simpli·city

The Road User Information System Of The Future

WP3 – Architecture, Functional & Technical Specification, Security & Privacy Concept, Integration

D3.3: Holistic Security and Privacy Concept

Deliverable Lead: ASC

Contributing Partners: TUDA, CRF, SRM

Delivery Date: 27.09.2013

Dissemination Level: Public

Version 1.10

This deliverable describes the concept for privacy and security questions of the project and ensures that the concept is applicable to and followed by all SIMPLI-CITY components. It considers existing approaches (e.g., pseudonymisation and anonymisation mechanisms), and regards the specific requirements of the mobility domain, especially the need for data privacy within this domain.

Document Status	
Deliverable Lead	Dr. Sven Abels, Abdelkarim El Moussaoui, Ascora GmbH
Internal Reviewer 1	Aris Gkoulalas-Divanis, Robert Tucker, IBM
Internal Reviewer 2	Fredrik Kronlid, Alex Berman, TALK
Type	Deliverable
Work Package	WP3: Architecture, Functional & Technical Specification, Security & Privacy Concept, Integration
ID	D3.3: Holistic Security and Privacy Concept
Due Date	30.09.2013
Delivery Date	27.09.2013
Status	Approved

Document History	
Contributions	V1.0, ASC, TUDA, CRF, SRM, 27.09.2013
Final Version	V1.10, 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

This deliverable describes the concept for privacy and security aspects of the project and ensures that the concept is applicable to and followed by all SIMPLI-CITY software components.

The document starts with an overview about privacy aspects and defines the scope of privacy issues within the project in comparison to security aspects. It then highlights the requirements towards the project in order to consider privacy and it defines a concept of the project which will be followed during the prototype developments and during the use cases. It considers existing approaches (e.g., pseudonymization and anonymization mechanisms), and regards the specific requirements of the mobility domain, especially the need for data privacy within this domain.

Afterwards, the document focuses on security aspects with a focus on technical security including a short overview about standards and approaches in this domain. It then defines a clear concept and a set of action points for the work packages in order to address security issues during the project lifetime.

Finally, the document briefly discusses ethical and legal aspects.

Table of Contents

1 Introduction

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

This document provides an in-depth description on how SIMPLI-CITY will address privacy and security aspects within the project with a focus on a technical viewpoint.

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 apps 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 App Marketplaces.

In order to foster the wide usage of these 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

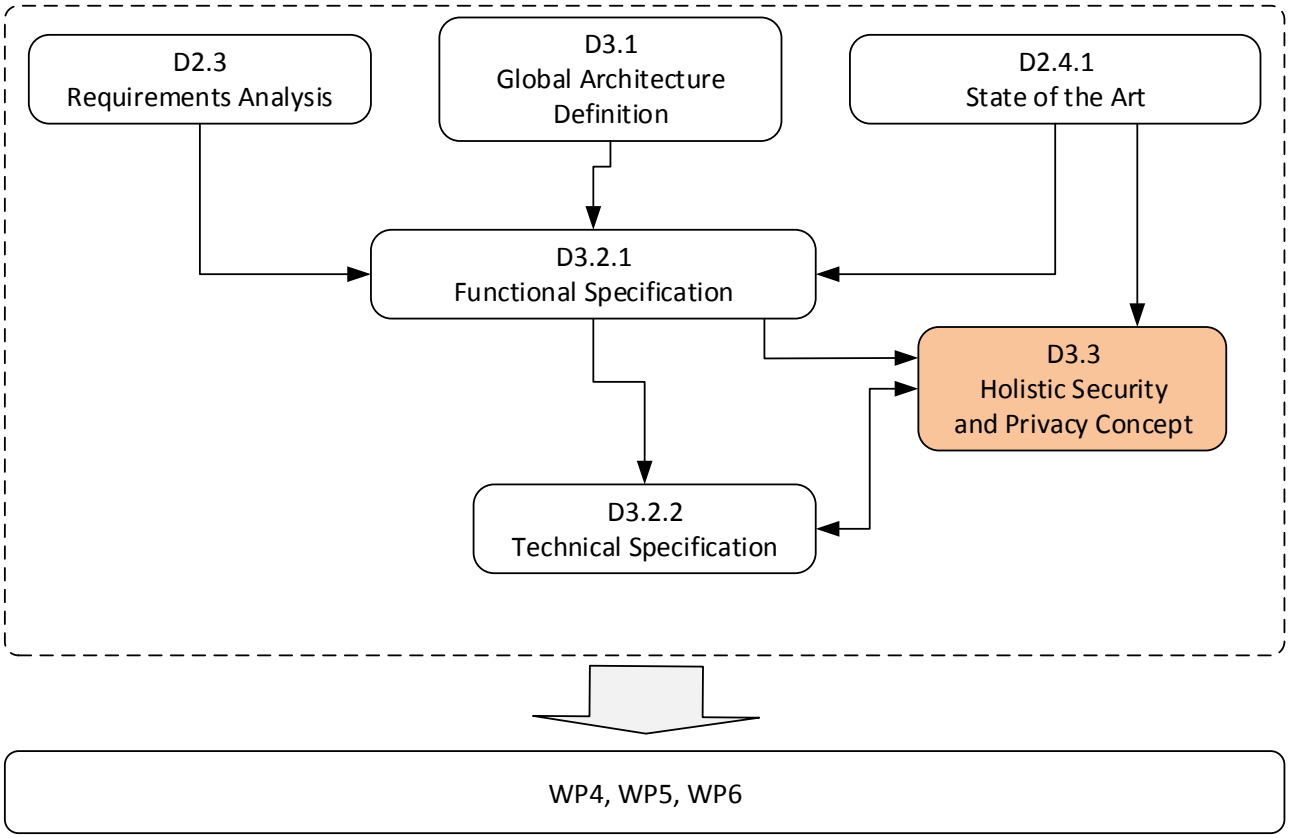

Figure 1: Context of Deliverable D3.3

The purpose of this document is to provide a concept for privacy and security aspects of the project. It ensures that the concept is applicable to and followed by all SIMPLI-CITY components and it considers existing approaches such as pseudonymization and anonymization mechanisms. For this purpose, this document considers the identified state of the art at deliverable D2.4.1 and takes into account the former deliverables of the project: the Global Architecture Specification (deliverable D3.1) and the Requirements Analysis (D2.3) indirectly via the Functional Specification (deliverable D3.2.1). The holistic security and privacy concept has a strong impact into the development of the different components in WP4-6 as depicted in Figure 1. As such, it also synchronizes with and extends the Technical Specification (deliverable D3.2.2).

1.3 Document Status and Target Audience

This document is listed in the Description of Work (DoW) as “public”, since it provides the general approach of the project for addressing privacy and security concerns 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 provides an overview about privacy aspects and starts with a short definition of the scope of privacy as such. Afterwards, it defines requirements to the project, which may be seen in addition to those requirements defined in deliverable D2.3. Section 2 then continues with a list of possible standards before it describes the concept of the project. Finally, the section ends with listing concrete action points for the technical work packages WP4-6 and the use case work packages WP7-8 for each concept.
- Section 3 follows the same structure as Section 2 but it addresses the security aspects of the project and focuses on a more technical level.
- In addition to the privacy and security aspects, Section 4 describes ethical and legal aspects of the project with a focus on a European level.
- The document finishes with a short conclusion in Section 5 and afterwards lists the references of this deliverable.

Figure 2 illustrates the structure of this deliverable.

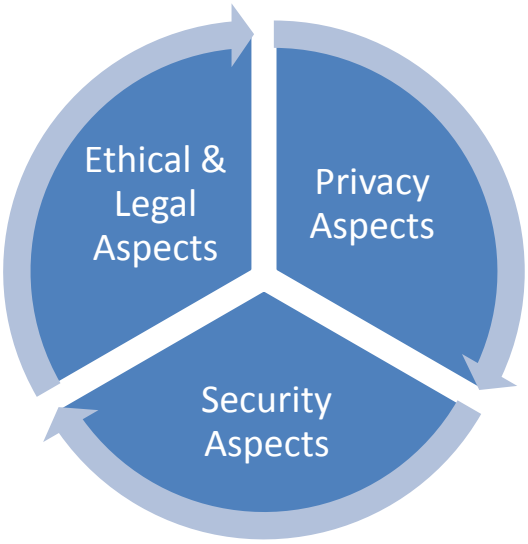

Figure 2: Structure of this Deliverable

2 Privacy Aspects

2.1 Scope

2.1.1 Definition of Privacy

Westin defines privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others" [We67].

