the general positioning of SIMPLI-CITY, the project's business and research/technological objectives, its stakeholders, the underlying vision enablers, and some preliminary usage scenarios.

SEVENTH FRAMEWORK PROGRAMME

Document Status	
Deliverable Lead	Stefan Schulte, TU Vienna
Internal Reviewer 1	Sven Abels, Ascora GmbH
Internal Reviewer 2	Giuseppe Liguori, SRM – RETI E MOBILITÀ
Type	Deliverable
Work Package	WP2: Vision and Requirements
ID	D2.1: Project Vision Consensus Document
Due Date	31.12.2012
Delivery Date	21.05.2013
Status	Approved

Document History	
Draft Version	V0.1, TUDA, 04.12.2012
Contributions	V0.2, TUDA, 05.12.2012 V0.3, ASC, TIE, TUDA, IBM, TALK, TEMP, SRM, 06.12.2012 V0.4, TUDA, TUV, 07.12.2012 V0.5, TUDA, TUV, IBM, 08.12.2012 V0.6, TUDA, TUV, 10.12.2012 V0.7, TUV, 10.12.2012 V0.8, TUV, 11.12.2012 V0.9, TUV, 11.12.2012 V0.91, TUV, 12.12.2012 V0.93, TUDA, TUV, 13.12.2012 (First Review Version) V0.94, TUDA, 20.12.2012 V0.96, TUV, 21.12.2012 V1.00, TUV, 03.01.2013 V1.1, TUDA, TUV, 22.01.2013 V1.2, TUV, 23.01.2013 V1.3, TUDA, TUV, 24.01.2013 V1.4, TUV, TUDA, 08.02.2013 V1.41, IBM, 13.02.1013 V1.42/V1.43, TUV, 27.02.2013 V1.44/V1.45, CRF, 27.03.2013 V1.46/V1.47, ASC, 14.04.2013

	V1.50, TUV, 18.04.2013 (Second Review Version) V1.60, TUV, 21.05.2013 (approved by the internal reviewers)
Final Version	V1.70, TUV, 13.01.2014 (Approved by the European Commission)

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

Tempos 21, Spain

Centro Ricerche FIAT, Italy

SRM – Reti e Mobilità, Italy

Executive Summary

The purpose of this SIMPLI-CITY deliverable D2.1 Project Vision Consensus Document is to enforce a common understanding of the project vision and the overall aims and objectives of the project. In the further course of the project, this document – together with the Description of Work – will be used as a guideline to ensure that the partners are following the same goals. It will help to synchronise the general project objectives and the goal-oriented Research and Technical Development (RTD) work.

The SIMPLI-CITY Project Vision Consensus Document is based on the Description of Work (DoW) and is the foundation for the upcoming Requirements Analysis (deliverable D2.3), Global Architecture Specification (D3.1) and Functional and Technical Specifications (D3.2.1 and D3.2.2). It extends the DoW by a more thorough presentation of the individual software components and related interaction patterns. Furthermore, it provides more detailed potential usage scenarios.

Please note that the purpose of this deliverable is *not* to fully specify the behaviour of the components, since this will be highly influenced by the requirements specification in D2.3 as well as the specifications of work package WP3 (especially deliverables D3.1, D3.2.1 and D3.2.2). Nevertheless, this deliverable already presents a very detailed view on the functionality of the components, their structure, and their interconnection in order to keep the vision sharp and concrete. However, these aspects will be subject to change within the course of tasks T2.3, T3.1 and T3.2 and should therefore be considered preliminary.

In detail, this document first presents the general positioning of SIMPLI-CITY in terms of the challenges that the project is going to meet, the business and RTD objectives of SIMPLI-CITY, and general business opportunities facilitated through SIMPLI-CITY. Regarding the objectives, the business-related ones focus on the facilitation of business innovation and the application of Information and Communications Technology (ICT) for a low carbon economy. The RTD objectives cover the main research objectives with regard to the SIMPLI-CITY Personal Mobility Assistant (PMA), European Wide Service Platform (EWSP), and Mobility-related Data as a Service (MDaaS). This discussion is completed by a discussion of those aspects which will *not* be in the focus of SIMPLI-CITY.

Second, this document presents the stakeholders involved in and affected by the project. Third, the vision enablers in terms of knowledge and information representation and the single software components will be discussed. Fourth, preliminary usage scenarios will be presented and the basic SIMPLI-CITY interaction patterns will be drafted.

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.

Within this deliverable, the overall project vision in terms of its general positioning, the project's business and research/technological objectives will be discussed. Further, the project stakeholders, the underlying vision enablers, and some preliminary usage scenarios will be presented.

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.

1.2 Deliverable Purpose, Scope and Context

The purpose of this deliverable is to act as a guideline to the project. It will be used by all partners to stay focused on the main ideas and goals of the project even in complex and technical phases of the project. To achieve this goal, the Project Vision Consensus Document provides information about the general positioning of SIMPLI-CITY, the project's business and research/technological objectives, its stakeholders, the underlying vision enablers, and some preliminary usage scenarios.

The Project Vision Consensus Document provides rather high-level information. In order to get a deeper insight into the project, refer to the upcoming deliverables D2.3 (Requirements Analysis Report), D3.1 (Global Architecture Definition), D3.2.1 (Functional Specification), and D3.2.2 (Technical Specification), which provide more details. The presented use case scenarios are preliminary and will be altered and extended in the use case specifications developed in work packages WP7 and WP8.

1.3 Document Status and Target Audience

This document is listed in the Description of Work (DoW) as "public", since 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 at 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 is broken down into 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 positions the project in terms of its business, research and technological objectives and also discusses the areas which are out of scope of the project.

Section 3 characterises the project-internal and -external stakeholders of SIMPLI-CITY.

Section 4 states the SIMPLI-CITY vision enablers, both with regard to the considered data types and the needed software components, which are required to achieve the envisioned functionalities.

Finally, Section 5 provides different preliminary use case scenarios.

2 Positioning

Even against the background that several initiatives have already been successfully conducted to shift road usage to other means of transport, such as the railway or public transport in general, the number of vehicles in Europe and all around the world is still increasing. The significant penetration of mobile Internet and other technological advances in recent years, for example in the context of sensor technologies, have led to a multitude of information sources to support road users. However, the exploitation of these information sources is currently rather limited and focused on providing specialized, customized, and often proprietary systems such as navigation devices, which also offer additional information apart from their core functionality. Today's mobility-related services, data sources, and apps[1] are designed as standalone products. They typically lack the integration of data originating from different sources and proactive behaviour for sophisticated early interaction with the road user and rarely provide means for collaboration between different traffic participants. Consequently, the chances provided by being always-online and the different information sources are not yet fully exploited. Realising these potentialities offers the opportunity to create a huge range of new services and apps for road users, for example targeting at safety, comfort, and environmental friendliness during their journeys.

The full exploitation of different information sources which are useful in the context of road transportation requires efficient integration means and easy accessibility, combined with appropriate support for developing services and apps on the basis of these information sources. From the user's perspective, a huge number of different data sources currently make it rather hard to retrieve the right information in the right form and at the right time. A simple and intuitively usable interface, which abstracts from the specific different data sources used, is hard to implement, since at present, data sources are usually not explicitly targeting interoperability and end-to-end integration. As such, today's users have to cope with the situation that they have to employ and control different apps and often they need to manually transfer data output from one app to another. Such lack of integration and ease-of-use consequently leads to user's distraction and can result in dangerous situations, especially while driving. Instead, interoperable and automatically interacting apps and services are needed, which should be accessible through a simple, easy-to-use and non-distracting user interface.

As can be seen from this discussion, several challenges arise in the context considered, for example:

- Efficient techniques have to be provided that offer possibilities to integrate, store and access large amounts of data originating from different mobility-related data sources, thus leading to homogeneous interfaces for data access and data views.
- Efficient methods and techniques to develop various mobility, energy efficiency, safety, and comfort services based on the different integrated data sources have to be provided, allowing an easy way to offer these services' functionalities to users.

[1] Please note that in the context of SIMPLI-CITY, there is a big difference between "apps" and "applications". The former refer to end user apps running on a smartphone, i.e., software, while the term "application" has a more general and not necessarily software-related meaning. For more details, refer to the SIMPLI-CITY Glossary (see Section 1.4).

- Since the services are tailored to the needs of road users, difficulties of service provisioning and consumption in a mobile context have to be overcome, e.g., bad connectivity or disrupted transmissions.
- In order to foster the development and availability of different high-quality mobility-related services and apps, possibilities for easy and cost-efficient service development is needed. In particular, it is necessary to support interested third party service developers.
- Third parties need also an easy way to advertise the developed services and apps to end users.
- The usage and corresponding interaction between drivers and end user apps and their different – potentially complex – functionalities must be possible in an unobtrusive and natural way to minimize drivers' distraction and thus prevent critical situations.
- Since required information and the usefulness of mobility-related services and apps are situation-dependent, the detection of the current context of a road user is necessary to provide the right mobility-related service at the right time, and even allows a proactive behaviour supplying drivers with the maximum amount of supporting service and apps functionalities.
- Since several relevant data sources in the considered context comprise personal information of road users, reliable and trustable operation of the used mobility-related services and apps is a must. Hence, security and privacy measures have to be explicitly addressed.
- The amount of data available for different mobility-related services and apps is potentially very large due to the high number of available data sources. Hence, an efficient operation of mobility-related services and apps is depending on efficient ways to determine data relevance, aggregate data, historicize data, and reason on data against the background of a user's actual context.

In the following subsections, the project objectives will be discussed. The objectives of SIMPLI-CITY can be partitioned into business objectives (Section 2.1) and research and technological objectives (Section 2.2). The former-mentioned objectives do not only constitute the foundations for the overall project, but also provide contributions on their own through the prototypical realizations in the use case work packages (WP7 and WP8). The research and technological objectives will be in the focus of work packages WP3-6. They will deliver the necessary technical foundation to facilitate the business objectives, but also help to foster innovation in general by doing basic research.

2.1 Business Opportunities and Objectives

Even in times of financial crisis, the ongoing globalisation leads to an increasing competition in the mobility domain in Europe and all over the world. To be able to compete with new players from low cost countries like China or India, European automotive companies need to strengthen their position as innovation leaders. Here, SIMPLI-CITY comes into play, since it aims to facilitate business innovation on three different levels – the development of end user apps, contributions to the *Internet of Services*, and the exploitation of mobility-related data sources.

The SIMPLI-CITY approach to provide homogeneous and easy access to the different available heterogeneous mobility-relevant data sources, as well as to support road users with a unified and easy access to mobility-related services and apps, creates new

business opportunities. SIMPLI-CITY sets a distinct highlight on software in the rather hardware-driven mobility sector. It therefore stimulates the development of exclusive value-adding services around mobility and transport, which ultimately could lead to completely new markets in the mobility domain.

Further, the establishment of new markets for the envisioned mobility services and end user apps offers new possibilities to small and medium software vendors. Through the provision of according development Application Programming Interfaces (APIs), tools, and step-by-step procedures, such companies will be able to bring in their innovation capacities without the need for high capital expenditures. The Service and Application Marketplaces, which SIMPLI-CITY aims to establish, will lead to low market entry barriers, because these marketplaces are open to everyone. In contrast, current special-purpose app marketplaces in the mobility domain are controlled by automotive companies.

City operators may make use of the SIMPLI-CITY framework by providing easy-to-access mobility data relevant to their cities. The SIMPLI-CITY framework could even be a way to attract tourists and provide citizens with additional benefits, since it will show how day-to-day mobility-related data can be combined to make city life easier. SIMPLI-CITY could be viewed as one of the very first platforms managing mobility-related city data from heterogeneous domains in a dynamic way. This could spark interest from service suppliers and data providers.

In the following subsection, the facilitation of business innovation on different levels will be identified. SIMPLI-CITY objectives that address environmental aspects will be discussed separately in Section 2.1.2. In addition, possible post-project commercialisation of SIMPLI-CITY will be briefly discussed in Section 2.1.3.

2.1.1 Facilitation of Business Innovation

Without a doubt, for the end user, the most notable business innovation provided by SIMPLI-CITY is the idea of transferring the concept of mobile apps to the automotive sector. As it has been seen in other industries, most notably the mobile phone industry, adding a “software layer” to a hardware-driven product can have a significant impact on an industry, especially if it provides the possibility for third parties to bring in their expertise and ideas. To achieve this, SIMPLI-CITY will provide the facilities to develop and run apps on mobile devices, develop and integrate backend functionalities through services, and exploit data from heterogeneous sources.

SIMPLI-CITY aims to facilitate business innovation both for large, well-established companies, as well as smaller companies who may not be able to take part in the mobility industry so far since current road user information systems make use of proprietary technologies or are not opened to third parties at all. Based on the project results, automotive manufacturers and large suppliers can integrate the app concept into their own systems and set up their own app markets, thus allowing them to offer new products and services in the market and show the innovation potential of the European industry. Even companies that are currently not involved in the automotive industry will benefit from the tools to design apps and services and the possibility to advertise them on according marketplaces. By providing an open, unified environment, SIMPLI-CITY decreases the market entry barriers for new players.

As SMEs commonly have to cope with restricted capital and workforce equipment, they will especially benefit from lowering the initial costs to enter a new market. Both the Service and Application Marketplaces are therefore business facilitators, enabling

potentially thousands of service and app developers to market their innovation power and expertise, bringing in new ideas to the mobility industry.

Of course, personal mobility is an important aspect of the daily life of European citizens and therefore a business opportunity in its own right. The facilitation and development of prototypical services aiming at general mobility as well as related energy efficiency-, comfort-, and safety-related services will lead to benefits both for the individual citizen as well as the overall community. This is especially illustrated by the “personalized traffic restrictions” use case regarded in work package WP7, which will help lessening the exposure of residents of a particular city area to through-traffic as well as making the journey of single road users easier. For a more thorough discussion of this use case, refer to the Use Case Definition deliverables D7.1.1 and D7.1.2.

Last, but not least, companies will profit from the integration of data from various sources, which is an important, increasing market enabler. For example, in 2011, the European Commission has estimated the potential economic gains from open (government) data to be €40 billion each year in the European Union alone[2]. Similarly, the German Federal Ministry of the Interior calls government data the “‘raw material’ for start-ups and new business models”[3]. By providing the means to easily integrate such data sources into services and app, SIMPLI-CITY will help to unleash the potential of (open government) data, leading to new, innovative services and end user apps.

Several factors play a role in the assessment of the outcomes of the project’s business objectives, most importantly the general ability of the project to provide the necessary underlying technologies (data integration, service framework, Personal Mobility Assistant) and their applicability. As the first assessment test, the outcome of the use case work packages will initially show whether SIMPLI-CITY is able to meet its business objectives. Furthermore, meeting the business objectives can only be deemed as a success if the awareness of the project is raised. This is necessary in order to ensure that the ideas, approaches, and technologies provided by SIMPLI-CITY are taken up by companies. In SIMPLI-CITY, project prominence is provided through workshops, expert engagement, standardisation activities, newsletters, and collaborations with EU project clusters and other initiatives.

2.1.2 ICT for a Low Carbon Economy

SIMPLI-CITY aims to provide a range of innovative approaches for achieving environmental friendliness of mobility and transport by alleviating the development of according apps. One particular prerequisite for this is the provision of timely, accurate, and detailed carbon footprint data that can then be used in data-based services and apps and may even lead to new business models aiming at carbon-efficient mobility.

Therefore, SIMPLI-CITY will develop the means to increase the awareness of the individual carbon footprint of personal transportation. Unlike existing approaches, SIMPLI-CITY will not be restricted to coarse-grained carbon footprints based on average values. Instead, it will enable services to automatically make use of relevant data such as the actual mileage (by acquiring the necessary information from the car systems) or estimating it based on the number of car passengers. This will allow users to track the carbon

[2] http://ec.europa.eu/information_society/policy/psi/docs/pdfs/opendata2012/open_data_communication/en.pdf

[3] http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/OED_Verwaltung/ModerneVerwaltung/opengovernment_kurzfassung_en.pdf?__blob=publicationFile

footprint of their daily journeys in a fine-grained, transparent, and accurate way. SIMPLI-CITY will develop the means to compare the carbon footprint with those from other journeys, drivers, and alternative transport modes by allowing services to access historical data. Thus, carbon footprint data can be utilized in mobility-related services and apps, raising the environmental awareness of road users and promoting a more environmentally friendly behaviour.

A major step towards a low carbon economy will be to provide the driver with information on possible road as well as public transport connections. For example, drivers who are heading to a concert may be informed where to park close to the most convenient public transport connection to the concert hall. Thereby, fuel consumption can be reduced by avoiding further driving into traffic-congested areas. At the same time, awareness of available sustainable alternatives for transport and mobility is raised among the driver and the other car passengers.

In order to achieve this, SIMPLI-CITY will implement exemplary use cases that focus on raising the environmental awareness of drivers. In addition to information about fuel consumption and carbon footprint of a journey, SIMPLI-CITY will give recommendations to the driver in order to let him be more eco-friendly. Another way to increase the environmental awareness of drivers is an eco-aware driving competition where driving styles will be compared to each other.

The consideration of environmental aspects will be assessed in different ways: First, with regard to the support for a fine-grained, accurate determination of the carbon footprint at the level of individual journeys. Second, regarding apps based on this data, i.e., apps providing analysis and comparison functionalities – exemplary apps will be prototypically implemented in the use case work packages (WP7 and WP8). The success to meet the envisioned outcomes of this objective will be directly reflected by the project’s ability to deliver the use case services on personalized traffic restrictions and the eco-driving contest.