The scope and consent of what is considered to be “private” differ among cultures and individuals, but share basic common themes. Privacy may therefore include anonymity, the wish to be unidentified in some cases. When an element within an IT-system such as SIMPLI-CITY is referred to as being “private”, it usually means that it may be considered as being especially important or personally sensitive. The degree to which private information is exposed therefore depends on how the public will receive this information, which differs between places and over time. Privacy may intersect with security aspects, including, e.g., the concepts of appropriate use, as well as protection of information. Almost all countries have laws which in some way limit privacy (see [AM03].)

The importance of privacy was recently highlighted by the on-going discussions about PRISM and Tempora which have been revealed during the course of SIMPLI-CITY. The broad absence of privacy in this domain and the discussion connected to it, have shown the need for IT systems to respect privacy in order to form a base for a trustable environment.

2.1.2 Data Privacy

The term privacy may be used for different things in different contexts. Different people, cultures, and nations have a wide variety of expectations about what privacy covers or what causes an invasion of privacy. As an ICT project, the most relevant aspect for SIMPLI-CITY is data privacy. Data privacy in SIMPLI-CITY needs to handle the trade-off between sharing data that is necessary for allowing SIMPLI-CITY to provide its services and end user apps and protecting user specific data from being shared unnecessarily. As such the challenge in here is to share data on the one hand while protecting personally identifiable information on the other hand.

For example, a goal of SIMPLI-CITY is to help users to “become green”, i.e., to be more sensitive to the environmental impact that is caused by their behaviour. As such, SIMPLI-CITY may notify users if they are driving in a more environmental unfriendly way than the average SIMPLI-CITY user. However, in order to do so, apps running within the PMA (Personal Mobility Assistant) of SIMPLI-CITY need to:

1. Collect data about the average environmental impact of SIMPLI-CITY users on the server side (e.g., in the Cloud Storage of the Cloud-based Information Infrastructure).
2. Measure the concrete environmental impact of a user while she/he is driving for comparing the own driving behaviour with the average data sets collected in step 1.

To achieve this, users have to expose their private information, which is in this case the information about their driving behaviour. As such, this aspect of SIMPLI-CITY is a good

example for a trade-off between functionality (= providing the best possibilities to help users to become green) and privacy (= avoiding any user data to leave the PMA).

The ability to control the information one reveals about oneself via the Internet, and who can access that information, has become a growing concern and as such, it needs to be addressed by SIMPLI-CITY seriously.

In order to not hinder the functionality of SIMPLI-CITY by introducing “too much” focus on privacy aspects, a good balance needs to be found. For example, the trade-off described above could be solved by anonymizing all user specific data before it leaves the PMA. This would allow the PMA - and the apps installed on it - to help users to “become green” without negative impacts to the data privacy of SIMPLI-CITY users.

2.1.3 Intersection of Privacy and Security

Privacy and security are often considered to be identical, while both describe different aspects. Security issues are discussed in detail within Section 3 of this deliverable.

It may, however, be stated that privacy aspects of SIMPLI-CITY mainly address user specific data involving the protection of the users from re-identification, sensitive information disclosure and membership disclosure attacks while security is mainly seen as the protection of information and their systems towards ungranted access and is therefore considered on a more technical level. Privacy considerations include anonymity, controlled life cycle and the negative notion of data profiling and integration. Security considerations include confidentiality, integrity, and availability as well as non-repudiation [CL08].

Of course, both areas overlap with each other and the following Figure 3 shows the intersection between privacy and security. For example, ensuring that user specific data is not accessed by unauthorized persons touches both areas. Both – privacy and security – are seen by the project as a base requirement for achieving trust and acceptance of potential users.

To make things more clear an important distinction between privacy and security may be described as:

- Data privacy ensures that user data (or patterns extracted from the data) are released to untrusted parties but in a way that these parties cannot learn anything sensitive about the users or re-identify the users. This is achieved through controlled data distortion/modification that preserves certain characteristics (defined as utility) of the data, e.g., statistical properties.

- Security ensures that data is released in its original form but only to trusted entities of the system. The untrusted entities cannot access the data (access control) or recognize the data (encryption).

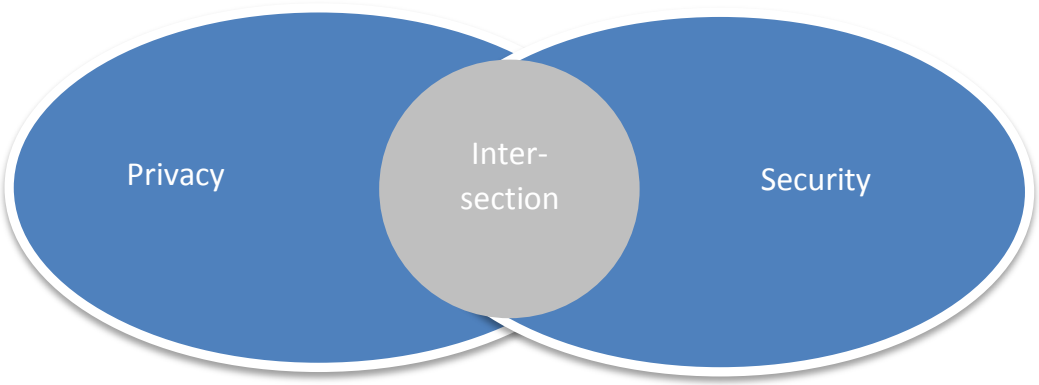

Figure 3: Intersection between Privacy and Security

2.1.4 Privacy Model

Unlike other EU projects, SIMPLI-CITY is not a closed set of components which are applied within one or more use cases. Instead, SIMPLI-CITY forms a base for building aps and services. It targets towards building a next-generation European Wide Service Platform. As such, the project cannot fully define a privacy model that lists all user data that will be stored or protected by SIMPLI-CITY because the data selection, storage and management is fully controlled by the services and apps that are to be developed by external developers and that will only partly be delivered within the use cases.

Therefore, SIMPLI-CITY does not offer a specific list of privacy guarantees issues by the project because of its nature. However, nevertheless the project approach is addressing privacy with two aspects:

- Firstly, SIMPLI-CITY will ensure that all components developed within the project respect privacy. This also includes the use cases developed within WP7 and 8.
- Secondly, SIMPLI-CITY provides a set of privacy concepts that will be fulfilled by the platform. Those concepts will be supported by the implementation of SIMPLI-CITY and will foster privacy on a conceptual level.

For achieving this, Section 2.4 describes the concept of SIMPLI-CITY for fostering privacy and lists precise action points for all technical work packages and for all use case work packages.

2.2 Impact to SIMPLI-CITY and Requirements to the Project

This section contains requirements to the SIMPLI-CITY project based on privacy considerations. These requirements can be seen as an extension of the user requirements defined in deliverable D2.3 of the project. They will be used as a base for the project in general and for the prototype implementations of WP4-8 in specific. Those requirements may be split into two types: (i) End user requirements considering privacy aspects from a user viewpoint and (ii) developer requirements considering developer viewpoints.