2.1.3 Post-Project Commercialization of SIMPLI-CITY

In addition to the wider business opportunities presented above, the consortium itself aims at driving SIMPLI-CITY forward even after the project ends. To realize this, the consortium aims at implementing a set of business opportunities that will allow the financing of further development after the end of the project, as well as the maintenance and reimbursing of ongoing hardware/software costs. Possible business opportunities include:

- Commercialization of the platform: The consortium may ask for license fees from project-external Original Equipment Manufacturer (OEM) partners that want to integrate SIMPLI-CITY into their devices. This may include pure one-time license costs for the installation itself as well as ongoing license costs on a per-user basis.
- App and service sales: SIMPLI-CITY will allow app providers to create new mobile apps which will run within the SIMPLI-CITY Personal Mobility Assistant (see Section 2.2.1). Those apps will be offered in the app market as described in Section 4.2.9. SIMPLI-CITY may allow the publication of commercial apps and may take a share of each sale, e.g., up to 30%, which is a normal figure in other app markets such as the Apple AppStore, Google Play, or Windows Phone Store. The same model may be applied by service providers that offer their services within the SIMPLI-CITY framework. SIMPLI-CITY may in this case charge on a per-request base with a small amount per request.

- Upsells: The consortium may offer a 100% free version of the project outcome and all services together with an additional commercial premium version which would allow the operator of a SIMPLI-CITY framework instance a tighter integration and customization of the platform to its needs and branding. This model is referred to as “freemium” approach.

2.2 Research and Technological Objectives

The SIMPLI-CITY project pursues research efforts and technological development in the fields of Mobility-related Data as a Service (MDaaS), Service-oriented Computing, and Mobile Computing. SIMPLI-CITY will integrate and extend concepts and approaches from these fields.

The research results of SIMPLI-CITY will be published by research-oriented partners (also in cooperation with other members from the consortium) in scholarly journals, in edited books, and furthermore distributed through the participation in scientific conferences and workshops. This way, the results of SIMPLI-CITY will be made visible to the international research community.

In addition to the scientific research community, the non-scientific community will benefit from the results of SIMPLI-CITY. In order to improve and foster the use of public transportation and intermodal mobility solutions, workshops, talks and refereed publications will be made available to potential operators and users of the SIMPLI-CITY results such as (but not limited to) public transport providers, transport experts, public authorities, municipalities and other associations across Europe. The applicability of SIMPLI-CITY results will be ensured through standardization activities.

SIMPLI-CITY follows an incremental and prototype-driven development approach with clearly defined milestones. Hence, the advancements in these objectives can be monitored throughout the project based on the degree to which the prototypes meet the requirements. The general approaches to measure the project’s outcomes with regard to the different RTD objectives will be part of the discussion in the following subsections.

2.2.1 Personal Mobility Assistant

To the average end user, the research efforts on the Personal Mobility Assistant (PMA) will be the most visible. The PMA is a full-fledged, voice-based multimodal end user interface as well as runtime environment that allows the execution of SIMPLI-CITY-facilitated apps. It will operate on mobile devices, such as smartphones or tablets running Google Android, Apple iOS or Microsoft Windows Phone. The decision which mobile operating system will be chosen is part of the project’s requirements engineering.

While a number of mobility assistants are offered by automotive companies and manufacturers of navigation devices, they are until now proprietary, mostly not open to third parties, and therefore limited to services provided by selected companies. In contrast to that, the SIMPLI-CITY PMA will be the first non-proprietary, open software system that allows to cost-efficiently extend the functionalities of the PMA by third party software developers through the integration of new apps.

From the user’s perspective, the most significant impact will be provided by the voice-based user interface of the PMA. Within cars, a voice-based user interface is a natural choice, since such interfaces provide user interaction with a lesser degree of distraction and lead to a higher degree of safety. Current automotive user interfaces are not very

sophisticated and do not provide the flexibility and user friendliness that will be achieved by the software developed within SIMPLI-CITY. For example, many voice-based user interfaces are designed for a “train during use” mode, i.e., users learn the system commands while they are using the system. This forces the user to use only the kind of sentences predefined by the system designers and thus expected by the system, in order to trigger an action. The problem is that variations in a question or in a sentence that have not been included in the system grammars are not comprehensible for the system. Therefore, for the PMA in SIMPLI-CITY, a voice-based multimodal end user interface will be developed, which provides an intuitive usage of the system and is able to mirror human dialogues.

To measure how SIMPLI-CITY achieves this objective, the PMA will be compared with proprietary products, and a use case evaluation will be carried out. It is planned to assess the prototype’s quality according to the usability criteria of the ISO/IEC 9126-1 quality model, i.e., understandability, learnability, operability, attractiveness, and usability compliance. The assessment of the dialogue interface will be done according to PARADISE[4], or a subset of PARADISE parameters, e.g., with regard to success rate, task completion time, turns needed to complete a task, subjective assessment, etc. Besides testing the functionality, extendibility, and usability of the PMA, it will also be evaluated in a use case scenario in order to measure driver behaviour, i.e., his or her distraction following standardised tests, driving mistakes done, etc.

2.2.2 European Wide Service Platform

Another objective of SIMPLI-CITY is to create a European Wide Service Platform (EWSP), which will be realized as the SIMPLI-CITY Mobility Services Framework. This framework will be the basis for providing drivers and other users with a large range of services by intelligently combining wireless communication technologies, developing required framework subsystems, and producing innovative, prototypical interoperable services.

To implement the framework, SIMPLI-CITY will drive and enhance research in the field of Service-oriented Architectures (SOA) and Service-oriented Computing (SOC) in order to facilitate the development, description, discovery, provision, personalization, and administration of services. Even though SOA and SOC are well-studied research topics, there are still some research questions that need to be taken into account; SIMPLI-CITY considers mobility-specific aspects and mobile services in connection with the other research and technological objectives. Mobile services are defined as services (of any type and purpose) that are invoked by a wireless mobile device. In such an environment, the invocation of a service can lead to problems arising from the possibly extensive message overhead. Furthermore, service invocation is done via potentially unstable connections, i.e., a particular backend service could be invoked by an app and afterwards, there is a connection loss. These factors need to be taken into account in the project.

Further, the integration of context information is important for developing personalized services, like the planning of a trip to an event by considering personal and context information, such as preferences of transportation modes, or location-based selection of data services. While context-based solutions for service personalization have already been proposed, existing approaches are usually not domain-specific and therefore only

[4] M. Walker, D. Litman, Candace, A. Kamm, and A. Abella, “PARADISE: a framework for evaluating spoken dialogue agents,” in *35th Annu. Meeting of the Association for Computational Linguistics and Eighth Conf. of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, 1997, pp. 271-280.

applicable to some degree, as their functionality is not specific enough. SIMPLI-CITY aims at solving this by providing a more fine-grained, domain-specific personalization approach.

In addition to this, SIMPLI-CITY aims at the facilitation of third party service development and marketing of services. Therefore, SIMPLI-CITY will develop a Mobility Service and an Application Marketplace, where third party developers can promote and further provide their services and apps to potential users. In addition, SIMPLI-CITY makes it possible for different cities, automotive companies, and other service providers to offer their services in individual marketplaces to the users.

In order to verify this objective, SIMPLI-CITY Mobility Services Framework functionalities will be assessed in terms of enabling energy efficiency-, mobility-, comfort-, and safety-related services. This includes the support of third party developers for easily implementing software and providing their services and end user apps. Last, but not least, a comparison with other service frameworks will be conducted.

2.2.3 Mobility-related Data as a Service

The integration of different sources and their data types through sensors, actuators, user data, and open government data is a prerequisite to implement the envisioned SIMPLI-CITY services and apps. Hence, another major contribution of the project is the description, provision, discovery, exchange, fusion, and storage of data from various, heterogeneous data sources, such as cooperative systems, telematics, intelligent infrastructures, sensors and sensor networks, media data streams, open (government) data repositories, people-centric sensing or user data in general.

SIMPLI-CITY will conduct research and technological development to access different types of data in an easy, unified way. Second, after these sources can be accessed and viewed via according interfaces, wrappers, and connectors, it is necessary to easily combine, query, and integrate historical and real-time data into services. Some functionality for easily analysing historical versus real-time information from streams is also required, mainly for cross-correlating information from data streams. By correlating information across multiple streams, it is ensured that common information is linked. The latter is strongly required for interpreting and contextualizing information from various streams. Both data access and data integration will be supported by methods, software solutions, and models developed within the SIMPLI-CITY project.

Following the MDaaS approach, SIMPLI-CITY will provide unified, yet domain-specific, data access and data models. For this, specific characteristics of both the mobility domain and the already mentioned mobility-typical data sources will be analysed and taken into account. As privacy and security are especially important in this domain, data modelling will consider this aspect from the beginning of the project.

First, the MDaaS research results are evaluated by assessing the efforts necessary to integrate data into the SIMPLI-CITY services and apps, i.e., how much work a software developer has to do in order to integrate data from a previously unknown source. The second assessment criterion is the number of different data source types (both in terms of format and dynamicity) SIMPLI-CITY is able to support. The third assessment criterion is the number of stream sources SIMPLI-CITY is able to support simultaneously at query time. The fourth assessment criterion is the ability of SIMPLI-CITY to allow abstract (i.e., generic), contextualized (i.e., cross-correlated) and summarized (i.e., compressed) views on single and multiple data streams. From a more technological perspective, SIMPLI-CITY will be assessed based on the robustness of the Cloud-based storage, i.e., is the Cloud

storage robust enough for retrieving complex objects, which may include a large number of historical records.

2.3 Out of Scope Areas

Topics that are out of scope for the SIMPLI-CITY project are as follows:

- SIMPLI-CITY does not aim at support for loosely coupled backend services in terms of late binding and service selection at runtime. End user apps therefore make use of predefined backend services instead of choosing these services during the runtime of the app. However, it is possible to apply late binding for data services.
- SIMPLI-CITY will not aim at the development of a new routing and trip planning solution. Instead, it will make use of existing, free APIs from Google Maps, Map24.de, etc.
- SIMPLI-CITY will not aim at data cleansing for open government and other data repositories. Data from such sources will be used as it is.
- The SIMPLI-CITY PMA will not aim at providing support for different languages. It will focus on British English only but will provide the means to extend it with other languages.
- SIMPLI-CITY will not develop new tools and APIs for speech recognition. Instead, existing solutions will be applied and combined with input from partner Talkamatic.
- SIMPLI-CITY will focus on enabling the execution of end user apps on mobile devices like smartphones. While the integration into car head units is imaginable, there will be no particular implementation of such functionality.
- SIMPLI-CITY does not aim at developing a completely new service registry. Instead, existing solutions like the Membrane SOA Registry or the OSGi services registry will be assessed regarding their applicability and extended in order to serve the purposes of the project.
- SIMPLI-CITY does not aim at integrating all existing types of data formats, but instead only provides generic data integration tools for achieving SIMPLI-CITY scenarios.
- SIMPLI-CITY does not aim at developing innovative security mechanisms for accessing Cloud-based data. Nevertheless, standard security and privacy mechanisms will be put into place.
- SIMPLI-CITY will not target the handling and management of Intelligent Property Rights (IPR) regarding the data, information, and knowledge created during the usage of the SIMPLI-CITY PMA. Thus, such IPR considerations have to be included separately and are out of the project’s scope.
- SIMPLI-CITY does not address and consider cross-border (legal) aspects and issues that may arise due to different regulations and laws prevailing in different European countries. Such aspects will be subject to juridical compliance validations, which are out of the project’s scope.

3 Stakeholders

In order to achieve the goal of SIMPLI-CITY to provide the road user information system of the future, needs and requirements of different stakeholders must be considered. In the following table, different stakeholder roles that are relevant to SIMPLI-CITY are identified and individual project outcomes are described per stakeholder.

Stakeholder	Project Outcome: Fit and Impact
Road Users (Drivers)	
General	SIMPLI-CITY improves comfort and safety for road users by providing an easy-to-use and unobtrusive means to uniformly access and combine different data sources, services, and apps, which provide journey-relevant information. Furthermore, proactive and context-sensitive driver support based on these different data sources, services, and apps is realized. Contributing technologies include Web services, advanced speech recognition, smartphone technology, sensor technology, and flexible data exchange.
Public Authorities	
General	SIMPLI-CITY allows provisioning of real-time mobility data to road users and means to employ this data context-sensitively. This provides drivers with more comfort and safety and also contributes to relaxing traffic situations and creating more environment-friendly traffic in urban areas, e.g., by enabling context-sensitive traffic routing during major events or enabling easy and timely interchange between different means of transport and co-modality.
SRM – Reti e Mobilità	SRM, as the Public Transport Authority of Bologna, plays a very important role at the local level. SRM provides a link to local authorities, transport operators and transport researchers under its umbrella, providing decision makers with support for planning, awarding, monitoring, and evaluating the transport service and its network. SRM is a transport stakeholder acting at the local level but with a watchful eye on the strategic importance of the wider knowledge and expertise coming from the participation in European projects. The approach of SRM is to focus on the passengers' perspective. In SIMPLI-CITY, SRM is a user partner and will act in close cooperation with the Municipality of Bologna. SRM will be especially involved in the use case work packages. SRM will also lead the definition of the Target Market Sector Descriptor (deliverable D2.2), where the scope of the Research and Technical Development (RTD) within SIMPLI-CITY will be clearly defined and the target market will be assessed in depth, in order to ensure that SIMPLI-CITY solutions are needed, updated, and used by the target audience. Another important contribution will be made by SRM in the field of data collection and integration (WP4): starting from the actual situation, SRM will not only collect sources of data at the local level for SIMPLI-CITY, but it will also act in order to foster a standard methodology for data access, following an open data model. Such an effort could lead to a broader sustainability of all the research activities based on mobility data.

ICT Companies	
General	One of the SIMPLI-CITY main targets is the definition of new ways towards the efficient development of mobility-related services and apps and an efficient means to sell these to customers. Hence, SIMPLI-CITY provides ICT companies with the opportunity to broaden their market reach by developing and selling new mobility-related services and apps.
Ascora GmbH	Within SIMPLI-CITY, Ascora is responsible for the Application Runtime Environment. Ascora has experience in delivering software for smartphone devices that require such frameworks. Ascora's development results in this context have, for example, been published in the Apple AppStore, in Google's Play Store, and in the Windows Phone Store, and thus are tested in real-world scenarios. Ascora will use the SIMPLI-CITY results to strengthen their own product line. It will reuse the components to extend existing products and to benefit from the software developed in SIMPLI-CITY. For example, Ascora will make use of the service infrastructure framework developed as part of the service platform to improve the structure of the Ascora FIPS system for customer data management. FIPS is currently integrating components from various payment providers into a customer data management system, but is lacking a flexible service interface for integrating new Web-based data sets. Second, Ascora will create own services and apps for the SIMPLI-CITY platform. Ascora is also responsible for the Mobility Service and Application Marketplaces and has a high interest in collecting data with respect to running a marketplace for apps and services. Ascora will profit from the results and is then able to fine-tune the creation process of new apps for the SIMPLI-CITY marketplaces. Third, since Ascora is one of the key players of the project, it will also offer active consulting for Cloud service providers and therefore benefit from the market success of SIMPLI-CITY.
TIE Nederland B.V.	TIE has already invested heavily in service technologies and recently has begun explorations on how to apply these in other areas beyond eCommerce such as in the Net-Challenge Factories of the Future research project and now the SIMPLI-CITY project. Thus, TIE will morph the ideas, knowledge, specification, and components of the project into existing products and services and use the knowledge, e.g., during prospecting and sales processes. As an experienced provider of software solutions, TIE will participate heavily in the architectural part of the project, working on the functional and technical specifications as well as integration of the final product. TIE will employ its wide technological partner and customer network to disseminate information about the project and raise awareness. As a player in the field of business integration and SOA, TIE is going to participate in the development of the SIMPLI-CITY Mobility Services Framework helping to integrate data coming from different sources into a single reusable and standardized stream accessible by SIMPLI-CITY. TIE is going to use results coming from its work on SIMPLI-CITY to strengthen its position in the business integration market and reuse knowledge of various technologies used in the project in new and existing TIE products.

IBM	IBM Research Smarter Cities Technology Centre will investigate how advanced analytics, reasoning, Cloud, stream, and semantic technologies can help city authorities make optimal use of resources and so meet the challenges of our increasingly urbanized world. In SIMPLI-CITY, the main contribution of IBM Research will be the MDaaS approach. This approach unifies the access to data coming from technologically heterogeneous sources that are originally only available in different formats. The impact of the data integration activities of SIMPLI-CITY can be divided into two major aspects: First, SIMPLI-CITY provides methods and software solutions to access different types of data, e.g., sensors and sensor networks, cooperative systems, open (government) data repositories on the Web, media data streams, and user-centric data, in an easy, unified way. Second, after these sources can be accessed via corresponding interfaces, wrappers, and connectors, SIMPLI-CITY will make it possible to easily combine, query, and integrate historical and real-time data into services. For this, a unified data model for the mobility domain will be conceptualised and implemented. The data model will be the foundation for the storage of mobility-related data within the Cloud-based Information Infrastructure, thus making data management and access scalable. Mainly IBM will focus on (i) techniques to represent heterogeneous data from various sources (users, sensors, open government), (ii) ensuring that data is safely accessible, and (iii) various basic views can be presented. From an exploitation perspective, IBM will consider integrating SIMPLI-CITY results into its already in-place transportation technologies, e.g., IBM Public Transportation Awareness asset.
Talkamatic	Talkamatic builds and designs software for voice interaction, based on research on human dialogue. The primary product is the Talkamatic Dialogue Manager (TDM), which implements a number of scientific theories about human dialogue. Talkamatic strives to deliver natural, efficient and low-distraction dialogue for in-vehicle and other uses. Talkamatic will use the project as a testbed for using the TDM as an interface layer for a wide variety of apps and services. Talkamatic will also put an emphasis on moving towards dictation-based recognizers, such as Nuance's Dragon Mobile and the Android Automatic Speech Recognition (ASR) service. This requires the implementation of more robust interpreters for natural language. Talkamatic will also work on solutions for non-distracting ways to handle system-initiated dialogue.
Tempos 21	Tempos 21 is the Mobile Competency Center of the international ICT company Atos. Tempos 21 has more than eleven years of experience in designing and developing mobile apps and services, as well as mobile frameworks. The main contributions of Tempos 21 in SIMPLI-CITY will be the development of the SIMPLI-CITY Mobility Services Framework in WP5, and the definition and implementation of the different use cases in WP7 and WP8. The SIMPLI-CITY Mobility Services Framework will be the interface between the integrated data sources and the end user apps. Tempos 21 plans to exploit different outcomes of the project. First of all, it plans to incorporate some functionalities of the Application Marketplace implemented within WP5 in a similar product it is currently implementing and commercializing (MyMarket). Moreover, it plans to exploit the implementation of the use cases of the project by incorporating some of the apps developed within its product portfolio, or by extending current ones.