2.2.1 End User Requirements

The consideration of privacy and security aspects from an end user viewpoint, lead to the following requirements which have to be considered during the prototype implementations:

- Encryption of all user-related information: All data exchanged between Application Runtime Environment and the Service Runtime Environment as well as any other web call from the PMA side should be encrypted.
- Disguise physical locations: Geo locations from PMA users have to be hidden from other end users so that end users cannot be tracked because of using the PMA.
- Disguise of historical locations: Just like the current location, historical Geo locations should be hidden as well. Otherwise, this would allow third parties to track the behaviour of a user. If historic locations are saved, this should happen in an anonymized way.
- Avoid saving unnecessary data: Information which is temporarily necessary for executing SIMPLI-CITY requests but not important for further use should not be saved in the Cloud Storage or elsewhere in order to reduce storage overhead.
- Trustful system: For reaching a high market acceptance, end users need to have trust in SIMPLI-CITY. For achieving this, it is important to actively communicate the security and privacy aspects of SIMPLI-CITY to the user and to inform her/him about the data that is used and exposed by SIMPLI-CITY. A strong quality assurance of apps – as performed by the Service and App Marketplaces review – helps to increase trust in the project quality.

2.2.2 Developer Requirements

The following list shows requirements from a developer's viewpoint:

- Displaying information about developers: Apps and services should display information about the corresponding author(s) in the Service and App Marketplaces. Consumer of aps (end users) or services (other developers) should see this information. Examples may include the name, address or contact email of the author(s).
- Copyright: Developers should have the right to copyright their works (apps or services) for SIMPLI-CITY and SIMPLI-CITY should avoid that apps and services are submitted that infringe copyrights of others. This could be done by reporting copyright infringements to the SIMPLI-CITY team.
- Data access: Developer should be able to secure their account with a username/password or with another authentication approach to avoid that other developers misuse their data or access their data without sufficient access rights.

2.3 Possible Standards and Approaches

Privacy as already described above has different sides. On the one side, privacy can be understood as "user-trust" and on the other side in terms of legal compliance [Per09]. No user will use a cloud service that deals with personal information and has no or unsecure privacy settings.

Besides that, privacy is a human right (cp. European Convention on Human Rights). By the use of personal information, in Germany (cp. Federal Data Protection Act (FDPA)) and

Europe (cp. European Directive on Data Protection (EDDP)) data collection, processing or even transmission is very restricted by both laws FDPA and EDDP.

When transmitting personal data, a legal basis for transmission is needed. This legal basis can be an agreement by the user or by a balancing of interest. The last one is only possible, if the protection of interests is observed for the concerning person. This balancing of interest must be done for every individual case [Bun11].

For an easier way to transmit data, there are several exceptions for avoiding these strong restrictions of BDSG or EU-DSRL:

1. There are no personal information (§3 Abs. 1 BDSG).
2. There are personal information, but there is an adequate anonymization (§3 Abs. 6 BDSG).
3. There are personal information, but there is an adequate pseudonymization (§3 Abs. 6a BDSG).

In all other case, all restrictions and duties of both laws must be fulfilled. Before evaluating these exceptions, an overview is given where personal data within outsourced data-transmission or cloud computing can be found:

- The content of the data can be personal referred.
- The connection data, e.g. the point in time or the involved partner.
- By using a service, data is generated in this session.
- By selling services, data necessary for accounting and invoicing must be used.
- Offline-data, e.g. the contract or names and addresses.

On the first like, one can see that personal data exists in a lot of physical and logical stages. Now let us evaluate the exception for not applying the BDSG or EU-DSRL:

For SIMPLI-CITY exception 1 is not valid, because there are at any time personal information about the user or his behaviour in use. These data are collected and send to a server. However, processing and sending data in plaintext is not advisable and even not needed for SIMPLI-CITY. Briefly, anonymization or – where not possible - pseudonymization should be used for the collected data within SIMPLI-CITY.

In the following it is explained what anonymization and pseudonymization is in detail:

Anonymization means a complete removal of the direct personal reference. After this removal, no relation between the data and the corresponding person should be found. After an anonymization of the collected data, no information referring to the concerned user can be found. The quality of the anonymization is determined by a lot of different criteria. First of all, the point in time of the anonymization is a criteria, another criteria is the cardinal number of the whole sample, i.e, the measure of the number of elements of the set, and finally the chain-linking between the data, i.e., possibility of concatenation of individual transactions, [EEH+97]. Gottschalk differs three different ways of anonymization [Got02]:

- A change of the data by increasing the characteristics range.
- A change of the data by deleting variables.
- A change of the data by adding false data.

The primary drawback for anonymization is, that in some cases, a unique reference of a user has to be kept in place. For example, if data is stored and reused at a later time, services do need to be able to make use of a reference ID to identify the user that the data

belongs to. Nevertheless, for general statistics, e.g. for congestion forecast, anonymization can be used.

By using pseudonymization, the information about the person is replaced by a pseudonym/alias. This alias should be secret and without knowledge about details of this alias, a relation between the data and the concerned person should not be possible. It has to be mentioned that long time data observations and data mining techniques could result in enough knowledge to a third person to assign a user to an alias.

An easy way to achieve pseudonymization is the usage of unique IDs for each user (User-IDs). Besides using user-IDs as pseudonym, there are three "ways" of pseudonymization [EEH+97]:

- Self-generated pseudonymization by the user.
- Reference-based pseudonymization, kind of using user-IDs. The link is made by a reference list.
- By using a single-use-function, that generates the pseudonym.

2.4 Privacy Concept of SIMPLI-CITY

SIMPLI-CITY will carefully consider privacy aspects in the project. The consortium is convinced that a project cannot succeed in the market if it does not respect the privacy of its users. As such, SIMPLI-CITY will follow specific rules that are derived from the requirements of Section 2.2 and which are described in this subsection. It should be noted that there is an overlap between this section (2.4), Section 3.4 and Section 4.4. In order to avoid redundant text in this deliverable, each topic is only described once in the document – even if it touches multiple topics.

2.4.1 Personal User Data Access, Anonymization and Pseudonomization

SIMPLI-CITY allows users to benefit from a European Wide Service Platform (EWSP) as well as a range of apps consuming those services and making them available to end users. Those apps and services may be bound to personal data. For example, an app that will inform users about delays of events in their calendar needs to know personal information about the user in order to work properly. However, in many other cases, the transfer of personal data is not required. Apps that allow users to find the next free parking lot, do not need to know the name or address of the user. They may, however, require some information about the current context of the user – such as the current GPS location.

In this aspect, SIMPLI-CITY will make use of the following approach:

- For protecting privacy, SIMPLI-CITY will allow users to start using apps and services without having to identify themselves and without having to use their personal name unless it is required (Anonymous usage).
- If a user has to be identified, then SIMPLI-CITY will use a unique ID for a user, i.e. a UUID[1] (Pseudonomization). This allows apps and services to recognize a user without being able to identify his real identity.
  A single unique UUID could lead to a security problem in case of that UUID was used to correlate data from different services and then subvert (to some degree) the pseudonym. In order to avoid this, SIMPLI-CITY will return a separate UUID *per app* so that each app of the PMA will have an own UUID.

[1] Universally Unique Identifier

- If a user has to be identified, then the identity of a user will only be passed to those apps and services that require this information and only after user confirmation.

2.4.1.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- In general, no component should force users to enter private data unless required.
- The App Marketplace of SIMPLI-CITY will show a list of access rights that an app requires before installing an app to the PMA. This list will allow users to see if their personal data is required in order to make use of an app and if their personal data will leave the PMA or not. Technically, those access rights will be described by the app developer within the app manifest, which is described in D3.2.1 and D3.2.2. Users will have to view and acknowledge this list before installing an app.
- On the PMA, the Application Runtime Environment will only forward personal user data (e.g. the user name, etc.) to an app if the app has specified this right in their manifest. This will make sure that only apps will be able to access private user data that have been confirmed by the user during the installation process.
- When starting an app on the PMA for the first time, the Application Runtime Environment will generate a UUID for the user per app. This UUID will be used as a pseudonym for the user instead of her/his real data whenever possible[2].
- The Service Marketplace of SIMPLI-CITY will allow developers to see if they need to forward user data to the service and – if so – which user data is required to use the service. Technically, those access rights will be described by the service developer within the Author UI of the Service Marketplace using textual descriptions.
- The Context-Based Service Personalization will not offer any service that would allow apps or services to find out the identity of a user in an unauthorized way – e.g. by aggregating context data.
- Context data captured in the Context-Based Service Personalization will not be made available to third party systems (i.e. outside SIMPLI-CITY).
- Data access to third party systems (e.g. a Google calendar or a sensor) that require user credentials will work with solutions where they do not need to store the full user access data whenever possible. For example, the OAuth[3] standard may be used in many cases instead of storing the username/password of a user. Any user credential information that is required will only be stored in an encrypted way.