	Finally, Tempos 21 will benefit from the knowledge gained in the development of the different components of the Mobility Services Framework, namely the Service Development API and the Context-based Service Personalisation component, and it will try to incorporate it in the Context Broker Platform of the Connected Car product line of Atos.
Automotive Industries	
General	SIMPLI-CITY will allow the automotive industry to develop and offer to its customers an enhanced, more comfortable and more user friendly (e.g., through the use of voice-based interaction modes) driving experience. Furthermore, functions and services included in the SIMPLI-CITY platform will contribute to improve traffic efficiency and quality of the environment while reducing fuel usage, thus enabling a sustainable deployment, which is consistent with the time envisioned time horizon of 2020.
Centro Ricerche FIAT	Within SIMPLI-CITY, Centro Ricerche FIAT (CRF), as research centre of FIAT, will operate twofold: from one side, the project outcome will represent the ground base for novel solutions for the next generation of connected vehicles in the direction of smart mobility solutions, from the other side, project results will be presented to the public authorities (e.g., Comune di Torino) to support the design and implementation of the smart cities of the future. In general, the approach is to support the engagement in a synchronized, public effort that creates in all market participants the confidence that the technology is ready, that all industry and public partners are committed, and that within a foreseeable time frame, the coverage of the major geographical areas is ensured. In particular, CRF is an independent legal entity, however it is an integral part of FIAT Research and Development and therefore the transfer of related project outcomes will be done with a direct link to support planning of novel vehicle solutions.
Researchers	
General	On the scientific side, SIMPLI-CITY will drive and enhance research in the field of Service-oriented Computing and Cloud Computing, as well as in the fields of Speech Recognition and Sensor Technology. This is achieved by integrating artefacts and heterogeneous data sources from the Internet of Services and from the Internet of Things with speech-based means of user interaction in a service and app platform tailored for road users and their needs.
Vienna University of Technology	As a research partner, Vienna University of Technology and the Distributed Systems Group focus on research and teaching tasks. Within SIMPLI-CITY, apart from its role as project coordinator, TU Vienna will focus on two key aspects: First on the conceptualization and implementation of the MDaaS approach with regard to sensor data, user-centric data, open (government) data, and media streams. Second, on research activities with regard to the domain-driven extension of current service framework approaches in the SIMPLI-CITY Mobility Service Framework, including context-based service personalisation. TU Vienna aims at publishing SIMPLI-CITY research results both at domain-driven as well as technology-driven high-tier conferences and journals. Outcomes from the project will be used within TU Vienna’s teaching, e.g., in undergraduate and graduate courses to exemplify the usage of sensor and service technologies, or as a foundation for student theses.

Technische Universität Darmstadt	The Technische Universität Darmstadt focuses on research and teaching tasks. It will thus primarily benefit from SIMPLI-CITY by applying the SIMPLI-CITY platform to drive empirical and theoretical research in the area of Service-oriented Computing, Cloud Computing, and the Internet of Things regarding service composition, Cloud-based data and service storage, provisioning and discovery, as well as sensor integration. Research results will be published in different scientific outlets, such as scholarly journals, scientific conferences, workshops, etc. TU Darmstadt further uses the SIMPLI-CITY application domain and project outcomes as examples for the usage of the Service-oriented communication paradigm and Cloud Computing technologies, as well as to disseminate different integration possibilities for heterogeneous sensor technology in seminars, lab exercises, and students' projects.
Forschungsgesellschaft Mobilität	FGM's core business is all-in-mobility solutions. Part of its business portfolio is the creation and provision of real-time information systems and of static routing systems for public transport providers and associations, including basic conditions such as Geographic Information System (GIS) data creation and processing. SIMPLI-CITY increases FGM's ability for future research and development activities both nationally and internationally in the field of real-time routing systems for intermodal trips, also, e.g., for special user groups (such as tourists). Accordingly, SIMPLI-CITY contributes to the enlargement of FGM's business branch of information systems from public transport to a comprehensive set of transport modes and multimodality as well as from static routing systems to real-time, data-based ones. This also enables FGM to provide (semi-)public transport providers and associations with innovative and easy-to-use apps for increased use of intermodal mobility solutions.
Other Stakeholders	
Standardisation Bodies	The consortium covers several partners that have expertise and are active in various fields of standards relevant to SIMPLI-CITY. For example, FGM was already involved in different standardisation activities concerning mobility and transportation, and IBM is actively involved in the World Wide Web Consortium (W3C). TU Vienna, TIE, Ascora, and Tempos 21 (being part of the Atos Group) are involved in the Networked European Software and Services Initiative (NESSI), which is an important incubator for upcoming standardisation efforts. Thus, SIMPLI-CITY will not only gain insights in current activities in relevant fields of standardisation, but also possesses the possibility to provide project results and findings as input to different standardisation bodies.

4 Vision Enablers

4.1 Knowledge and Information

The key enabler for all apps and processes within SIMPLI-CITY is information and the integration and exploitation of it. This information comes from various sources, which can rather coarse and non-exclusionary be distinguished into the following categories:

- *Sensor Data*: Within Intelligent Transportation Systems and cooperative systems, sensors provide a huge range of mobility-related data, e.g., about the current traffic situation at a particular location, traffic light switching cycles, weather conditions, or about vacant and occupied parking spots. Depending on whether this data comes from public or private sources and the potential for abuse, it may require privacy-aware treatment.
- *Open Data*: An increasing number of organisations, including public authorities, provide historical and real-time transport-related information on the Web. Such sources may, for example, include data about congestions, road traffic counts and estimates, accident casualties, or speed statistics. Due to the public nature of such data, privacy requirements are of very limited relevance, given that an adequate pre-processing (such as cleaning or aggregation of data) has been conducted by the releasing organisation.
- *User-centric (Context) Data*: User-centric data is another valuable input for mobility-related services and can be provided in many different types. For example, calendar data (providing, e.g., information about future events and locations), which is combined with a navigation service, as well as data from in-vehicle telematics (showing, e.g., if the car needs to be refuelled) may be combined with other data sources, such as an online list of garages offering natural gas. The most important source of user-centric (context) data is, without a doubt, spatial and temporal data. This information is potentially very sensitive and thus requires stringent enforcement of data privacy mechanisms.
- *Media Streams*: Media streams are especially used in SIMPLI-CITY-facilitated services and end user apps aiming at comfort and leisure functionalities. Examples include music or video streams, such as offered by popular Web platforms like Deezer, Last.FM, or SoundCloud.
- *Holistic Privacy and Security Concept:* Since SIMPLI-CITY often deals with highly intimate data, such as appointments of users, holistic privacy and security concepts are a crucial topic in SIMPLI-CITY. Therefore, privacy relevant data will be classified as such and anonymized or pseudoanonymized and encrypted and protected by Role-based Access Control (RBAC).

In the following, the four introduced data categories are explained in detail, followed by a description of the holistic privacy and security concept.

4.1.1 Sensor Data

With the increasing spread of information technology into all areas of life, which marks the rise of Ubiquitous Computing, sensing technology has also seen increasing deployment. Sensor data refers to any information that has been acquired by such sensing technology.

In the context of SIMPLI-CITY, the focus is on mobility-related information, originating from Intelligent Transportation Systems and cooperative systems. For example, road administrations in many countries have already installed underground sensors for measuring traffic density and traffic flow several years ago. This permits enhanced mobility management, e.g., through the enforcement of appropriate speed limits depending on the usage and congestion status of a street.

Similarly, commercial and municipal parking site or multi-storey car park operators have installed occupancy sensors for some time now, which permits the enhancement of the routing of vehicles, but also sophisticated steering mechanisms such as flexible pricing.

Sensor data can be interpreted as a complement to user-centric data, because it provides real-time information on the state of the overall transportation system. Integrated and combined with user-centric data in an appropriate way, this provides useful information on real-time conditions in cities.

4.1.2 Open Data

As it has been briefly explained before, open data refers to information that has been released by an organisation, most likely a governmental institution, to the general public. As of January 2013, the English version of Wikipedia[5] lists more than 25 initiatives for open data by national governments, predominantly from industrialized countries, but also from developing nations such as India or Kenya.

An overview of selected, mobility-related data sources that may be of interest within the SIMPLI-CITY project is given in Table 1 in the Annex of this document. The table lists the geographical locations of the data sources and their type, and also provides a brief description and a URL pointing to additional information.

The specific source of open data depends on its type; in some cases, it may consist of real-time information that has been acquired by sensors, such as parking information, thus giving an overlap with the sensor data category. In other cases, such as information about bike routes, the information may be much more static in nature and has been acquired by different technical means.

4.1.3 User-centric Data

User-centric data is information that, in some aspect, describes the context of a road user, and it may stem from many sources. In the context of SIMPLI-CITY, cellular phones, or more generally connected nomadic devices and in-vehicle systems are probably the most relevant, but Web-based sources like an online calendar could also be relevant.

In the first decade of the 21st century, cellular phones have advanced from simple means of voice communication to almost versatile personal communication and information devices, e.g., smartphones and connected tablets. They contain a multitude of information on the respective user, such as calendar data, contacts, and emails. This information can be combined and used to provide enhanced mobility, e.g., by identifying regular travel patterns and personal requirements and preferences.

Similarly, in recent years, car manufactures have started to install increasingly sophisticated sensing technology into their vehicles – not only in the luxury class, but also in low-cost vehicles. These sensors facilitate both security and convenience features, such

[5] http://en.wikipedia.org/wiki/Open_Data

as automatic activation of headlights and windshield wipers, or emergency brakes in the case of insufficient distance. They also provide a multitude of information on the vehicle status, which includes information such as fuel level, current fuel consumption, or average speed.

Given that a major goal of SIMPLI-CITY consists in enhancing the mobility experience of its users, user-centric data plays a crucial role. It facilitates the adaptation of the overall system and individual applications to specific needs of every user, thus complementing the two previously, more general information categories of sensor data and open data.

4.1.4 Media Streams

Media streams are commonly comprised of an audio or audio/video content that is delivered to the user in real-time or near real-time. In the context of mobility, they usually serve leisure or convenience purposes, i.e., provide entertainment to the user. Popular examples of media stream providers are, for example, Deezer, Last.FM, or SoundCloud. Naturally, in SIMPLI-CITY the primary type of media streams will be audio streams, as video streams would potentially distract the driver too much. In some cases, media streams may also serve certain security or informational purposes. For example, information about road conditions may be broadcasted in the form of an audio stream, comparable to traditional radio broadcasts.

Media streams are characterized by comparatively high demands on the network, because they usually involve substantial amounts of data and have relatively strict requirements with respect to latency and jitter. As such, a support of media streams within SIMPLI-CITY needs to provide approaches to handle instable or slow connections by incorporating caching or prefetching techniques. This may include pure data streaming such as streaming music, but it may also include intelligent invocation of content-related services for prefetching data or requesting information in advance. Thus, SIMPLI-CITY will facilitate uninterrupted media playback.

4.1.5 Holistic Privacy and Security Concept

SIMPLI-CITY is dealing with sensor data which can be very intimate data, e.g., the GPS position of the user, and user-centric data, which can include appointments from the calendar of a user. Regarding the amount of personal data SIMPLI-CITY is dealing with, it is necessary to guarantee data privacy in order to ensure data is protected against malicious parties. This will be achieved by putting security mechanisms into place.

In SIMPLI-CITY, privacy-relevant data will be classified as such and then be protected by applying the RBAC concept. This is a concept in which roles are defined with individual access rights. Each user acts then under the access rights of a particular role. In addition, SIMPLI-CITY will make use of the concept of anonymisation and pseudonymisation of data. This means that the user is represented by a key and her data, e.g., sensor data, is then stored anonymously. The owner of this data can only be found out by this key.

On the PMA side, apps will be secured with grating them specific access rights. When installing an app, users will be able to see which access rights this app requires. This will allow users to block apps that make use of, e.g., personal media streams. Also, SIMPLI-CITY will offer adequate security, privacy, and control measures. Furthermore, all data transformation will be fully encrypted and will use HTTPS-based communication, and user information will only be transferred if necessary, e.g., for making a payment for a parking reservation service.

4.2 SIMPLI-CITY Software Components

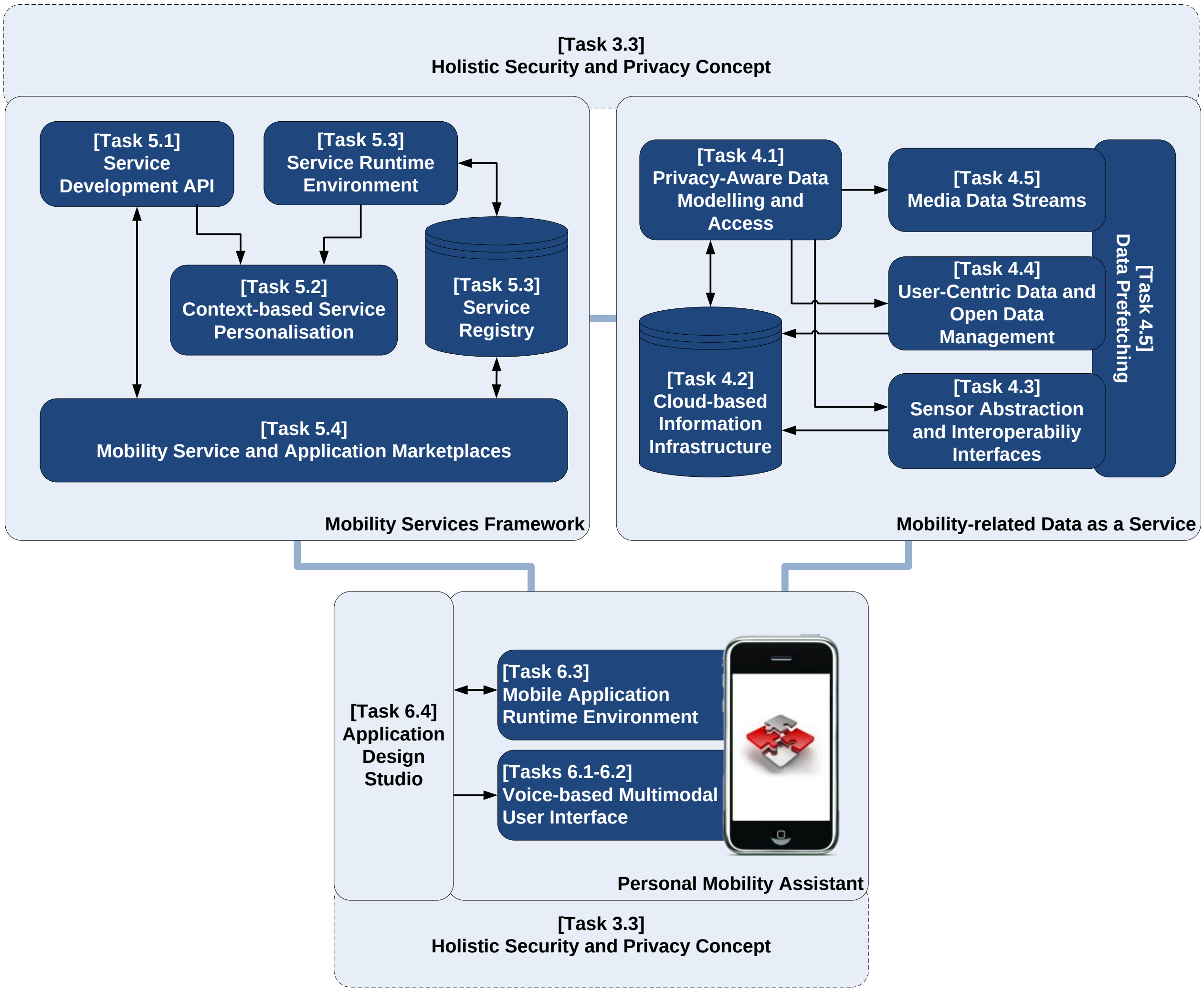

Figure 1: Overview of (Software) Components within the SIMPLI-CITY System

As can be observed from Figure 1, the SIMPLI-CITY system is comprised of various components, which exhibit a substantial amount of bi- and multilateral interaction. Specifically, three major groups of components are identified:

- *Mobility-related Data as a Service*, which deals with the collection, aggregation, consolidation, storage, and retrieval of data. A key component is the Cloud-based Information Infrastructure, which serves as highly scalable and flexible database. The corresponding components are developed in WP4 and presented in Sections 4.2.1 to 4.2.5.
- *Mobility Services Framework*, which facilitates development, publication, and execution of services and mobile apps. It targets both service consumers and developers and also provides intermediation between these two stakeholders. The constituent software components are implemented in WP5 and presented in Sections 4.2.6 to 4.2.9.
- *Personal Mobility Assistant*, which is the primary interface to SIMPLI-CITY for the end user, but also encompasses a mobile runtime environment. This group of components is targeted in WP6 and presented in Sections 4.2.10 to 4.2.12.

The individual components are explained in detail in the following section, ordered by the tasks in which they will be implemented within the SIMPLI-CITY project.

4.2.1 Privacy-Aware Data Modelling and Access Framework (Task T4.1)

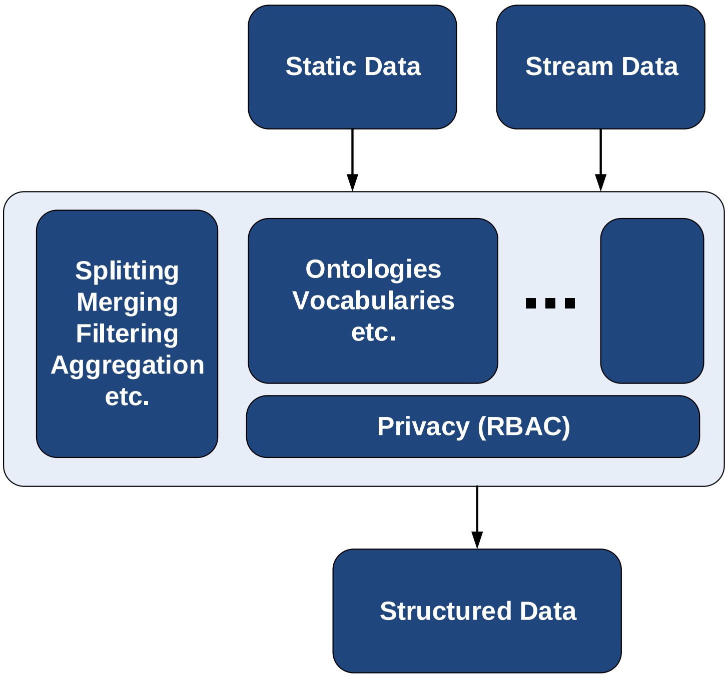

Figure 2: Privacy-Aware Data Modelling and Access Framework

Motivation: Data from the SIMPLI-CITY domain is defined by its heterogeneity in terms of formats. The latter makes data integration (e.g., combination, filtering) a very difficult task to achieve. In addition, mobility-aware data could be privacy-sensitive, making its access even more problematic. Since the data models of heterogeneous sources cannot be easily integrated, a common data representation model is required. The latter will be the basis for supporting advanced data correlation, combination and linkage.

Purpose: 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, all ensuring privacy protection of the most sensitive data and easy access to data following principles of the Cloud-based Information Infrastructure (T4.2). This task aims at providing a semantic reference data model, primarily for spatial, temporal, and behavioural-based representation of data.

Input(s): Static and stream data from diverse sources.

Output(s): Data in structured, standardized formats and privacy-protected if required. Data will be also contextualized i.e., cross-correlated if possible (e.g., spatio-temporal correlation is one basic correlation method). The output of this task will consist on a core semantic representation layer for achieving the MDaaS principle.

Description: This component (see Figure 2) provides a semantic layer to raw data and ensures that data can be integrated using a unified and standardized format. It also provides basic functionalities for semantic combination (e.g., splitting, merging, filtering, and aggregation) in a context of heterogeneous data (both format and up-to-dateness). Semantic representation is considered as a key to bring data into a common

representation. Utilities for cross- and multi-stream-correlation along static data are also part of this task. Since space and time are both important features in city- and mobility-related data, SIMPLI-CITY will strongly focus on the spatiotemporal representation of the data model. In case of sensitive data, privacy-protection techniques are used to make sure sensitive information cannot be leaked. Different levels of abstractions will be provided, with different roles assigned to them, e.g., based on an RBAC system where each role is provided with access to data that is protected to the level necessary for the protection role. Regarding the privacy dimension, a combination of RBAC with data anonymisation strategies, all together with semantic representation of roles/levels of abstractions, is proposed. This will ensure a privacy-aware access to data.

Interactions and Dependencies: T4.1 provides (i) a description layer together with (ii) basic functionalities for data integration (e.g., splitting, merging, filtering, and aggregation) for tasks T4.2, T4.3 and T4.4. The central storage of data and its representation – which is acquired through this component – is facilitated by the SIMPLI-CITY Cloud-based Information Infrastructure (T4.2). While all data acquired and processed within SIMPLI-CITY is protected against data abuse by measures such as user login and encrypted data transfer, the mechanisms in place to ensure privacy of sensitive user-centric data are especially strict, including pre-processing steps like anonymisation or pseudonymisation. For this, the Holistic Security and Privacy Concept (T3.3) will be applied.

4.2.2 Cloud-based Information Infrastructure (Task T4.2)

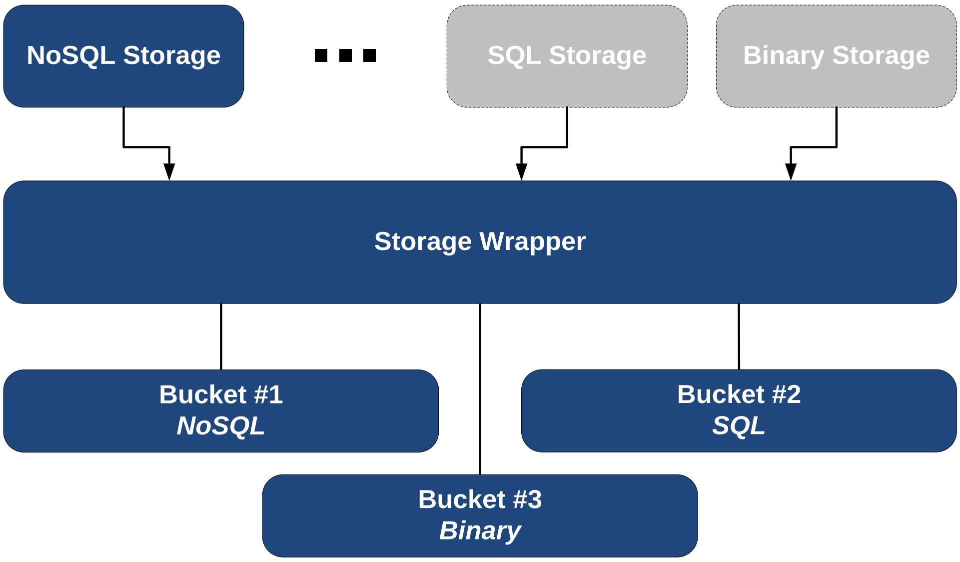

Figure 3: Cloud-based Information Infrastructure – Isolated Buckets

Motivation: Within the SIMPLI-CITY use cases, a large amount of data is consumed by services and apps. This data may be live data, e.g., extracted from sensors using the sensor abstraction framework, or it may be persistent data, which is stored and later received. As such, a component for managing persistent data is needed by SIMPLI-CITY.

Purpose: Access, storage, and retrieval of mobility-related data are performed within the Cloud-based Information Infrastructure. The component will act as a data storage solution for information and serves as a source for static data that can be used by apps and services.

Input(s): Structured data sets, e.g., as JavaScript Object Notation (JSON)-based data. Alternatively, queries may be issued to receive information which is stored in the Cloud. Also, API calls are supported to, e.g., delete data from the storage.

Output(s): Query results are delivered in a structured data format which acts as an envelope for delivery. The actual result information will be exchanged as payload data and will be delivered within different formats depending on the bucket type of the data storage.

Description: Data managed by this component may be classified as two different types:

- Data which is up-to-date, such as information stored by sensors or data about specific events. This data is referred to as “real-time” information, e.g., a temperature value for a specific sensor. This data will need to be accessible in a very fast way.
- Data which is archived and covers historical information. This may include historic temperature values or older traffic information. This data may be used to allow services and apps to query information from the past, e.g., in order to “learn” from former traffic data. Since the amount of data may be very large, the archive does focus on the scalability as first priority and speed as second. As such, queries on this information source may be significantly slower than queries on the real-time data storage.

Data may be originating from all types of data sources, most prominently information coming from sensor sources including intelligent infrastructures and cooperative systems, as well as people-centric sensing, e.g., data coming from vehicle-internal sensors (telematics) and mobile end user devices that can be directly related to a particular person. Furthermore, data coming from open data sources and other social and sensor data sources will be storable. The Information Infrastructure is fully content-agnostic and as such, it does not know what the semantics of data are. It may therefore be compared to a database for managing information in a fast, scalable and reliable way.

The SIMPLI-CITY Cloud-based Information Infrastructure harmonises access to the data sources, which will be defined in tasks T4.3 and T4.4, by providing service developers with unified methods to access data by using MDaaS mechanisms. The component will provide an interface which will allow other components, services and applications to store, modify, and delete information (CRUD) and it will also provide a query mechanism for requesting data. Information may be stored in a NoSQL approach and may provide data storage and a query interface based on JSON.

Data will be managed in the Cloud, which will provide an elastic way of storing information. For components, apps and services, this fact will be hidden by the unified API mentioned above. The storage itself will, however, be based on a distributable and scalable technology in order to prevent bottlenecks in terms of speed. The detailed technology will be subject to the functional and technical specifications of SIMPLI-CITY.

One particular aspect of data access and data storage is the identification of *relevant* and *valid* data. For this, SIMPLI-CITY will develop mechanisms to estimate relevance of data changes and events regarding mobility, environmental impact, safety, and comfort. Only the data deemed to be relevant will be stored within the Information Infrastructure. The control over this will be done by the apps, components and services using the storage. They will be able to define rules allowing the storage component to decide how long data is kept in the real-time data storage and when data is either deleted or archived. For example, a rule may define that temperature information is stored in the real-time data

storage for two days for each minute of the day and that the data should afterwards be moved to the historic data archive with one measurement per hour allowing the storage to delete the other data.

The component will allow services, apps and components to create isolated data storage spaces which will not overlap with those of other components, services or apps. Those isolated storage spaces are referred to as “Buckets” (see Figure 3), which is a concept originating from the Amazon S3 storage solution and has proved to be robust in many Cloud-based storage solutions. The bucket concept allows the usage of different storage backends in order to support different types of data storage.

Attention will be paid to data privacy by putting adequate security mechanisms into place, in order to ensure that data is protected against malicious parties. This is performed by allowing the data buckets to define their visibility using visibility tokens. Those tokens may either open a bucket to public read, public read-write, or restrict the data of the bucket to one specific component, user, app, or service.

Real-time data provisioning is facilitated through pulling as well as pushing mechanisms. This means that data may either be queried using the API of the storage solution or that data may be subscribed to, in which case the storage will inform apps, components, or services about the arrival of data in a new bucket.

Interactions and Dependencies: There is no direct dependency of this component on another component, with the exception of the Holistic Security and Privacy Concept (T3.3) and the SIMPLI-CITY Privacy-Aware Data Modelling and Access Framework (T4.1). The component will allow a content-agnostic interaction with components, services or apps (directly or indirectly) and will as such provide an API which is usable by all SIMPLI-CITY tasks.

4.2.3 Sensor Abstraction and Interoperability Interfaces (Task T4.3)

Motivation: One important data source in the SIMPLI-CITY use case scenarios is data originating from different types of sensors. Thus, these sensor data provide an essential input to the MDaaS approach envisioned to be realized by SIMPLI-CITY. However, current sensor technology is characterized by a high degree of heterogeneity, as is reflected in different hardware platforms employed, different interfaces, different communication protocols, and varying data formats. This complicates access to sensor data and poses substantial overhead to application developers, because specific implementations are required for each regarded sensor type. These problems are addressed through the SIMPLI-CITY Sensor Abstraction and Interoperability Interfaces.

Purpose: Provide the means to integrate real-time data coming from individual sensors, sensor networks, smart objects, telematics, and cooperative systems, using various wired and wireless communication technologies for the usage in SIMPLI-CITY MDaaS.

Input(s): Predominantly real-time, raw sensor data from diverse sources, in various formats and differing levels of abstraction, e.g., triggered by certain local events or requested by corresponding function calls.

Output(s): Consolidated, possibly filtered sensor data in structured, standardized formats.

Description: This component (see Figure 4) provides an intermediate layer, i.e., middleware, between various real-time data capturing systems, and other components that require access to the corresponding data. Car sensors as well as other sensors will be

supported – however, the middleware may be logically divided into different software components in order to meet the requirements from these sensor classes. The component implements a common interface, i.e., set of well-defined methods, which permit the acquisition of data from technically heterogeneous sensor systems in a structured, standardized form. In order to facilitate maximal flexibility and extensibility, this component will employ the concept of wrappers. These wrappers translate method calls to the generic interface into specific function calls for the respective sensor system. Likewise, the wrappers perform translation of raw data from specific sensor systems into harmonized, generic output format. This middleware will also allow directly accessing aggregated and composed data. In the context of SIMPLI-CITY and specifically this task, a set of sensors will be identified for which wrappers will be implemented. In addition, this component will provide architectural guidelines for the development of new wrappers in the form of abstract classes or interfaces.

Interactions and Dependencies: The technical representation of data and the privacy-aware access control in this component is governed by the SIMPLI-CITY Holistic Security and Privacy Concept (T3.3) and the SIMPLI-CITY Privacy-Aware Data Modelling and Access Framework (T4.1). The central storage of (historical) data originating from the different sensor sources and transferred via the component described here is facilitated by the SIMPLI-CITY Cloud-based Information Infrastructure (T4.2).

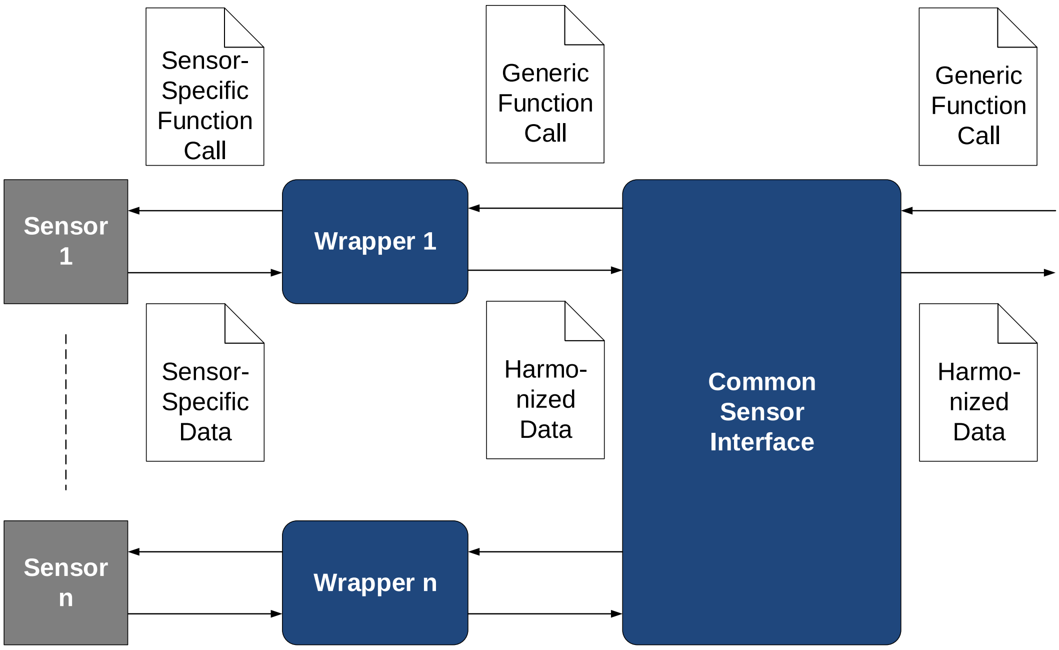

Figure 4: Sensor Abstraction and Interoperability Interfaces

4.2.4 User-Centric Data and Open Data Management (Task T4.4)

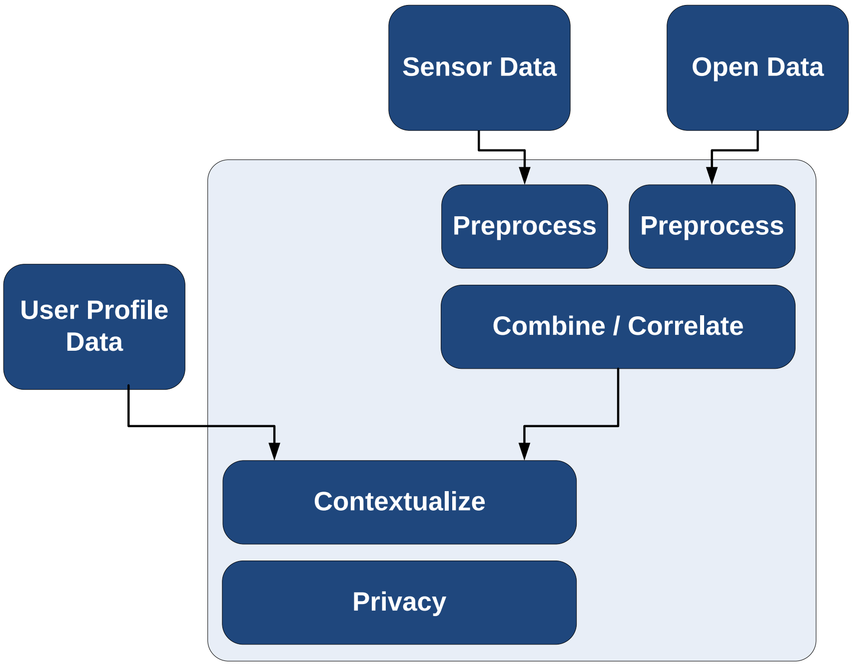

Figure 5: User-Centric Data and Open Data Management

Motivation: Context data is defined as all information that can be used to describe the status of an entity. An entity might be a person, location or object, which is deemed relevant with regard to the interaction between the end user and an app. User-related information and environmental context (e.g., spatial and temporal) are important for contextualizing services, e.g., the selection of a car park based on the user's location, or the selection of a payment method based on the user's preferences.