2.4.1.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Apps and Services that will be created within the use cases will be usable without the need to identify users whenever possible.
- Apps and Services developed within the use cases will not forward personal information of users to third party systems unless absolutely necessary[2]. If anonymous usage is not possible, pseudonomization will be used.

[2] The usage of a UUID instead of „real“ user data may not always be possible. For example, in case of accessing the personal calendar of a user, the real user name may be necessary – depending on the implementation and the calendar type.

[3] see [CC12], http://oauth.net

2.4.2 Data Isolation

Data that belongs to one user should under no circumstances be accessible by other users unless it is explicitly marked as public or unless it has been anonymized. In order to realize this, SIMPLI-CITY will ensure that apps can manage data in an isolated way – keeping user information separate from each other. Consequently, user specific data that is needed by an app and that is stored in the Cloud Storage needs to be bound to the user ID (UUID) for the corresponding app.

2.4.2.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Cloud Storage will provide isolated storage spaces that allow apps and services to store data. Data in one storage space will not be influenced by data in another storage space. This will support the isolation of data on a technical level.
- Apps and Services published in the Marketplace will be tested in order to ensure that data management is properly implemented and that users cannot access data from other users in an unauthorized way.
- User specific data coming from sensors that are related to a specific user (via the Sensor Abstraction and Interoperability Interfaces) will only be accessible with the correct user access token (see Section 3.4). This is also true for data from the Context-Based Service Personalization component.

2.4.2.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Use cases implemented by WP7 and WP8 will ensure that user data is isolated from each other whenever dealing with sensitive user data.
- If user data is meant to be shared with other users, then users will have the chance to stay anonymous if possible. For example, user data collected and aggregated in order to realize an eco-driving contest. In this case, the aggregated data will not allow users to detect the driving behavior or the environmental impact of a specific other user. Instead, only aggregated anonymized data will be shown to the user.

2.4.3 Data Reduction and Data Economy

One of the big benefits of today's technology is the possibility to handle large amounts of data with low costs. This has, however, led to the fact that many data is collected and kept even if it is not necessary for fulfilling the purpose of a task. SIMPLI-CITY will advise app and service developers to avoid this unnecessary data collection. It will only ask its users for data that is required in order to make use of the services or for data that is at least helpful for raising the comfort of the driver. As such, SIMPLI-CITY will reduce the amount of permanently stored data whenever possible hence keeping the data footprint of its users as low as possible.

2.4.3.1 Concrete Action Points for WP4, WP5 and WP6

Which data that is stored in SIMPLI-CITY is decided by the services and apps, which will be implemented on top of the technical components of SIMPLI-CITY. As such, there is no direct impact of this concept to WP4-6.

2.4.3.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Apps and services that are created within the use cases of SIMPLI-CITY will be created with data reduction in mind. User data will only be captured if necessary.
- User specific data that has to be captured, will be stored temporarily if possible in order to avoid a long and permanent user data storage. User specific data will be deleted if a user closes his/her account.

2.4.4 Data Location

SIMPLI-CITY is service-driven, meaning that apps will strongly make use of backend services running in the SIMPLI-CITY Service Runtime Environment. This will keep apps lightweight with the main purpose to consume services and to interact with the user in a multimodal way. This means that user data will in many cases be needed at the server side in order to use it within a service. As such, data may either be passed to a service (via the Application Runtime Environment and the Service Runtime Environment) or to the Cloud-based Information Infrastructure.

However, despite this strategic decision to build a service-driven system, not all user data needs to be transferred to a service. For example, users may configure an app for a specific behavior or an app may need vehicle sensor data for its local functionality. In those cases, any data that is not needed at the server side should not leave the PMA.

This will be realized for transient as well as for permanent data. Only data that is needed at the server side will be stored at the Cloud Storage while local data will be stored locally at the PMA inside the Local Key Storage.

2.4.4.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Cloud-based Information Infrastructure that is provided within task T4.2 of the project will realize an additional PMA-based storage (called the Local Key Storage). This storage will allow a simplistic way of storing key-value-type data on the PMA. Data that is stored in this Local Key Storage will not leave the PMA at any time[4].
- Data that is not needed outside the PMA will not leave the PMA. The PMA-Based Sensor Abstraction will therefore provide an additional local interface allowing apps to query local vehicle sensors without having to send them to the server side first unless services have to access it.

2.4.4.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Use cases developed within WP7 and WP8 will keep user settings on the PMA. Especially, payment data similar sensitive user information will not leave the PMA and will not be sent to any external service unless absolutely necessary.

[4] SIMPLI-CITY will provide a platform allowing developers to write new apps and services. The consortium cannot influence foreign code of external developers and cannot avoid that they are sending user data to external systems. However, the project will perform an app/service review and check each app/service before adding it to the App and Service Marketplaces.

- If personal data is transferred, apps and services will make use of an encrypted data connection (see Section 3.4)

2.4.5 App and Service Review

SIMPLI-CITY wants to raise the comfort of the driver by giving her/him a set of non-distracting apps that increase the driving experience. As such, it is important that apps and services that are available via the PMA are

- Stable and bug free,
- Usable in a multimodal way,
- Privacy aware[5],
- In sync with the description in the Service and App Marketplaces and
- Of high quality.

In order to achieve this, apps and services will be reviewed before they will be published in the Service and App Marketplaces.

2.4.5.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Service and App Marketplaces will provide a review system, which has been described in deliverable D3.1. It allows authors to submit an app/service via the Author UI. Reviewers of the Service and Apps Marketplaces will be able to accept or reject submissions.
- Users will be able to report an app or a service to the review team. This will trigger an additional review process allowing the review team to deactivate the service/app from the Service and App Marketplaces is necessary.
- Reviewers may mark a service as “dangerous”. The Service Runtime Environment will in this case block any additional request to this service so that the service will not be invoked anymore.

2.4.5.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- All apps and services that will be developed in the use cases of SIMPLI-CITY will undergo the same review process that apps and services from third parties would do. This allows the consortium to test the review process and it also to minimize bugs and problems in the use case implementations.

2.4.6 Developer Verification

For achieving a trustful environment, it is important for users to know that apps and services are provided by trustful and reliable developers. The achievement of creating a trustful environment will therefore be a key factor for the acceptance of SIMPLI-CITY within Europe. Therefore, SIMPLI-CITY will allow developers to get themselves listed as a “verified developer”. Developers can offer apps and services without a verification in order to allow them a quick start with SIMPLI-CITY. However, verified developers will be marked with a quality sign in their profile allowing app and service users to see that they have

[5] by following the concepts of this Section 2.4 and of Sections 3.4 and 4.4

been verified. Verification in this context means the verification of the identity meaning the check of the real name and the location of a developer.

2.4.6.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Author UI of the Service and App Marketplaces will allow developers to submit verification documents. Those will be considered during the app/service review process.
- The Service and App Marketplaces will highlight verified developers with a ”verified” seal.

2.4.6.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- In order to provide a good example, all developers involved in the app and service creations of the use cases will be verified by SIMPLI-CITY.

2.4.7 User Feedback

The great success of the Apple AppStore and the Google Play Market have shown the importance of user-generated feedback. SIMPLI-CITY will learn from this and will allow users to rate apps. Additionally, it will allow developers to rate the services that they are consuming.

2.4.7.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Service and App Marketplaces will allow end users to rate apps in the PMA.
- The web UI of the Service and App Marketplaces will allow developers to rate services that they are consuming in their apps.

2.4.7.2 Concrete Action Points for WP7 and WP8

This concept has no direct impact to the use cases of SIMPLI-CITY.

3 Security Aspects

3.1 Scope

The protection of information and IT systems towards unauthorized access or modification, whether during storage, processing, or transit is one of the most important responsibility of information security in IT systems. Information security includes those measures necessary to detect, document, and counter such threats [US05].

3.1.1 Definition

Information security is developed to preserve the three information characteristics: confidentiality, integrity and availability. Those are often referred to as the “CIA” triad. The goal of SIMPLI-CITY in this domain is to ensure the preservation of those aspects and to use appropriate technologies to protect the SIMPLI-CITY infrastructure from malicious intentions. For example, in the context of SIMPLI-CITY these intentions may be

- Unauthorized access: This could be the access to information stored in the Cloud Storage from users or developers who should not have access to data. This may, e.g., be the access to credit card information used for the payment process in the Service and App Marketplaces.
- Misuse of information stored in SIMPLI-CITY: This may be the usage of passwords or usernames for external services, e.g. Facebook.
- Disclosure of private information: Within SIMPLI-CITY this may be a publication of personal calendar entries of a user.
- Disruption or unauthorized modification of content: This may be the deletion of a user profile by another – unauthorized – party.

To prevent those aspects, SIMPLI-CITY will follow strict rules on how to handle user-based information (see Sections 3.4, 3.5 and 3.6). For example, the misuse of Facebook login data may be prevented by either not storing the Facebook data at all or by storing it on the PMA side only (as opposed to storing it in the Cloud Storage) or by using a more secure way of accessing Facebook (e.g. OAuth instead of username/password).

The most important task of information security within SIMPLI-CITY is to handle risk management whereby all above-mentioned acts could represent a threat to the CIA triad. Especially, sensitive data must be protected by the SIMPLI-CITY components. For example, a third party could intercept within the transmission procedure and change, modify or steal data before it reaches the recipient. Since most of the transmission systems are unsecured (such as Internet) a good cryptography could help to make the stolen data unreadable for the hacker.

3.1.2 Key Elements

The CIA objectives - confidentiality, integrity and availability - can be seen as generic terms for security [ITS91, SCG12].

- Confidentiality:
  - “Prevention of the unauthorised disclosure of information“ [ITS91].
  - A service, classified as confidential, offers a high level of assurance that no unauthorized subjects are involved. In general, access control is the way of protecting [SCG12].
- Integrity:
  - “Prevention of the unauthorised modification of information“ [ITS91].
  - Integrity grants that the data is unaltered from the original state [SCG12].
- Availability:
  - “Prevention of the unauthorised withholding of information or resources“ [ITS91].
  - Availability grants authorized subjects a timely and uninterrupted access [SCG12].

The CIA triad encompasses the most important characteristics of information security for SIMPLI-CITY. Other characteristics such as Accountability have sometimes been proposed for addition as discussed in e.g. [Pe11]. Further aspects such as legal aspects are becoming a key consideration for practical security installations and will be discussed in more detail in Section 4 of this deliverable.

In 1992 and – as a revised version – in 2002 the Organisation for Economic Co-operation and Development (OECD) has created Guidelines for the Security of Information Systems and Networks. Within those guidelines, the OECD has proposed the nine principles: Awareness, Responsibility, Response, Ethics, Democracy, Risk Assessment, Security Design and Implementation, Security Management, and Reassessment [OE02].

In 2002, an alternative model for the classic CIA triad was proposed by Donn Parker and widely accepted by many IT security experts. Those elements are confidentiality, possession, integrity, authenticity, availability, and utility and are known as the “Parkerian hexad“ [FK07], [CDG13].

- Confidentiality: Protecting the exposure of sensitive information to unauthorized parties. Confidentiality is a point where security with privacy intersects and it is important for maintaining the privacy of information hold by ICT systems such as SIMPLI-CITY. It is especially important for direct critical data such as payment information (e.g. credit card details in the SIMPLI-CITY Service and App Marketplaces) or indirect data which is not stored explicitly by SIMPLI-CITY but which may be indirectly extracted. For example, this could be health related information from the user calendar (e.g. a doctoral appointment).
- Possession: Referring to the control about data. This often goes hand-in-hand with confidentiality. Within the project, this point could be achieved by allowing users to delete their account and their data when leaving SIMPLI-CITY.
- Integrity: Ensuring the correctness and consistency of data during its whole life-cycle. Within SIMPLI-CITY this means that no one should modify or manipulate data unless being authorized to do so. For example, integrity is breached when information was modified or manipulated during its transmission between the PMA and the server side.

- Availability: Meaning that required information has to be available when it is needed. This could be interpreted as all involved systems in a transmission must be running correctly. High availability systems are designed for preventing service interrupting due to power outages, hardware crashes or denial-of-service attacks.
- Authenticity: Ensuring that data that has been transferred between two or several systems (such as transactions) are authentic. Another important task of authenticity is to check that each one of both parties (sender and receiver) involved in a transmission is the party he is posing as. SIMPLI-CITY will provide authentication features (as described in this deliverable).
- Utility: Meaning the usefulness of data that has been secured. Within SIMPLI-CITY, data that has been encrypted and stored in the Cloud Storage may be considered as being secure and may fulfil all characteristics above. However, this data is – of course – of no use if the key for decryption is not known.

Apart from the six elements above, the “non-repudiation” often plays an important role in IT security implying the intention to fulfil their obligations to a contract [CDG13]. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction. It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is at its core a legal concept transcending the realm of technology (see [CDG13] as well as Section 4 of this deliverable).

3.2 Impact to SIMPLI-CITY and Requirements to the Project

This section contains requirements to the SIMPLI-CITY project based on security considerations. These requirements can be seen as an extension of the user requirements defined in deliverable D2.3 of the project. They will be used as a base for the project in general and for the prototype implementations of WP4-8 in specific. This subsection is based on the security considerations of deliverable D3.1. They are equally important from both viewpoints - end users and developers:

- Secure data channel: All data transferred between different components of SIMPLI-CITY should preferably be transferred between the different systems using a secure communication channel.
  In SIMPLI-CITY, this concerns apps (e.g. for the consumption of services via the Application Runtime Environment and Service Runtime Environment), components (e.g. for offering the Cloud Storage interface to other components) and services (e.g. for service-to-service communication).
  “Secure” in this aspect means that the data channel that is used to send and receive data should not be openly accessible by third parties. However, technically SIMPLI-CITY will be using the internet, which means that the channel itself is not completely isolated from third parties. For example, the internet provider will have to be in the middle of the Application Runtime Environment and Service Runtime Environment.
  However, in some cases, the communication channel will be isolated from third parties, e.g. in case of communication between the Application Runtime Environment and the PMA-based Sensor Abstraction, which will both be located inside the PMA and will therefore not need to transfer data via the internet.
  Similarly, the communication between two services that are fully deployed to the

Service Runtime Environment will not need to transfer any data via an insecure communication channel.

- Encrypted data transmission: As discussed above, SIMPLI-CITY will in some cases use the internet to exchange data, e.g. in order to invoke external services or in order to perform a server-based speech recognition (see deliverable D3.2.2). In those cases, SIMPLI-CITY should use encryption to transfer data whenever possible.
  Please note that it may not always possible to enforce this if external – third party – services are consumed. However, SIMPLI-CITY should ensure that encryption is at the very least used whenever sensitive user data is processed such as credit card or other payment related data.
- End-to-End security: End-to-End security relies on protocols and mechanisms that are implemented exclusively on the endpoints of a connection [Be09]. The most typical example is an HTTPS connection (based, for example, on Transport Layer Security (TLS)) to a web server; IP Security (IPsec) can also be used for End-to-End security, as was initially proposed as a default connection mechanism for IPv6 [Be09]. Within SIMPLI-CITY, data may be "routed" via different components of the project as it is described in the sequence diagrams of each component in deliverable D3.2.1.
  For example, an app which finds free parking lots may use the Application Runtime Environment to perform a service call via the Service Runtime Environment. The Service Runtime Environment will invoke the corresponding parking service which may again rely on one or more external service calls. SIMPLI-CITY may encrypt the communication between all involved parties separately. In this case, each communication step would be encrypted and for most scenarios (e.g. for finding free parking lots), this approach is considered to be sufficient for SIMPLI-CITY. However, different services and components between the data service and the app would be able to access the data. In some cases, this may not be appropriate. For example, an app that may allow the user to request his current bank account status handles sensitive information. As such, this app and the final bank service should use End-to-End security encrypting the data at the app side and allowing only the final bank service to decrypt it. No component in the middle of the communication process should have the possibility to decrypt or modify the data.
- Authentication towards the system: Both end users and developers have to be authenticated before accessing any sensitive part of SIMPLI-CITY. For example a developer may have to enter his credentials to have possibility to add an app or a service to the marketplaces of task T5.4. End users need to authenticate themselves whenever accessing personal information such as their settings, their driving history or their payment data. For example, end users may be authenticated through a username and a password.
  End users: The requirement for authentication does not mean that end users need to be identified as a person. They may still use SIMPLI-CITY anonymously if they prefer but it is important that each user has a unique identifier to secure his/her from access by other users.
  Developers: Third parties may use SIMPLI-CITY to develop and publish new apps and services. As described in deliverable D3.1, apps and services will only be published after reviewing. Developers may add, update or delete their apps and services. As such, developers also need to authenticate. However, in contrast to end users that use the PMA, developers will need to also add their personal data

(e.g. organization name, contact data, etc.) for allowing reviewers to contact the developers and also for increasing trust by displaying the real name in the developer profile at the marketplace.