Purpose: This task will aim at making data context-aware, i.e., (i) how to find relevant data set for interpreting sensor data from T4.3 (e.g., in the simple case of temperature sensor reading, context could be related to its GPS location, distance to other sensors, wind speed in this GPS location), (ii) which open data can be used for contextualizing relevant information from heterogeneous data (e.g., in the case of a private loop detector, some public bodies make available the number of incidents within a few kilometres, making the correlation of the latter sets of stream data very useful). It is very difficult to retrieve relevant information for specific users in some contexts (and even find relevant open data and sensor data within a specific context), or at least require a large amount of pre-processing steps. Task T4.4 aims at integrating and correlating relevant information from sensors and open (government) data with user-centric data. All these elements together will help to achieve the MDaaS principle.

The main purpose of T4.4 consists in personalizing city and mobility information for end users by coupling and correlating information from (i) sensors, (ii) personal user data, and (iii) open and public information. The correlation will be mainly achieved based on spatial and temporal representations as well as on the user profile level.

Input(s): Relevant sensor data from T4.3, together with user profiles and relevant open and public (government) data. All previous data is stored following principles of T4.2, and described following principles of T4.1.

Output(s): Basic personalized and accurate information inferred from combinations of relevant sensor data from T4.3, together with user profiles and relevant open and public government data.

Description: This component (see Figure 5) provides mechanisms for correlating and combining data (and its descriptions) coming from various heterogeneous domains, mainly user centric data (e.g., personal calendar data, user needs and preferences) together with open government data (which could be sensor-based data), e.g., real-time or historical data from city traffic management systems. In particular, the semantic data model (T4.1) will be used for representing and connecting relevant data within a specific context. Basic elements of context identification and materialization (e.g., through personalization) will be delivered. To this end, the spatiotemporal representation of data (derived from T4.1) will be coupled with user-centric data together with open data. As end-user data will be required for contextualization purpose, its access is crucial.

The User-Centric Data Management also aims at storing correlation and potential combinations in a re-usable format, which is easy to be queried. Mainly, semantic standards like RDF, RDF-S, and OWL may be used as data formats.

Interactions and Dependencies: This component depends on the SIMPLI-CITY data model from T4.1 and makes use of the Cloud-based Information Infrastructure (T4.2) to store relevant data. Furthermore, the SIMPLI-CITY Holistic Security and Privacy Concept (T3.3) will be integrated in order to achieve data privacy. Basic elements of contextualization will be exposed through T5.2 and T5.3, and executable through T6.3.

4.2.5 Media Data Streams and Data Prefetching (Task T4.5)

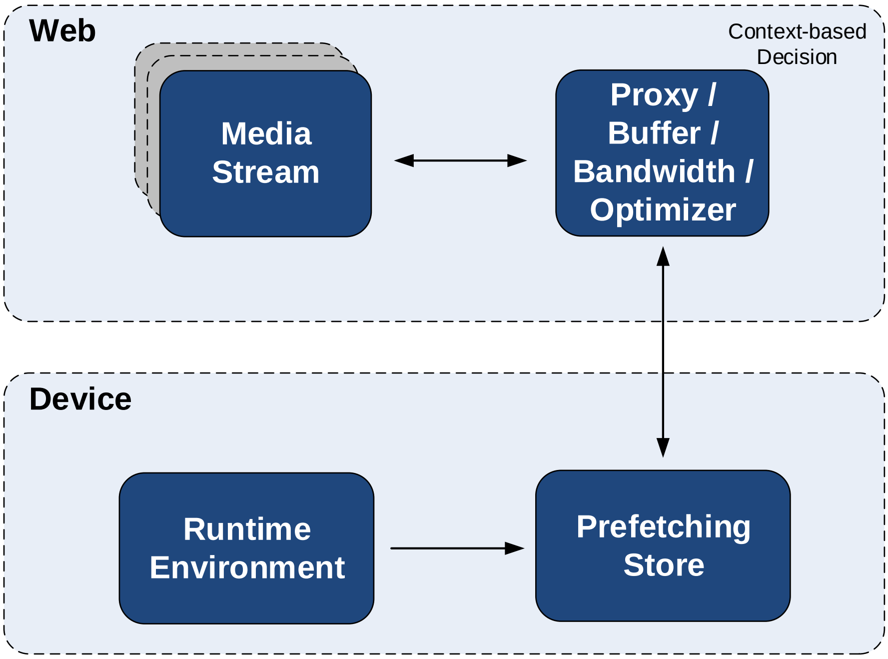

Figure 6: Media Data Streams and Data Prefetching

Motivation: The usage of the PMA for consuming services and data via apps is an essential element of SIMPLI-CITY. Two principal open issues in this domain are connectivity problems and bandwidth fluctuations. The observance of this fact is needed in SIMPLI-CITY in order to cater for seamless media playback and service consumption.

Data prefetching is an adequate way to overcome these problems and to reach an uninterrupted experience with the PMA.

Purpose: Apart from integrating media streams, this component increases availability of media streams and other data sources (including services) by introducing relevant data prefetching techniques.

Input(s): Request of media streams or prefetchable data from the PMA. Optionally: Parameters connected to the current context of the user (e.g., the GPS location).

Output(s): Regarding media streaming, the output of this component would be a media stream; regarding other data types, the output is simply a response to the request. This output would then be already prefetched (see Figure 6) on the device or it would be transported with help of the most optimal communication technologies and techniques in the current context. In addition to prefetching data for media streaming, the component will also support the prefetching of data by invoking services automatically based on the current context. In this case, the user may receive a notification about potential data that might be of interest for her/him.

Description: This component will support streaming of media information to apps. This may be used to create apps that support the playback of music or other media information. By nature, the consumption of media is sensitive to interruptions: Even a connectivity loss of a few seconds will give the user a bad experience if it happens in the middle of a song that the user is listening to. For this purpose, the component will integrate a media buffering solution by prefetching relevant data in a local buffer.

As data prefetching can lead to faster battery depletion of the mobile device, SIMPLI-CITY may develop new, context-based data prefetching mechanisms. This means that context data about, e.g., the user's location and destination would be used in order to find a suitable degree of data prefetching. If context information indicates that the user will soon be in an area with poor cellular coverage, the data prefetching mechanisms would retrieve a relatively large amount of media data in order to guarantee playback. The prefetching mechanisms would then be able to distinguish between a short term (e.g., the user drives through a tunnel) or long term need for prefetching (e.g., the area with poor coverage is relatively large or the user wants to download as much as possible from a WIFI- instead of a 3G-based data connection).

Context data including usage statistics might be stored in the Cloud-based Information Infrastructure in order to provide users with enhanced experience. However, the actual data streams will *not* be stored within the Cloud-based Information Infrastructure, although the Cloud-based Information Infrastructure may manage information about where and how streams may be accessed.

As shown in Figure 6, the prefetching of data will be divided into two parts for media data: A server side component will be used to optimize the media stream based on the current connection quality and the bandwidth. For example, it may deliver music with a lower bitrate when detecting that the user is using a slow connection and with a good quality when the user is using, e.g., LTE. In addition to this, the device part of this component will provide a local storage for buffering the prefetched data.

Media data streams are without a doubt the most obvious kind of data where prefetching may be applied. However, it might also make sense to prefetch non-stream data in case of an area with weak network coverage, for example, to get information about petrol stations beforehand, based on the notion that the driver needs to refuel the car shortly. This

component will therefore prefetch the corresponding information and keep it in a local buffer for the time that it is requested by an app. Obviously this data prefetching is not possible in all cases as some data might be time-critical and cannot be buffered for a long time. For example, parking information about free parking lots may be outdated after 20 minutes. Therefore, the component will support the expiration of data, which means that prefetched data will expire when it becomes too old. For supporting this, an app may mark services which it invokes as “prefetchable” data services and it may specify an “expired-after” flag to mark the data as outdated after a certain amount of time.

As described above, the component will allow both media prefetching (e.g., for music streaming) and service prefetching. Service prefetching will invoke services which the user is likely to use in the next time (see Figure 7). This is performed by monitoring the user’s behaviour and combining it with historical behaviour of SIMPLI-CITY users. The service selection will be made on the device, based on local user settings and local sensor data, and on the server side, based on historical behaviour of SIMPLI-CITY users. The results of service invocations will be buffered in a local prefetching storage.

Interactions and Dependencies: The actual usage of the data prefetching mechanisms will be regarded within task T5.2 (Context-based Service Personalisation). Additionally, the task may make use of the local storage facilities of the Application Runtime Environment of task T6.3. Furthermore, user interaction may be needed for informing the user about prefetched service data. This will be performed in conjunction with tasks T6.1 and T6.2.

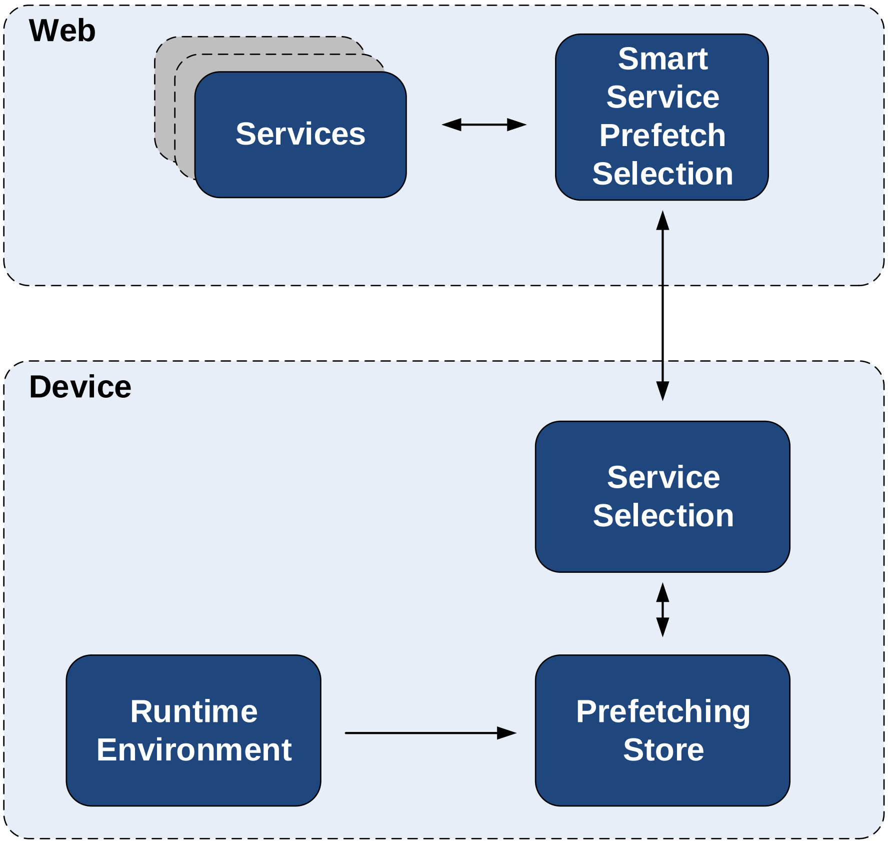

Figure 7: Service Prefetching

4.2.6 Service Development API (Task T5.1)

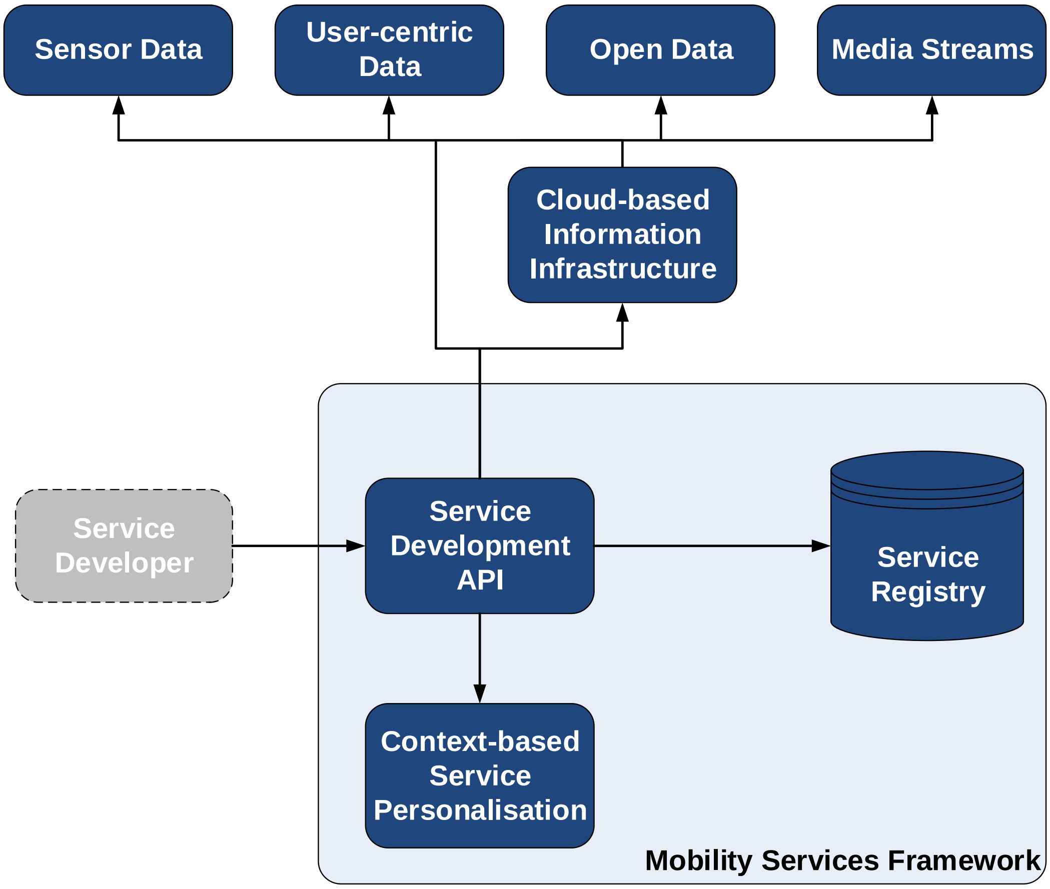

Figure 8: Service Development API

Motivation: A component is needed in order to provide service developers with the means to create services and integrate data from various sources. These services may have arbitrary functionalities and may be able to integrate data from different MDaaS-enabled data sources. These (backend) services are the ones that will be consumed by the end user apps running in the PMA.

Purpose: To develop an API that provides functionalities to easily define, design, develop, and orchestrate mobility-related services, and functionalities for accessing the SIMPLI-CITY server side APIs and components. The API also contains the functionality to manually combine services in order to provide value-added service compositions.

Input(s): Data from various real-time data sources, including cooperative systems, telematics, intelligent infrastructures, sensors and sensor networks, media data streams, open data repositories, or people-centric sensing.

Output(s): Services and service compositions.

Description: The Service Development API provides developers with the necessary tools to create their services on the SIMPLI-CITY platform.

The Service Development API assists developers during the entire process of service creation. It permits to integrate into services the different data sources provided by WP4 components following the MDaaS approach. It also allows services to make use of context

data provided by the Context-based Service Personalisation Component, as illustrated in Figure 8.

The Service Development API offers the methods needed to access the functionalities of all technical components that a service developer may need, including the easy integration of data from the Cloud-based Information Infrastructure, the adaptation of a service's outcome based on context data, and the storage of service-specific data within the Cloud.

The data used by services can be retrieved later in order to adapt service delivery based on past user interactions and user decisions. In SIMPLI-CITY, (sensor) data sources will not necessarily be directly connected to backend services. Instead, backend services may be linked to a class of sensors, allowing getting data from sensor instances during runtime. One example would be that an app makes use of the abstract data class "Parking Spaces", which provides information about (free) parking spaces in a particular area. During app execution and at service runtime, a concrete data source/service that implements this abstract data class will be selected, e.g., providing parking spaces for a particular town. This way, it is possible to customize apps and services to different locations or implement other adaptations.

The Service Development API will be implemented as a Web-based tool providing documentation, examples and interface descriptions in order to help developers during the service creation process. The services created will be available to other developers through the Mobility Service Marketplace, allowing the reutilisation of services.

Interactions and Dependencies: This component interacts with the Cloud-based Information Infrastructure (T4.2) in order to get access to different types of data sources provided by the Sensor Abstraction and Interoperability Interfaces (T4.3), the User-Centric Data and Open Data Management (T4.4), and the Media Data Streams and Data Prefetching component (T4.5). Moreover, it interacts with the Context-based Service Personalisation component (T5.2), in order to integrate context data in the service's outcome. Services may run in the Service Runtime Environment (T5.3), may be stored in the Service Registry (T5.3), and can be offered on the Mobility Service Marketplace (T5.4).

4.2.7 Context-based Service Personalisation (Task T5.2)

Motivation: Especially in an end user-driven domain like mobility, the output of a service or app should be adapted to the personal preferences of the user, even if these are not explicitly provided as input. Examples for this are, e.g., personal transport mode preferences in an according recommender service/app, inclusion of local data sources based on the user's location, or the decision to carry out data prefetching based on the route the user has taken, the (likely) network coverage during travel, or his/her mobile phone contract, which indicates if it is better to download data within the home network or on the road. In these examples, it should not be necessary to state information such as "I'd prefer to go by car and will by no means take a public bus", "I will be in London, so please make use of the local Open Government Data source about London's accident statistics", "I have already made use of 90% of my monthly mobile data bandwidth cap, so please download my music while I am at home" or "I will be out of country and I want to use data roaming at a minimum". Instead, such context data should be automatically retrieved and taken into account. For this, the SIMPLI-CITY Context-based Service Personalisation software component will provide the necessary functionality.