3.3 Possible Standards and Approaches

Security is a very large aspect, because every piece should be safe on its own and, additionally, the whole system should be safe. Security begins at local geographical security (buildings, staff, etc.) and ends up in software security of programs or protocols.

This section only concentrates on the operational, platform and application security, which means, the physical security is out of the scope, e.g. door security, fire security or safety guards.

As introduction to the virtual security typical attack methods are listed to illustrate the range of possible attack situations.

Typical methods of attack are:

- Brute-force and dictionary attacks: In this case, the attacker tries to get access by trying every possible combination. This could be letter by letter or in case of dictionary attacks word by word [SCG12]. This is a very promising attack, because at some time, the attacker will get access to the system. It is advisable to limit the amount of wrong inputs to a system to ensure, that trying out is not an option. Furthermore, long and strong passwords are advisable. Mixing up capital letter with numbers and special characters increase the security.
- Denial-of-service attacks: A Denial of Service (DoS) attack or Distributed Denial of Service (DDoS) attack describes an attack to a network or singe computer. The attacker tries to make his target unavailable for other users [SCG12]. In detail, the attacker uses flooding-attacks like SYN-Flooding to get an amount of connections that are too much for the server. This means, the host ends up in a system crash or in a CPU load of 100% [SCG12]. In contrast to other attacks, there is no need for passwords. This means, everyone can start a denial-of-service-attack.
- Malware: Malware is a kind of software that infects and manipulates the software in an unwanted way. This can be viruses, worms, trojans, spyware, and more. By using malware, the attacker tries to get access by bringing up the user to click/install on this kind of software. After clicking/installing this malware, the attacker gets access to software or devices. Besides getting access to the system, it is also possible to steal passwords or data from the user.
- Man-in-the-middle- attacks: A man-in-the-middle-attack is a way to attack a system. The attacker tries to get between the sender and receiver of an information [SCG12]. This attack is also possible in a cloud-environment. Here, the cloud user sends his information to a cloud, owned by the attacker. There are some ways to protect against a man-in-the-middle-attack:
    - By using encryption and a authentication for both sides.
    - By using Integrity Protection. Besides the information, the data-packages includes a Message Authentication Code.
    - By using a second authentication via a second communication channel. Mobile TAN is a popular example for this solution.
- Sniffers: Sniffing (sometime also called snooping attack [SCG12]) is a way for analysing a network. Here, the attacker tries to “sniff” (capture) over the network.

Sniffers are often used for normal network analysis or intrusion detection systems but can also be used for attacking a system. In all cases, the network traffic is scanned and evaluated [SCG12]. To protect against network sniffers, one can use secure communication channel. The attacker needs access to the network, if it is physically not possible to get access to the communication, sniffing is not possible. A second way to protect against sniffing is encryption.

- Spamming: Spamming comes from Spam or Junk. By using the term spamming, sending unwanted information to a system is mend [SCG12]. The content of this Spam has often a commercial character but can also imply malware or viruses. To protect against spamming, filter systems can be used.
- Spoofing: Spoofing means a manipulation or fakery to an information system. It is used to hide the real identity. There a different ways of spoofing, some are: DHCP-Spoofing, DNS-Spoofing, IP-Spoofing or GPS-Spoofing, which means sending a fake position to a GPS-receiver. In summary, by spoofing is an attack by pretending to be someone other [SCG12].

Several protection mechanisms may be named for applying the Security Management Concepts and Principles described in Section 3.1.2:

- Password Security: The most common way to get access to a restricted area is the use of a password. When attackers try to get access to this area, they can employ the methods, mentioned above. Thus a strong password is recommended. There are three common ways for *getting* a password: password guessing attacks where arbitrary Byte combinations are used, dictionary attacks where a data base of possible password is used and social-engineering attacks where the users are forced to sell out their passwords, or an attacker spies out the passwords. *Social-engineering* is the most effective tool [SCG12]. By *social-engineering* This attack often works because the attacker pretends to be someone other, e.g. a system administrator [SCG12].
- Layering: With layering (or *defense in depth* [SCG12]), several protection mechanisms are used in a series, e.g., firewalls and routers, web and application layers of protection [SCG12]. Generally, the use of one protection mechanism cannot protect against all threats. The use of a collection of protection mechanism is advisable [SCG12].

Securing the Communication Channel

By sending data through a network, a Virtual Private Network (VPN) is a safe and practical solution. The classical definition of VPN is a connection from a network to another network [SCG12]. By using this connection, the transfer of data can be encrypted. In an “unsecure” environment, the user can send critical data over the encrypted VPN-channel to the receiver. There are three sorts of VPNs: Site-to-Site for connection several users over the internet or End-to-Site for connecting a client to an end network. The last one is an End-to-End connection between a client and a Server.

VPN is based on several protocols that are showed in the following:

- IPsec (Internet Protocol Security) is a protocol collection for building up a secure connection over a potential unsecure connection. By using IPsec, encryption and data-integrity can be used. IPsec consists of two components:
    - Authentication Header (AH) provides integrity, authentication and nonrepudiation [SCG12].

    - Encapsulating Security Payload (ESP) provides encryption and a limited authentication [SCG12].
- Transport Layer Security (TLS) better known on his predecessor Secure Sockets Layer (SSL) is an encryption protocol for securing a connection over the Internet. By using TLS-encryption, it is used with HTTPS (Hypertext Transfer Protocol Secure), a protocol for secure transferring data over the World Wide Web. Besides HTTPS, by using TLS a certificate-based authentication is made by the server, in prevention e.g. man-in-the-middle-attacks.
- Layer 2 Forwarding (L2F) is a protocol for non-encrypted tunneling mechanism [SCG12].
- Point-to-Point Tunneling Protocol (PPTP) is a protocol based on the Internet protocol for enabling a VPN connection. PPTP builds up a tunnel for the VPN connection.
- Layer 2 Tunneling Protocol (L2TP) is a protocol for the second layer of the OSI-Model. L2TP exchanges frames on the second layer for enabling the VPN connection. These frames are packed by the LAC (Access Concentrator) and send to the LNS (Network Server). The frames are unpacked and send to the receiver. L2TP combines PPTP and L2F [SCG12].For better understanding, the a short version of the OSI-Model will be presented in the following: The OSI-Model divides a network into seven layers. These layers are (1-Phyical, 2-DataLink, 3-Network, 4-Transport, 5-Session, 6-Presentation, 7-Application) [SCG12]. Each layer has its own task for exchanging data within the network. Data is at the sender side handed over from the top layer to the subjacent layer until it is transmitted via the physical layer. Afterwards at the receiver side the data is handed over from the lowest layer up to the application, in each case by the respective layer to the overlying layer.