Purpose: Provide the means to alter the outcome (output) of a service invocation based on relevant (user) context data.

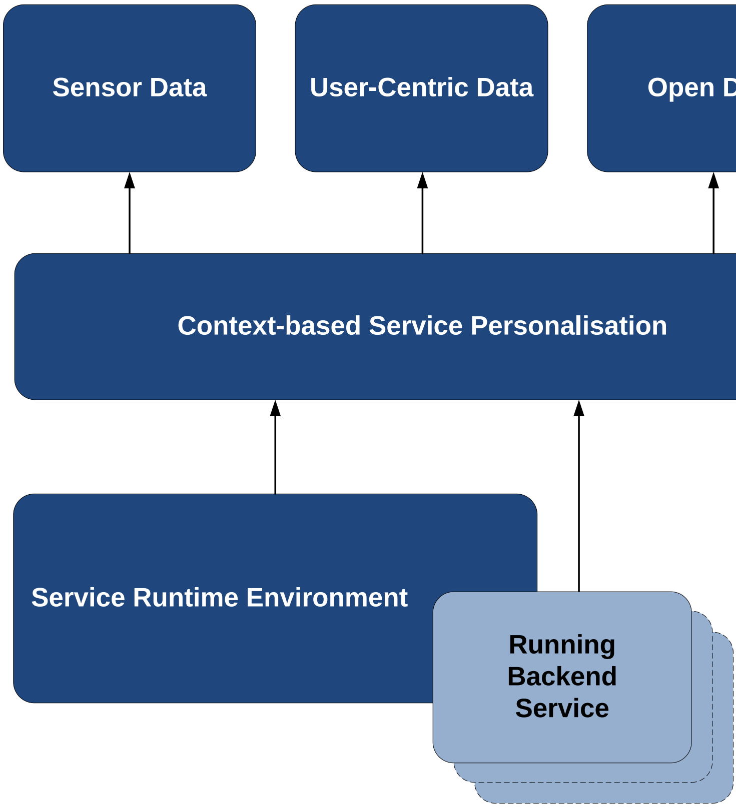

Figure 9: Context-based Service Personalisation

Input(s): Context data, e.g., user profile including the user's location; from local or global sensors; information about different (context-based) variants of the same service; information about service invocations. The component will partly get the according context data when being invoked, but will also retrieve such data from eligible sources.

Output(s): Based on the kind of personalisation, there are different possibilities for the output of this component: (i) Data that will be used as input for another service (e.g., payment details), (ii) Control statement for a service (e.g., bring forward data prefetching), (iii) A particular service instance/variant which fulfils a functionality for a particular context (e.g., some local functionality/information).

Description: As depicted in Figure 9, the Context-based Service Personalisation component is a helper component that supports single services to make use of the correct data in a particular situation. It chooses a service instance which provides a particular functionality with regard to a particular context (in most cases a different location), or provides decision support to a service instance. Thus, it can be invoked both by single, running backend services, or the Service Runtime Environment. Regarding the decision support functionality for service instances, this component will provide different logics that are capable to identify in which context situations a personalisation should be applied and to which extent. Such logic can then be used by relevant services.

Notably, the Context-based Service Personalisation component does not provide data access or service execution itself, but provides either information about which data source or service to invoke. In any case, the to-be-adapted service needs to implement a context model or provide information about the desired functionalities so that a particular data service can be chosen.

Interactions and Dependencies: As it can be seen in Figure 9, this component is invoked by the SIMPLI-CITY Service Runtime Environment (T5.3), i.e., it is a helper component. It interacts with the three components providing access to different types of data sources, the SIMPLI-CITY Sensor Abstraction and Interoperability Interfaces (T4.3), the SIMPLI-CITY User-Centric Data and Open Data Management (T4.4), and the SIMPLI-CITY Media Data Streams and Data Prefetching component (T4.5); all these components may deliver context data. T3.3 (SIMPLI-CITY Holistic Security and Privacy Concept), T4.1 (Privacy-Aware Data Modelling and Access Framework), and T4.2 (SIMPLI-CITY Cloud-based Information Infrastructure) may also be indirectly concerned through T4.3-T4.5 but are not depicted in Figure 9 in order to keep the figure well-arranged.

4.2.8 Service Registry and Service Runtime Environment (Task T5.3)

Motivation: The SIMPLI-CITY Service Registry and Service Runtime Environment provide necessary backend functionality to execute services. Notably, SIMPLI-CITY differentiates between services and end user apps. While the latter provide some local functionality running on a mobile device, the services are providing backend functionality. Services are used to facilitate reuse and shifting resource- or data-intensive tasks from a mobile device to a more powerful server counterpart. Furthermore, it is easier to keep a fixed connection between data sources and the backend than between data sources and a mobile, probably moving device.

Purpose: Provision of functionality necessary to register, execute, bind, and monitor backend services.

Input(s): For the Service Registry: descriptions or deployable service implementations; for the Service Runtime Environment: Service requests (either issued by end user apps or other services).

Output(s): For the Service Registry: Service endpoint and information on how to invoke a service; for the Service Runtime Environment, the output is provided by the invoked services; the output of the Service Runtime Environment is therefore an actual service invocation.

Description: An overview of the Service Runtime Environment and the Service Registry is depicted in Figure 10. As can be seen, the SIMPLI-CITY Service Registry offers functionality to register services, select, and invoke them. Service Level Agreements (SLAs) are also supported and are the foundation to monitor and guarantee non-functional service aspects such as response time. The concept of registering services is closely related to offering services on the Service Marketplace, but not every service provider will necessarily want to offer their services on the marketplace.

After a service has been set up using the Service Development API and registered within the Service Registry, it can be executed using the Service Runtime Environment. For this, the component allows service selection and enables the service execution (binding) as well as the detection of SLA violations.

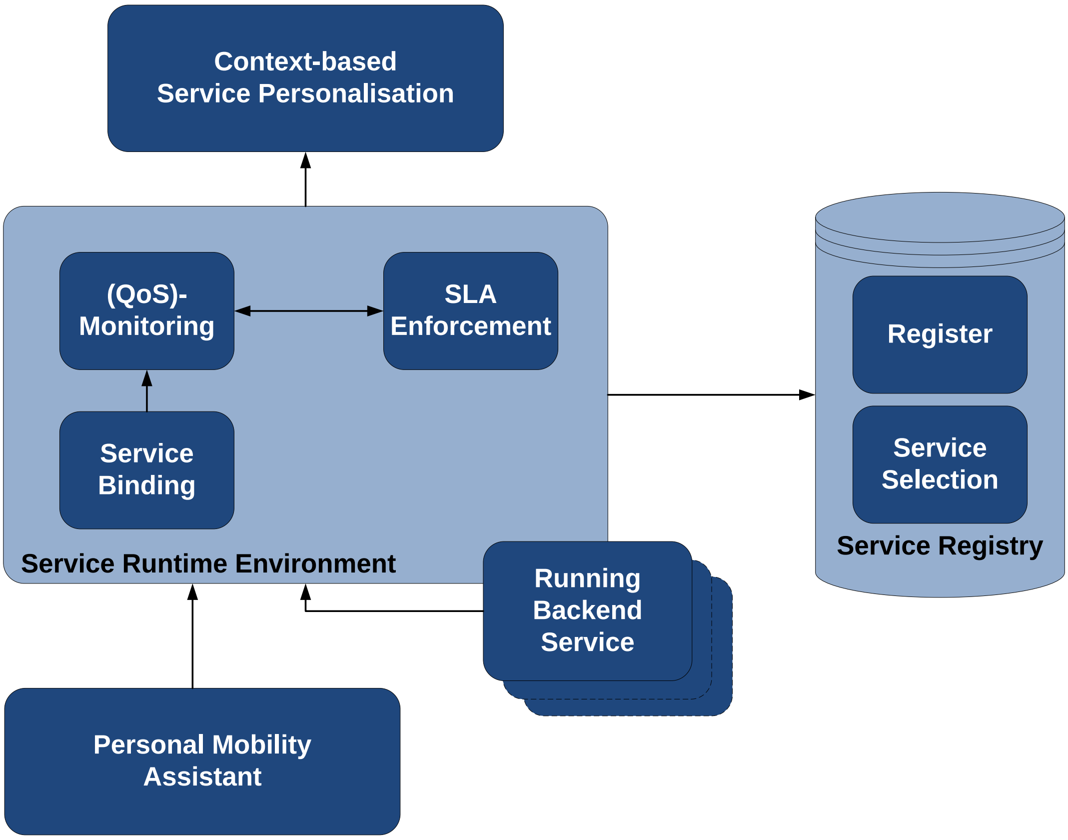

Figure 10: Service Runtime Environment and Service Registry

The Service Runtime Environment executes services based on according requests by users or other services, and therefore provides a service middleware. In many cases, service requests will be issued by the PMA, processed in the Service Runtime Environment and the response to a request will also be routed back through this middleware. As described before, the PMA is prone to connectivity issues including connection losses. Hence, the Service Runtime Environment needs to make sure that responses are delivered even if there are connectivity issues. Furthermore, service execution includes binding to concrete data sources based on the sensor classes defined in service development (see Section 4.2.6) and personalisation of the service's output based on context data (in cooperation with the SIMPLI-CITY Context-based Service Personalisation). The means to handle user (input) data coming from the PMA will also be included.

Interactions and Dependencies: Services are primarily invoked through apps running in the PMA (T6.3), hence, there is a direct connection between these two components; furthermore, services may themselves invoke other services. The Runtime Environment relies on T5.2 for the binding to concrete data sources and personalisation of service functionalities/outcomes. For this, the components interact with the components provided in T4.3-T4.5 (see interactions and dependencies of the SIMPLI-CITY Context-based Service Personalisation). Last but not least, the Service Registry provides basic technologies for the implementation of the Mobility Service and Application Marketplaces (T5.4).

4.2.9 Mobility Service and Application Marketplaces (Task T5.4)

Motivation: One of the main advantages of SIMPLI-CITY compared to existing road user information systems is the facilitated flexibility and the possibility to add new apps and services. This approach is inspired by the broad success of mobile apps, which offer users a wide range of new functionalities, showcasing the innovative power and ideas of thousands of developers.

Purpose: The purpose of this task is the provision of two marketplaces for services and apps which are usable with SIMPLI-CITY.

Input(s): Apps and services and their descriptions and their manifests (created via the Service Development API and Application Design Studio, respectively).

Output(s): The marketplace provides a user interface allowing end users to select and install apps on their PMA and allowing service developers to discover and select services. It also provides feedback to developers in form of comments and ratings.

Description: SIMPLI-CITY will learn from the app marketplaces approach as provided by Apple, Google, or Microsoft and transfer it to the mobility sector, based on the data integration and service framework that will be created within the project. In fact, the SIMPLI-CITY Mobility Service and Application Marketplaces take the original idea of app marketplaces even further, as not only end user apps will be provided which can be easily downloaded and used on mobile devices, but also services which can be used within apps or other services.

Furthermore, the architecture of the SIMPLI-CITY marketplaces allows their integration in other Web-based platforms, e.g., in order to provide marketplaces where a specific city can offer all their location-specific services and apps to their local residents, or integration of services from other service platforms into the SIMPLI-CITY marketplaces.

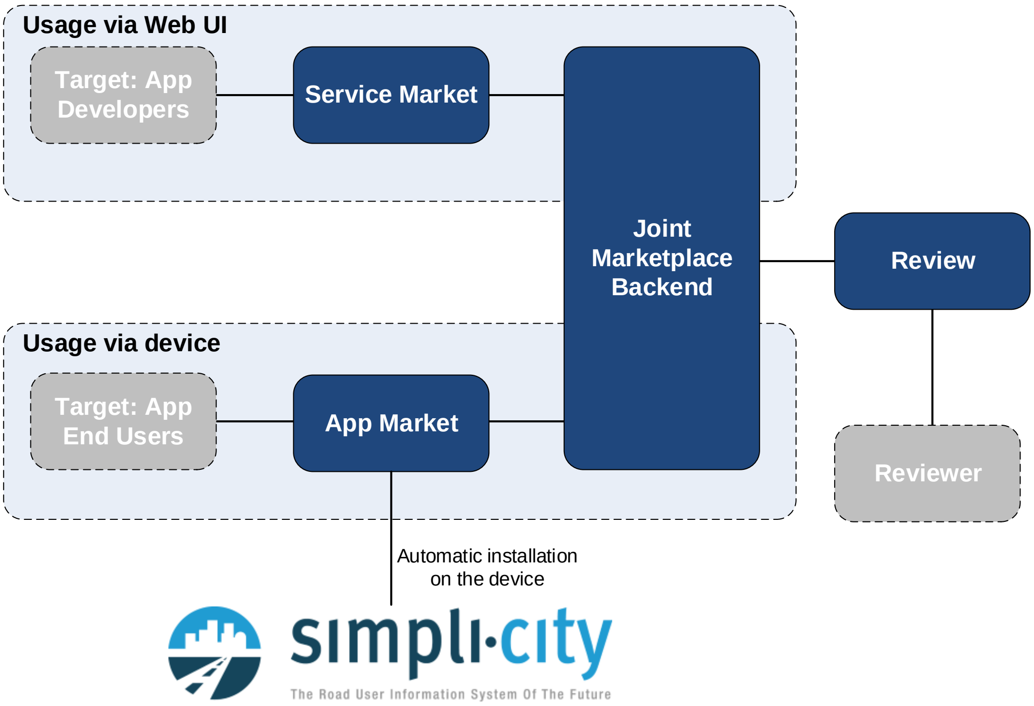

Figure 11: Marketplace Structure

While the two parts of the marketplaces will make use of a similar underlying technology for describing and storing software artefacts, they will be clearly separated (as depicted in Figure 11):

The Mobility *Application* Marketplace will list all apps that have been developed and will allow users to browse, search, find, and install them on their mobile devices by integrating it into the PMA. The marketplace will also facilitate user comments and apps ratings for stating feedback to other users of the SIMPLI-CITY apps. Those apps will be implemented within the use cases of the project, namely within the tasks T7.2 and T8.2. Nevertheless, the Application Marketplace will be open to all submissions.

As services aim at software developers, the Mobility *Service* Marketplace will additionally present technological information about a service's functionalities, including details like offered operations and how to invoke them.

Although the SIMPLI-CITY project itself will only be able to implement a limited set of services and apps, the marketplace itself will be constructed for the wider audience in order to allow third parties to contribute to the project results. As such, the marketplace will therefore implement a solid submission, review, and acceptance process, like it is found in mobile app stores today (e.g., the Apple AppStore or Google Play).

Services that deliver required data to the apps of the SIMPLI-CITY use cases will be implemented within the tasks T7.2 and T8.2. These services will be available and accessible from those apps inside the Mobility *Application* Marketplace, coming from third party developers. A second scenario will be a developer who creates a bundle of an app and an own set of services and then submits them to both marketplaces. This way, the Mobility *Service* Marketplace achieves synergy, because it increases the amount of possibilities for every new developer to create apps and combine them with the new services.

Both parts of the marketplace share certain functions, most notably certifications, payment, and licensing. Licensing models include free usage of apps, "freemium" content (i.e., the basic app is offered for free, but for additional functionalities a premium is charged), or classic payment models.

While the Application Marketplace is mainly used by end users to install new apps on their SIMPLI-CITY-enabled device, the service marketplace is targeting app developers to create new apps based on services. Developers will use a Web-based frontend to find and use services. This may include creation of API keys and acceptance of licensing conditions. End users will use a device-specific frontend to find and install new apps.

Interactions and Dependencies: This component depends on the Service Registry (T5.3), which will register and manage new services based on their individual manifests. For deploying new services, a connection with T5.3 will be established allowing a deployment and undeployment of services. There may be feedback messages sent between the Service Runtime Environment (T5.3) and the marketplaces in order to display information about the SLA conformity as described in Section 4.2.8. The technical description of apps and its commands is derived from the manifest file of the Application Design Studio (T6.4).

4.2.10 Voice-based Multimodal User Interface (Tasks T6.1 and T6.2)

Motivation: Voice interfaces are becoming more common –- and for a system that is supposed to be used while driving, voice is the preferred mode of interaction. Most currently available voice-controlled apps are built using a state-based approach, where application logic and dialogue logic are managed within one single structure. The Talkamatic Dialogue Manager (TDM), which will be the base of the PMA user interface, separates interaction logic from application logic, resulting in a shorter development cycle, which is crucial in the context of this project where third parties are invited to build new apps.

Purpose: To provide a generic user interface to a wide variety of apps. The user interface allows the kind of low-distraction/high-efficiency interaction that is required in an in-vehicle environment.

Input(s): Haptic input from touchscreen/buttons etc., as well as speech from user, events and data from apps.

Output(s): Spoken feedback and information to the user, graphical and textual output on screen, well-formed application calls to apps.

Description: This component is the user interface layer of SIMPLI-CITY, taking user input in the form of utterances and Graphical User Interface (GUI) input, managing the need for further user input, and transforming them into application calls (see Figure 12). The result of the application call is then fed back to the user, using a combination of speech, text and graphics. User input is collected using an ASR (Automatic Speech Recognition) subcomponent or from user activity in the GUI component. Both input types, speech and haptic, are interpreted to dialogue “moves” (dialogue acts, such as ask, answer, request) with some propositional content. The dialogue component can also be triggered by application activity, and fetch input data from the application instead of asking the user.

Every dialogue interface description includes its own interpretation resources (or a method for accessing such resources) along with descriptions of domain logic and interfaces to applications and services.

The ASR can be either local or Cloud-based, or a combination of both. In the case of a local ASR, the ASR is dependent on a language model for each application. This model – or methods for obtaining it – needs to be provided along with the other components of the interface description within the application manifest. Interpretation can be done on-board or in the Cloud. If the ASR engine is a Cloud-based general dictation engine (e.g., Nuance Dragon Mobile), the interpretation engine needs to be more powerful, as the recognition hypotheses from a dictation ASR are not restricted by a tailor-made language model, but are restricted only by a very general language model. These general language models are based on statistical models describing other kinds of talk than the specialised utterances of the current applications.

The dialogue moves are forwarded to the central subcomponent DME, the Dialogue Move Engine. The DME contains a description of the dialogue context, including information about recent utterances, recent questions, active apps etc., and makes decisions about the system’s next move based on this information. The next move can be to “do nothing”, to “ask a question”, “answer a question” or to “carry out an app(s) call”. Any resulting dialogue move output from the DME will be sent to the two generation subcomponents, which will generate GUI contents and an utterance, respectively.

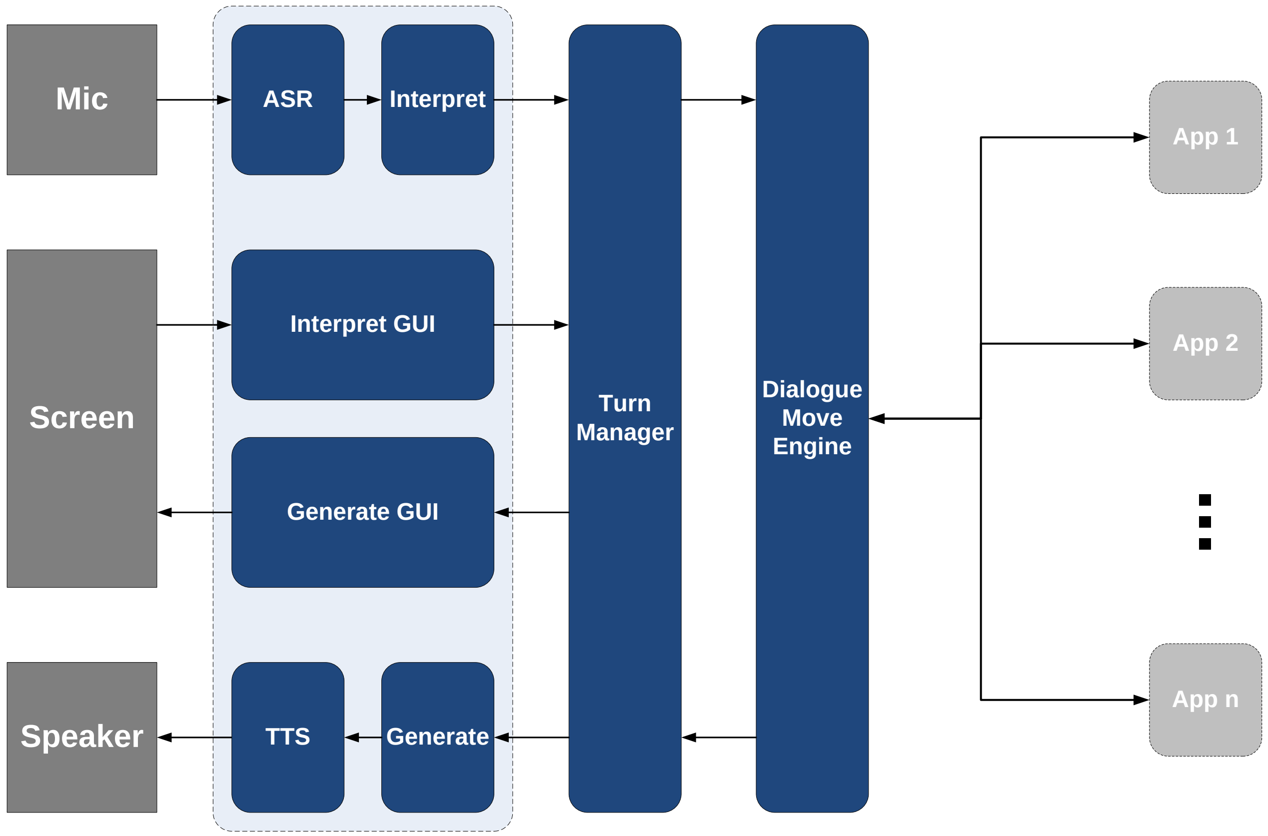

Figure 12: Multimodal User Interface

The final step in the iteration is to utter the utterance, which is done using a TTS (Text-To-Speech) component, and to send the GUI contents to the GUI component. A Turn Manager subcomponent is used to manage the right and opportunity to speak during the dialogue. To make an utterance in a dialogue is often referred to as a “turn”. The Turn Manager distributes the turn between the user and the system.

Interactions and Dependencies: The component interacts with the T6.3 Mobile Application Runtime Environment for calling apps. The interface descriptions used for determining the app logic will be generated by the T6.4 Application Design Studio.

4.2.11 Mobile Application Runtime Environment (Task T6.3)

Motivation: SIMPLI-CITY will allow users to run various apps for different purposes. This means that an Application Runtime Environment is needed in order to launch, update and install apps once they are selected in the app marketplace.

Purpose: The Mobile Application Runtime Environment will provide an API that is usable by (third party) app developers to access typical functions of SIMPLI-CITY such as the communication with server-side systems. The Application Runtime Environment will also manage a list of app commands derived from the app manifest of the design studio. This list will be the base for the Voice-based Multimodal User Interface, too.

Input(s): API calls of running apps, e.g., to retrieve data, to invoke backend services or to communicate with backend services.

Output(s): The result(s) of one or more functions of a set of API calls.

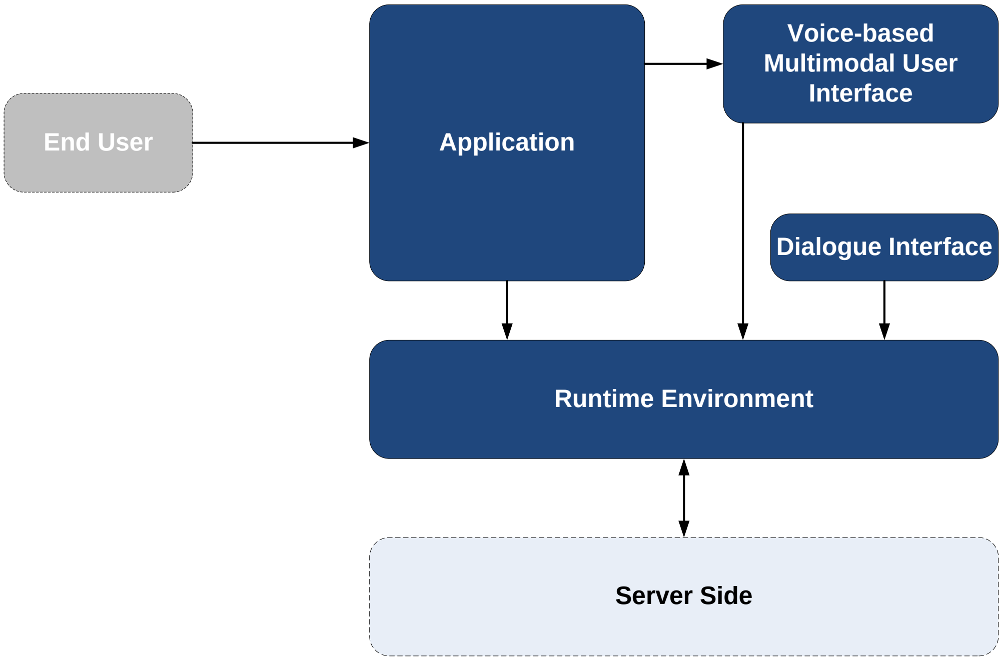

Figure 13: Mobile Application Runtime Environment

Description: This task will provide a runtime environment for apps running on the mobile device. Analogous to the SIMPLI-CITY Service Runtime Environment, which provides the backend functionality for the apps, this component will deliver a user frontend, i.e., execute apps and integrate them into the PMA (see Figure 13).

The execution of apps includes the local operations on the user's device as well as the interaction with the services in the backend. One particular aspect that needs to be facilitated by the Application Runtime Environment is proactive behaviour of some apps and the interaction with the functionalities of the Voice-based Multimodal User Interface.

The SIMPLI-CITY Mobile Application Runtime Environment also brings the technological foundation for apps to be deployed on different end user devices.

Succinctly, the Application Runtime Environment will be acting as a controller for executing apps and it will provide an API with a set of core functionality. This core functionality will extend the API of the operating system with typical tasks needed by SIMPLI-CITY apps. For example, this may be a wrapper to access the Cloud-based Information Infrastructure or an easy access to local sensors.

Additionally, the Application Runtime Environment will provide a local storage which may be used during runtime by apps. This storage is a local completion of the Cloud-based Information Infrastructure and will be used for data that will only be needed locally and where the latency of Cloud-based Information Infrastructure is unsuitable.

Interactions and Dependencies: The main interaction point of the Application Runtime Environment is with the apps which will be developed within the tasks T7.2 and T8.2. Additionally, the Mobile Application Runtime Environment will invoke services via the Service Runtime Environment (T5.3) and will therefore handle communication with the server side. Additionally there is a link to the Application Marketplace (T5.4) for finding and installing new apps.

4.2.12 Application Design Studio (Task T6.4)

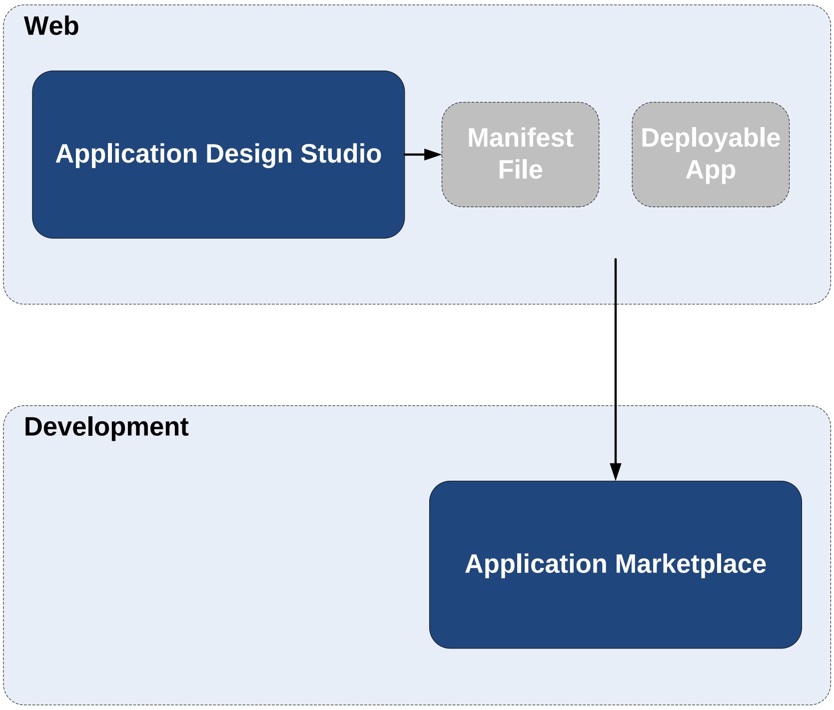

Figure 14: Application Design Studio

Motivation: Creating new apps for SIMPLI-CITY is a key to the success of the project. As such, third party app developers will be assisted in their development work.

Purpose: Provide assistance to app developers within an integrated toolset (Studio) for describing applications in form of a manifest file and preparing them for the deployment within the Application Marketplace.

Input(s): App code

Output(s): Deployable apps

Description: The Application Design Studio provides app developers with the necessary tools and means to develop apps on end user devices (see Figure 14). The app developer will be supported by a step-by-step procedure that covers the entire development process, including the registration of new apps (or corresponding updates) at the SIMPLI-CITY Application Marketplace.

This registration and deployment process will be performed similar to the app registration and deployment in existing app stores such as the Windows Phone Marketplace, the Apple AppStore or Google Play. The Application Design Studio may therefore be compared to the Google developer dashboard and will have to closely interlink to the Application Marketplace (T5.4).

The Studio will be realized as a tool providing a GUI to define all app data. It will support app developers to easily submit their apps to the Application Marketplace (T5.4). This includes definition of an app manifest file to specify the runtime environment requirements and to specify properties of the apps such as the app icon or the dependencies to other apps. The resulting manifest file will be included along with the app when it is submitted by the developer.

Additionally, the manifest will define commands that an app can react to. This list of commands will be used by the T6.1 and T6.2 components and also by the Mobile Application Runtime Environment (T6.3) in order to launch an app. The manifest will therefore be the main step in the app design and deployment process. The commands listed in it will form the connection between the user and the Application Runtime Environment. For example, a statement like “find the next parking space” will be used to later allow the Application Runtime Environment to find out which application to launch as soon as the user has pressed the “Speak With The PMA” (SWP) button and has spoken those commands.

Overall, the Application Design Studio offers an appropriate step-by-step procedure to app development, assisting the app developer during the entire development process. It delivers everything that a developer needs to prepare an app for the usage within SIMPLI-CITY.

Interactions and Dependencies: The Application Design Studio has no direct dependency but it will act as a tool for making the development easier and will therefore have an indirect interaction with the Application Marketplace (T5.4) and with the app development conducted in the use case tasks T7.2 and T8.2.

5 SIMPLI-CITY Scenarios

In this section, an outline for the scenarios and the use cases that are considered in SIMPLI-CITY will be presented. For this, potential scenarios from the SIMPLI-CITY use case partners SRM (Section 5.1), IBM (Section 5.2), CRF (Section 5.3), and Tempos 21 (Section 5.4) will be introduced. This information is illustrative only, but it shows examples of what the project aims to achieve. It also provides a foundation for the Requirements Analysis Report (deliverable D2.3). The actual use cases will be defined within separate tasks in SIMPLI-CITY.

While the following scenarios are used to demonstrate different aspects of possible SIMPLI-CITY use cases, they are per se unrelated. During the use case specifications in WP7 and WP8, commonalities of the scenarios will be identified in order to avoid double efforts.

Notably, the scenarios focus on car drivers, but could also be applied for truck or bus drivers, and more importantly to any road traffic related systems. Some of the apps that will be presented, including the accident prevention, media streaming apps, road traffic diagnosis, or prediction can be beneficial for bicyclists and pedestrians and even city managers. Per se, SIMPLI-CITY is not bound to specific data sources, existing services, or particular modes of transportation. SIMPLI-CITY, by design, is for potential use in arbitrary road user scenarios. Furthermore, in the upcoming Use Case Definitions, the integration of data sources like bus, train timetables, or park and ride facilities, will also be regarded, thus allowing a car driver to switch modes of transport but also judging and anticipating traffic situations.

5.1 SRM Scenario

The City of Bologna has started a policy of sustainable mobility, in particular with regard to the historic central area, which aims to extend pedestrian areas during Saturday, Sunday and bank holidays (the so-called TDays). Furthermore, the City of Bologna aims at reducing cars', motorbikes' and buses' access to the city centre and increasing the use of bicycles. Usually, the historic centre is mostly a Limited Traffic Area (LTA) where only residents and vehicles with specific authorization are allowed to get in.

As a consequence, the access to certain areas of the city as well as the public transport systems (i.e., bus lines network) are subject to relevant changes from weekdays to TDays, so that the quality of information that is provided to users plays a strategic role in this scenario. The City of Bologna is addressing all its efforts towards eco-friendly vehicles, such as bicycles, extending the cycle lanes network, connecting existing paths, creating new ones, providing information about cycle lanes' locations as well as enhancing bike sharing. Furthermore, three charging points for electric cars have recently been installed and advertised among city residents.

Besides all actions mentioned above, the SIMPLI-CITY project introduces a new idea of sustainability relating to road users. The aim is to enlarge the concept of inter-modality involving car drivers and promoting car sustainability through the optimization of routes and the implementation of a real-time information system. This is an example of important instruments that support car drivers and, at the same time, reduce congestion in the city.

5.1.1 Finding the Best Transport Mode

On a Saturday afternoon, Guido wants to go to Via Rizzoli, in the city centre (T-Zone) to meet his friends and have a drink with them. It is currently 5 pm and he has to be there by 6 pm. Being a Saturday afternoon, driving restrictions are in force: in addition to the normal Limited Traffic Zone restrictions on TDays, the three main streets in the central area (T-zone) are pedestrianized and therefore even buses are rerouted elsewhere.

Guido does not know what might be the fastest way to get there. Should he take the car? Or would it be faster to go by bus? As it is a sunny afternoon, the bicycle could be a good choice as well, but he is 5 km away from the centre and going by bicycle could take too much time. Guido decides to start an appropriate app in his PMA in order to ask for a suggestion. The PMA is a portable device which he can connect to his car or even to his bicycle if necessary.

At this stage, Guido enters the necessary inputs into the app: the starting point (which may also be the user's current position, detected by GPS) and the destination, which can be selected on the map, or can be an address inserted manually (by voice) or can be chosen among POIs (Point of Interests). Furthermore, as the destination is one of the city areas where some traffic limitations are generally in force, the user should also specify if an authorization to access that area is held. However, in SIMPLI-CITY, such information about the authorization is already stored in the user profile and can hence be automatically used as context data.

The app then processes all the information, calculating some possible routes, choosing between different possible means of transport by taking into account predefined user preferences. The app takes into account the real-time data available, i.e., the traffic congestion level on the route to the destination, the road works, available parking lots, bus paths and delays, etc.