In the following, solutions to protect critical information is given:

- Encryption of Critical Information: Encryption is a way for changing the visible information of data. This means, by using an encryption method, information cannot be read without knowing the encryption key. The key is generated by the encryption method. In other words, the encryption is a function with a key as parameter for changing information. There are two different ways of encryption: By using *Symmetric-Key Algorithms*, the key for encryption and decryption is the same [SCG12]. This means, a secure exchange of the key is necessary. By using *Asymmetric-Key Algorithms* (also called public key algorithms [SCG12]), the key for encryption and decryptions are not the same. One key is public (this key is used for the encryption) and the other one is a private key (used for the decryption) [SCG12]. By following encryption news, a new way for processing encrypted data is on the way: Homomorphic Encryption [Gen09]. This way of encryption allows the service provider to process the data without an encryption. This solution seems very useful and practical mainly for the use of Cloud computing. Here, the user has no special need for trusting the Cloud provider.
- Data Classification: Classification of data or labelling/marking of data is an important way for securing sensible information. Without marking critical information as critical, there cannot be an adequate level of security. Giving all data the same level of protection is inefficient, because some data need more security than others [SCG12]. In other words, the cost for more sensitive data increases with the increase of security [SCG12]. Besides addressing how data is stored or moved, data classification also addresses the way it should be removed. In praxis, two

schemes of data classification are used: government/military classification (Top secret, Secret, Confidential, Sensitive but unclassified, Unclassified) and commercial business/private sector classification (Confidential, Private, Sensitive, Public) [SCG12].

- Web Application Security: Some parts of SIMPLI-CITY are connected over the Internet using web applications. For showing how insecure a not well-programmed application can be, two common web attacks will be presented [SCG12].
  - Cross-Site-Scripting (XSS) attacks are used by user inputs. Some web-applications ask for input. If the input is a script instead of e.g. name, the server could acts in a different way [SCG12]. To protect against XSS, input validation is very advisable [SCG12]. Inputs that contain "<SCRIPT>" should never be allowed [SCG12].
  - SQL-Injection attacks are also some kind of input to a web application. SQL-Injections are used to get access to the underlying database [SCG12]. To protect against SQL-Injections an input validation, like the one by XSS, should be done. Another way to protect against SQL-Injection is limiting the privileges [SCG12].

3.4 Security Concept of SIMPLI-CITY

SIMPLI-CITY will handle security aspects in order to guarantee a trustful and secure environment. As such, SIMPLI-CITY will follow specific rules that are derived from the requirements of Section 3.2 and which are described in this subsection.

Due to the concepts outlined in Section 2, it should be noted that the transfer of private and sensitive information between the PMA and the server side will be limited. Nevertheless, sensitive information may in some cases have to be transferred. For example, a service that informs users about the status of their train connection has to access data from the personal calendar and may have to transfer it to the PMA. Therefore security is a topic that cannot be neglected in the project.

It should, however, also be noted that there is a trade-off between creating a very secure system on the one hand and a very user friendly system on the other hand. For example, forcing the user to enter a username and password at each PMA start will certainly be a way of improving security of the system but it will also be annoying for the user and may therefore lower the usability. -- The concepts outlined below will consider those aspects and have been created with a balance between security and usability/comfort.

3.4.1 Registration and Authentication of Users

The PMA will allow users to make use of apps inside the PMA. Since apps are based on services, users will indirectly make use of services as well, even if they will not notice this fact.

SIMPLI-CITY will in many cases use services that do not need to know user details. For example, a service that finds the next parking place will require the GPS location of the user but it does not need to know any other use data[6]. In other cases, aps or services may, however, have to access more specific user data such as the name or the address of

[6] This is a simplified example because in some cases more sure data may be needed, e.g. in order to know if a user is disabled or in order to know the type of the user car (electric, gas driven, etc.)

the user. Additionally, apps or services may need to store specific user data such as historical driving information or user settings.

For ensuring this, data should only be accessible by the user that the data belongs to. As such, SIMPLI-CITY – and more precisely the PMA – will require users to register when they start the PMA for the first time. This registration process will allow users to define a way of authenticating themselves when they use the PMA. The PMA will ask for this authentication each time it is switched on. For example, users may specify a password (PIN) which needs to be typed each time the PMA is started. Users may choose between different ways of authenticating themselves. In addition to typing in a PIN code, they could choose to use gestures or even face or voice recognition or even via OpenID support[7]. However, during the prototype stage of the project, SIMPLI-CITY will only realize the PIN code approach, while other approaches may be added during exploitation and commercialization of the project results after project ending.

SIMPLI-CITY will allow users to remember the PIN code, meaning that users do not need to authenticate. This will be offered to increase the usability but users will receive a big and unambiguous warning to make them aware that this will make it easy for third parties to misuse their account if the PMA device is stolen. Therefore SIMPLI-CITY will follow the approach of modern cell phones and will even make use of the integrated Android functionalities for realizing this concept, which means that WP4-6 do not need to manually implement this concept.

When users register themselves with the PMA, a UUID will be generated for them. This UUID is a 32 character code which uniquely identified a user and it is app specific (i.e. each app will receive a separate UUID)[8]. It may be used as a pseudonym for the real user data whenever a user ID is needed in SIMPLI-CITY and will therefore implement the pseudonomization approach described in Section 2.

3.4.1.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Application Runtime Environment will implement a registration process, which will be launched the first time the PMA is switched on. This registration process will allow the user to configure the PMA and to choose different settings. Towards the end of the registration process, the Application Runtime Environment will automatically generate a UUID for the user and per app.
- Services of WP4 and WP5 will work without having to identify a user whenever possible. In case that a user needs to be identified, the UUID will be accepted as a placeholder to uniquely identifying a user. Full user details will not be required by any of the WP4 and WP5 components.

[7] http://openid.net

[8] As described in section 2, the UUID will be app and user specific, meaning that one PMA may contain many different UUIDs. However, from an app perspective, the UUID for the current user is always the same. Considering this, messages will be signed with a unique user specific UUID from an app and from a service perspective. As such, whenever the UUID is mentioned in this document, the current unique UUID is meant, i.e. the UUID of the current user under her/his current app context.

3.4.1.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Use case implementations of apps will use the UUID that is provided by the Application Runtime Environment via a “getUDDI()” method which will return the UUID for the current user and for the invoking app.
- Use case implementations of services will accept the UUID as a parameter which will be passed to them during service invocation. The Application Runtime Environment will pass t to the Service Runtime Environment as part of the service call that has been issues by the PMA apps.

3.4.2 Authentication of Developers

Developers may create new apps and services for SIMPLI-CITY. Those apps and services will be offered in the Service and App Marketplaces of the project and may therefore be consumed by other developers or end users respectively. Apps and services may be free of charge or they may be commercial. As such, developers need to be able to protect their account as other developers would otherwise be able to access or modify their work.

In order to address this, SIMPLI-CITY will follow the following approach:

- Service and App Developers can register at the Service and App Marketplaces via the integrated web UI (Author UI). This registration will create a permanent account for them.
- The authentication of developers after registering is performed using OpenID (see [CC12], http://openid.net). This allows developers to make use of their existing OpenID account or to create a new one for SIMPLI-CITY if needed.
- Each time, a developer access his/her account, an authentication via the OpenID provider is required. This will ensure that their account is inaccessible by third parties.

3.4.2.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Services and App Marketplaces (the Author UI) will support developers to register themselves and to protect their account. Full OpenID support will be provided for realizing the login procedure.
- The Application Design Studio and the Service Development API will not directly modify the market data. Instead, they will export a bundle (WAR or JAR) which will be uploaded by the developers inside their account.
- The Service Registry will receive information about which service version to publish from the service side of the Services and App Marketplaces.

3.4.2.2 Concrete Action Points for WP7 and 8

This concept has no direct impact to the use cases of WP7 and 8 except that all SIMPLI-CITY team members need to create a developer account if they are involved in the implementation phase of WP7 or WP8.

3.4.3 Message Signing

SIMPLI-CITY is a highly distributed system consisting of many different services at many locations as well as many different devices (PMAs) from many users. A user may use an app which invokes services to gather information. This invocation is usually performed by forwarding the command to invoke a service to the Service Runtime Environment via the Application Runtime Environment. The Service Runtime Environment then forwards the request to the corresponding service. Along this route, the app may ask the Application Runtime Environment to pass the UUID to the service. Finally, the service which is invoked will receive the request from the Service Runtime Environment with all information that is needed to invoke the service including the UUID of the user (if required).