The app proposes three potential alternatives to Guido, giving him also information about trip costs (e.g., bus tickets, parking costs): He can use his car and park at the underground parking in Piazza VIII Agosto, where at the moment there are 25 parking lots available and then walk for 400 meters – it would take 35 minutes in current traffic conditions. Alternatively, Guido can walk to the nearest bus stop, get on bus number 27, get off at Via Irnerio (not at the usual stop, Via Rizzoli, because it is a pedestrian area and buses are rerouted on Saturdays), walk 600 meters from the bus stop: Considering the bus delay of five minutes and the current traffic, the whole trip would take 45 minutes. Cycling is another opportunity – the estimated arrival time is within about 50 minutes. Even if the car is probably the most expensive choice, it is the fastest one, so Guido prefers to use it.

5.1.2 Travelling

Guido begins his trip by car. When Guido is 1 km away from the envisaged parking lot, the app informs him that the number of lots available in the selected parking has dropped to four. As a consequence, the app recognises that the chance to not find a parking spot becomes very high and consequently recalculates the route. The app suggests that after 500 meters, there is another parking where the number of available lots is 40. Guido accepts the change and the app leads Guido through the voice system. Guido arrives at his new destination and parks the car.

During the trip, the app checks the real-time data that may affect the travel by car, in particular the traffic congestion level and the parking availability. At this stage, the app will

execute two tasks: First, it provides the user with information and updates concerning the route to follow and second, it assesses the need to modify the route, based on this real-time data. All information is provided through the voice system, so Guido has his hands free, in order to drive in a safe way.

5.2 Dublin Scenario

By recently opening up its data as part of the Dublinked initiative (see Table 1 in Annex), Dublin City has enabled new ways of managing its data, especially in the transportation domain. In particular, Dublin City asked for new analytics tools for managing large cities such as Dublin City. The following scenario focuses on providing to road users applications and services that will help to improve their driving and travelling experience in Dublin City (and could be generalized to other larger cities). The scenario shows some examples of applications and services that make use of media streams, open data repositories, or link to social networks.

5.2.1 Road Traffic Diagnosis

Every year, road traffic congestion wastes billions of hours of time and produces tons of carbon in the atmosphere. More important, it is getting worse, year by year. It also stresses and frustrates motorists, encouraging road rage and reducing health of motorists, and more dramatically interferes with the passage of emergency vehicles traveling to their destinations where they are urgently needed. All of them, among others, are examples of negative effects of congestion in cities.

Traffic congestions can be easily detected, visualized and analysed through stream traffic data (e.g., GPS location of vehicles, loop detector). Optimization mechanisms using existing data mining and machine learning approaches are examples of techniques used by modern traffic systems. However, the problem of explaining congestion causes is a more complex challenge. About half of traffic congestions is recurring, and is attributed to rush hours; most of the rest is attributed to traffic incidents, road works, major events or weather conditions. Obtaining explanations in real-time when a congestion suddenly occurs in the road network is yet an unexplored problem in transportation research.

This scenario addresses this diagnosis problem, i.e., how to identify the nature and causes of congestions by answering related questions, e.g.,

- How do large events such as a concert could impact traffic conditions?
- Shall delays be expected?
- Is re-routing appropriate?

Such questions remain open because (i) relevant data sets (e.g., about road works, city events), (ii) their correlation (e.g., road works and city events connected to the same city area) and (iii) historical traffic conditions (e.g., road works and congestion in Dublin's Canal street on July 24th, 2010) are not fully open and jointly exploited. Pure Artificial Intelligence (AI) diagnosis approaches focus on point (iii) for inferring the cause-effect relationships while Semantic Web technologies tackle points (i) and (ii) for integrating heterogeneous and large data. However, pure AI diagnosis approaches fail to timely compute diagnosis results for large-scale applications such as the traffic one. In SIMPLI-CITY, current approaches will be extended to compute accurate diagnoses for situations where cause-effect relationships have not been established before. The list of potential

heterogeneous sources of effects (road traffic congestion) and their causes (e.g., road weather conditions, events) that are considered in this scenario will be provided within city partners of the consortium, i.e., Dublin City (through dublinked.ie agreement) and Bologna.

Using the context of SIMPLI-CITY, various sources of data from the transportation department of large cities (first Dublin, Bologna) will be used to decrypt the reason of congestions in such cities. The automated diagnosis method, core reasoning service of SIMPLI-CITY, will be used for helping the car driver to have an understanding of real-time traffic situation in cities. This aspect of the SIMPLI-CITY framework can be used by car, bus or truck drivers, and pedestrian but also city managers.

Using SIMPLI-CITY, it will be possible to automatically detect real-time congestions as traffic anomalies and retrieve their diagnosis as the set of possible events that could be the causes. In a traffic context, queries such as “Why is there a traffic congestion in Dawson road now?” are answered by our approach: “This is caused by a music concert in Canal road that starts in 30 minutes, with a probability of 0.4”.

5.2.2 Road Traffic Prediction

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, through this scenario and its exposed services, aims to provide such a navigation system. Road traffic conditions will be predicted so road users are satisfied with the road infrastructure of the city.

SIMPLI-CITY will make use of the spatial-temporal correlation of heterogeneous data for anticipating critical situation in road traffic and then provide flexible services to car drivers. In particular, SIMPLI-CITY will make strong use of the auto-correlation of heterogeneous stream data from various Dublin City sensors (real-time weather information, bus information, travel time estimates) to understand how data is linked over time and sensors. This information can then be used in backend services and end user apps.

Contrary to diagnosis, which correlates causes to its effects, road traffic prediction aims at deriving its effects. Similarly to the diagnosis scenario, SIMPLI-CITY will integrate relevant information from various sensors in Dublin City.

5.3 CRF Scenario

The number of drivers who are interested in an eco-sustainable mobility is growing every day. This scenario will provide an overview of the services and end user apps that SIMPLI-CITY will offer in order to enhance the driver’s eco-driving experience. SIMPLI-CITY will enhance the driver’s experience in order to achieve the optimal balance among a number of factors, like for example:

- Eco-friendliness (saving unnecessary fuel)
- Estimated travel time (for each single destination and for the total time to the final destination)
- Other specific features (e.g., related to the target destination or the recommended time to reach a destination)

Furthermore, the scenario contains sophisticated functionalities for Eco-efficiency comparison between different journeys, routes, and even drivers, including the Eco-driving Contest.

5.3.1 Different Route Comparison

Every weekend, Sara performs some different repeating tasks, e.g., she goes to the her favourite supermarket to do grocery shopping, pays her bills at the same post office or bank, does shopping and other business she has not time to perform on the working days, brings her children to the swimming pool, takes care of herself going to the gym, the hairdresser or the beautician. SIMPLI-CITY will help Sara to plan and perform her tasks considering which are the ones with a fixed time schedule like the swimming lessons of her children and her appointments with the hairdresser, but at the same time suggesting to go to her favourite supermarket at a certain time due to the traffic flows in that area at the weekend and avoid to go to her usual post office because nearby there is a special event and all the streets are closed to traffic in that particular area.

Similarly, Sara drives every (working) day in order to perform different repeating tasks, e.g., to bring her son to the primary school and her daughter to the nursery, and then reach her office. In this case, SIMPLI-CITY will help Sara to define the best route in order to maximize her performance in terms of actual mileage, duration of the trip, departure and destination, and the comparison with her own goals, keeping into account the current traffic and weather conditions and possible special events on her usual route.

SIMPLI-CITY will also allow comparing (eco-)efficiency of different routes by combining the available real-time information about routes, information about the driver's average driving style, the vehicle topology, etc. Data may also be obtained from cooperative systems, telematics, mobility-related smart objects, and open (government) data repositories on the Web. Information about Sara's driving style and vehicle characteristics will be taken from and calculated by the car, respectively. The best eco-aware route solution will then be proposed to Sara, based on her personal preferences.

5.3.2 Eco-Driving and Fuel Consumption of a Journey

Sara focuses much attention to her eco-drive style and tries to reduce the carbon footprint and the fuel consumption on her route. In this context, SIMPLI-CITY will inform Sara about her driving style performance with regard to acceleration, braking, gear changing, etc. An according app making use of this information will be able to provide her with suggestions on how to improve her performance and how to save fuel for that particular recurrent journey. The SIMPLI-CITY PMA will also be able to provide live feedback about Sara's driving style through an according app and will suggest Sara to decrease the vehicle speed in order to reduce the current CO_2 emission.

5.3.3 Vehicle Information on the Web

When Sara gets back home, she would like to check the (eco-)driving performance of her daily journeys. For this, SIMPLI-CITY allows Sara to access journey-related data using her Web browser and the verification if she follows an eco-driving style or not. She can check how to improve the duration of the trip and how she can save money by reducing the fuel consumption. Furthermore, she is able to access a driver's log on the car's Website.

Sara's ability will be represented with some leaf icons on the Website. For instance, Sara will receive 4 leaves out of 5 possible leaves for her acceleration behaviour if she is

following a near-optimal driving style. Similar evaluations will be given for other eco-driving performance metrics like braking and gear changing; an overall assessment will also be given.

5.3.4 Eco-efficiency Comparison and Eco-driving Contest

Sara is part of a community that shares its eco-driving information through a SIMPLI-CITY-enabled app in order to compare the individual performances. This way, it is possible to 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.

Sara also participates in the *Eco-driving Contest*, which compares drivers using similar vehicles on similar routes and awards the “winners” on a regular basis. The best performing drivers will earn rewards from the SIMPLI-CITY system, e.g., by “liking” or giving “+1” to their driving performance in social networks like Facebook or GooglePlus. This further improves the motivation of Sara to act environmentally aware by publicly announcing good results.

An Eco-driving Context could also be hosted by municipalities or similar governmental agencies in order to award eco-friendly driving behaviour: In such a scenario, Sara would receive some award points from a municipality which she can convert into a 20% discount on her next parking ticket or a free ticket for the bus.

5.4 Tempos 21 Scenario

The Tempos 21 scenario shows different apps and services implemented using the SIMPLI-CITY framework, which make use of media streams, open data repositories on the Web, or link to social networks, with the objective to increase the driver’s safety and to provide entertainment services to the passengers of the vehicle.

5.4.1 Holiday Route Planning and Navigation

Mary is preparing a holiday by car that will take place the following weekend. For this, she uses an already installed SIMPLI-CITY-based *Holiday Route* app running within her PMA, which she has connected to her car. This app allows her to plan the route and specify the main places she wants to travel to. She also indicates some POIs near the planned route that she wants to visit. The information of this route and the associated POIs are stored in her Cloud-based user account.

Once the day of the journey has arrived, Mary is reminded by her PMA that she has planned a holiday for this day. Hence, the Holiday Route app is started. The app asks Mary about her desired destination, and she selects the route she has prepared for the journey. The app downloads the route from the Cloud and loads it. Then the app guides her through the journey.

Mary interacts with the PMA by means of voice commands. The app invokes a Holiday Route backend service in order to make use of the stored route by interacting with the Service Runtime Environment and the Service Registry. Then, the Runtime Environment invokes the Holiday Route backend service, which provides the required route. The Holiday Route app also makes use of a navigation backend service, which provides real-time information about the route.

5.4.2 Multimedia Guide Integration

The Holiday Route backend service makes use of the Context-based Service Personalisation component in order to detect if the car is driving to a previously selected POI. When Mary is close to a POI that she had previously selected when preparing her route, the Holiday Route service detects it and notifies the Holiday Route app, which is therefore able to inform the user about the POI. In this case, the POI has a multimedia guide available. The app informs Mary about the proximity of the POI, and also asks her if she wants to playback the multimedia guide. Mary does not have time to stop and visit the POI, so she decides to listen to the multimedia guide while driving. The app starts the reproduction of the multimedia guide.

When the proximity of a POI is detected, the service notifies the Holiday Route app about it, which informs the user by means of the PMA.

5.4.3 Media Stream Access

Mary likes listening to music while driving, so she asks the PMA to start the *Radio* app she normally uses. Then she asks the app to connect to her preferred streaming station. The Radio app interacts with a *Music Streaming* backend service. This service manages the connection with an external streaming service like SoundCloud or Last.FM. The streaming backend service makes use of the functionalities of the Media Data Streams and Data Prefetching component, which provides the means to use the external radio data streams and offers functions for data prefetching, aiming at non-disruptive media playback, which is especially useful in cases of bad connectivity. This media stream is sent to the Radio app, and played to the user through the PMA using the car hifi facilities.

5.4.4 Accident Prevention Warning

The Holiday Route app is also able to get data from the *Accident Prevention* backend service of SIMPLI-CITY. This service makes use of the Context-based Service Personalisation component and it continuously monitors the location of Mary in order to determine if she is close to any dangerous driving location; for this, an open government data source hosting such information is queried. When the proximity of a dangerous driving point is detected, the Accident Prevention service notifies the Navigation app about it, and then the app informs Mary that she is close to an area with high number of car accidents. Mary was a little distracted, so she appreciates this notification and decides to reduce the speed of the car.

5.4.5 Social Network Data Integration

Mary is driving next to a POI and she thinks that it would be cool to share her location with her friends using a location-based social networking application like Foursquare or Facebook. She knows that the required app is not installed in the PMA, so she asks the PMA to use the Applications Marketplace component of the SIMPLI-CITY framework and search for the application. The PMA downloads, installs, and starts it. Mary logs into the service (e.g., Foursquare service or Facebook service) using her user credentials and then indicates to share her location with her friends. The communication of this location-based app is routed through the Service Runtime Environment to the corresponding backend service.

Annex

Table 1: Open Data Sources that may be used in SIMPLI-CITY

Location	Type	Data Source	Description	URL
Amsterdam, Netherlands	Public transport	Amsterdam transport API	Public transport information.	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Events	Cultural events in Catalonia	Cultural events that take place throughout Catalonia.	http://www20.gencat.cat/portal/site/dadesobertes
Barcelona, Spain	Public transport	Biking service in Barcelona	Information about the Biking stations of Barcelona, including number of bikes and parkings available in each station.	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Public transport	Bus stations of Barcelona	Location of the bus stations of Barcelona, and the bus lines of each station.	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Public transport	Metro stations of Barcelona	Location of the metro stations of Barcelona.	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Public transport	Public transport incidents in Barcelona	Status of the public transport in the metropolitan area of Barcelona.	http://www20.gencat.cat/portal/site/dadesobertes
Barcelona, Spain	Public transport	State of train lines in Catalonia	State of the lines of the Rodalies de Barcelona and regional services.	http://www20.gencat.cat/portal/site/rodalies/?newLang=en_GB
Barcelona, Spain	Tourism	Tourism information of Barcelona	Information about touristic sightseeing in Barcelona	http://opencities.upf.edu/osnweb/serviceList.action

Barcelona, Spain	Traffic	Traffic status in Barcelona	Current status and prevision of the traffic in the streets of Barcelona	http://w20.bcn.cat/opendata/
Barcelona, Spain	Traffic	Traffic status in Barcelona	Current status and prevision of the traffic in the streets of Barcelona	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Traffic	Private Parkings in Barcelona	Location of private parkin facilities in Barcelona, including information about price per minute	http://opencities.upf.edu/osnweb/serviceList.action
Barcelona, Spain	Weather	Weather forecast of Catalonia	Weather forecast of Catalonia, updated 3 times a day.	http://www20.gencat.cat/portal/site/dadesobertes
Berlin, Germany	Public transport	Berlin transport API	Public transport information about bus and metro lines.	http://www.vbb.de/en/index.html
Bologna, Italy	Public transport	CISIUM - BUS delay data	Delays of buses in real-time	http://cisium.webhop.net/
Bologna, Italy	Traffic	Open data Bologna - Cycle lanes	Digital mapping of urban cycling lanes	http://dati.comune.bologna.it/node/249
Bologna, Italy	Traffic	Open data Bologna - Park-o -meter	Digital mapping of park-o-meters locations	http://dati.comune.bologna.it/node/439
Bologna, Italy	Traffic	Traffic control centre - Traffic flow	Dynamic data on Traffic flow	http://cisium.webhop.net/
Bologna, Italy	Traffic	CISIUM - Index of traffic congestion	Traffic status on road links	http://cisium.webhop.net/
Bologna, Italy	Traffic	CISIUM - parking status	Parking spaces availability in parking structures	http://cisium.webhop.net/
Dublin, Ireland	Events	Events in Dublin	Events with small attendance	https://www.eventbrite.com/api http://api.eventful.com

Dublin, Ireland	Events	Events in Dublin	Events with large attendance	http://upcoming.yahoo.com/services/api/
Dublin, Ireland	n.a.	DBPedia	Structured facts extracted from Wikipedia.	http://dbpedia.org/
Dublin, Ireland	Traffic	Journey times across Dublin City (47 routes)	Dublin Traffic Department's TRIPS system	http://dublinked.ie/datastore/datasets/dataset-215.php
Dublin, Ireland	Traffic	Social-Media Related Feeds	Reputable sources in of traffic information In Dublin Ireland	https://twitter.com/LiveDrive https://twitter.com/aaroadwatch https://twitter.com/GardaTraffic
Dublin, Ireland	Traffic	Road Works and Maintenance		http://dublinked.ie/
Dublin, Ireland	Traffic	Dublin City Roads	Listing of street type, junctions, GPS coordinate	http://linkedgeodata.org
Dublin, Ireland	Weather	Wunderground for Dublin	Real-time weather information	http://www.wunderground.com/weather/api/
Dublin, Ireland	Weather	Road Weather Condition	Weather conditions from 54 stations	http://www.nratraffic.ie/weather
Helsinki, Finland	Public transport	Helsinki transport API	Public transport information about bus stops, lines, routing between two points, cycling route.	http://developer.reittiopas.fi/pages/en/home.php