In many cases, this process will be sufficient. However, in some situations, the service may want to verify that a message of a specific UUID really comes from this user and that the message was not manipulated. This is especially important for any service call that queried sensitive user information (e.g. payment information or the request of user details).

For those purposes, SIMPLI-CITY will support the use of digital signatures to sign a message. From a technical viewpoint, RSA[9] will be used to implement this functionality [MAO96], [Sc08].

Within the project, this concept will be realized as follows:

1. As soon as a user registers at the PMA, a UUID will be created for the user as described in Section 2.4.1. This will form the base for generating UUIDs for each app that is or will be installed on the PMA. The UUID is then used to generate a private and a public key using the RSA approach per user and app[10].
2. The public key of the user will be published at the T4.2 Cloud Storage in a public accessible Bucket. This will allow any component, any service and any app to receive the public key via a simple REST call as described in D3.2.2 at Section 6.2.5. More precisely, the call will be[11]:
   http://simpli-city.eu/api/cs/keys/crud/
   with
   { “uuid”:”508cd480-1623-11e3-8ffd-0800200c9a66” }
   as parameter in the request. Alternatively, the following URL will work as well:
   http://simpli-city.eu/api/cs/keys/crud/uuid/508cd480-1623-11e3-8ffd-0800200c9a66
3. Additionally, the public key of a user will also be included in the header of each signed message that is sent between the PMA and the server side – or more precisely, between the Application Runtime Environment and the Service Runtime Environment.

It is important to note that apps *may* choose to sign service calls but this is optional. As service invocation is performed via the Application Runtime Environment, apps need to use an overloaded method with a Boolean parameter indicating whether they want their message to be signed. If message signature if required, the Message Handler

[9] An encryption algorithm named after Rivest, Shamir und Adleman for asymmetric encryption of data.
[10] As described in section 2, the UUID will be app and user specific, meaning that one PMA may contain many different UUIDs. However, from an app perspective, the UUID for the current user is always the same. Considering this, messages will be signed with a unique user specific UUID from an app and from a service perspective. As such, whenever the UUID is mentioned in this document, the current unique UUID is meant, i.e. the UUID of the current user under her/his current app context.
[11] Please note that the UUID (508cd...) will of course differ from user and only acts as an example value.

subcomponent of the Application Runtime Environment (see Section 8.1 of D3.2.2) will generate the service call message. Before sending it to the Service Runtime Environment, it will calculate the hash of this message and then sign it with the private key of the user using RSA. For generating the hash value, SHA-256 will be used (see [Sc08]).

Whenever a signed service call is received by the Service Runtime Environment, the Service Runtime Environment will automatically check it for validity by verifying the message signature. This will be performed using the public key of the UUID with the RSA approach and by comparing the SHA-256 hash of the message with the decrypted SHA-256 hash.

- A service call that is not valid will not be processed. Instead, the Service Runtime Environment will respond with an error return value to the Application Runtime Environment as described in D3.2.2 at Section 7.1.4 in the error handling subsection.
- A service call that has been verified successfully will be processed normally and the corresponding service will be invoked. The original message – including the RSA signed hash – will be passed to the service together with the UUID of the PMA user.

This approach allows each service to verify the service call, hence allowing the service to verify that the message sender is really identical to the UUID that has been passed to the service.

Analogous to the process above, services may also send messages to apps – either as part of a response to a service call or by using the Push Service of the Application Runtime Environment (see deliverable D3.2.1). As such, each SIMPLI-CITY component and each service may also make use of the approach described above by applying it in the other direction: Services and components may publish their public key in the key Bucket of the Cloud Storage and will also send it together with any signed message. They may then sign their message using the SHA-256 based hash of the message and using RSA for signing it with their private key. The Service Runtime Environment will pass this signed message to the Application Runtime Environment. If a message is not valid, the Message Handler component will respond with an error code and will not process the message. Otherwise, the message is processed and the original message – including the signature and the public key – is passed to the application together with the URI of the sender (i.e. the URI of the service or the URI of the SIMPLI-CITY component). This will allow an app to verify that the sender of the message is really the component/service that has been passed to the app.

3.4.3.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Cloud Storage has to provide a public Bucket for publishing keys. This Bucket will only allow the owner of a key to change its data entry but will allow everyone to add new public keys for themselves.
- The Application Runtime Environment will generate a public and a private key during the registration of a user and will send the public key to the Cloud Storage Bucket.
- The Application Runtime Environment will support the signing of messages with a simple Boolean flag. This will allow apps an easy to use way to sign their

messages. The Application Runtime Environment will automatically handle the signature of the message before sending it to the Service Runtime Environment.
- The Service Runtime Environment and the Application Runtime Environment will both check each signed message on arrival and will reject invalid messages. Messages that have a valid signature will be processed and forwarded to the destination.
- The Service Registry will generate a public and a private key during the registration of a service and will send the public key to the Cloud Storage Bucket.
- Each SIMPLI-CITY component will generate a public and a private key during the project and will send the public key to the Cloud Storage Bucket.
- The Application Design Studio and the Service Development API should at least provide one example on how to make use of this feature in order to ease the understanding of it.

3.4.3.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Apps and services which will be developed in the course of the use cases have to make use of the signing feature of SIMPLI-CITY for any service call that contains sensitive or user specific data. For example, a call that contains an entry of the personal user calendar has to be performed using the signing approach.
- Apps and services which will be developed in the course of the use cases may optionally make use of the signing feature of SIMPLI-CITY for any call that does not contain user specific or sensitive data. For example, a service call to receive the current weather conditions does not necessarily need to be performed using the message signing feature of SIMPLI-CITY.

3.4.4 Authorization

In some cases, SIMPLI-CITY will make use of external systems to provide apps and services that will raise the comfort of the user. The following examples show some situations where this is useful:

- Access to the Facebook account of a user may be used to publish the current eco footprint of a use while she/he is travelling.
- Access to the G+ account of a user may be used to publish messages that a user might speak to an app while driving.
- Access to Twitter may be used to read out posts of friends to the driver or to monitor status messages.
- Access to the Flickr account of a user may be used to display photos of the surrounding to the user (only suitable for non-car users in order to avoid the distraction of the driver).

In those cases, users need to authorize apps of the PMA and services/components of SIMPLI-CITY to access this data. This will be performed using the OAuth standard (see [CC12], http://oauth.net).

Within the project, this concept will be realized as follows:

1. Users may make use of apps that require access to a service from an external system that supports OAuth such as a Twitter service for porting tweets.

It should be noticed that this also covers indirect access meaning that an app may not necessarily have to directly interact with the external service. Instead, it could use the Application Runtime Environment and the Service Runtime Environment to call a backend service which then interacts with the external service.

2. As soon as the access to the external service is performed for the first time, the user will be displayed the OAuth dialog of the service provider. This will happen in a web view within the Application Runtime Environment. This dialog is used to authorize the access of the app/service to the corresponding external service provider.
3. The result of the OAuth process is an access token, which the app/service will pass to the external service provider on any request. This access token acts as a placeholder for the user allowing the app/service to make use of his/her data or to interact on his/her behalf by using the API of the external service provider.
   It should be noted that OAuth allows users to reject/cancel this authorization at any time using the service providers' web interface hence blocking SIMPLI-CITY from accessing his/her data. In this case, the user will have to re-authorize the app/service.

OAuth may also be used by services that are running within the SIMPLI-CITY Service Runtime Environment. For those cases, the process above is exactly the same so that there is no difference between authorizing against an external service (e.g. Twitter) or an internal service running within SIMPLI-CITY.

3.4.4.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Application Runtime Environment will support the usage of OAuth and will perform the authorization process via a web view within the PMA allowing users to authorize the access to external services.
- The Service Runtime Environment will support the deployment of services that consume external services that are protected with OAuth such as Twitter or Flickr.
- The Service Runtime Environment will support the deployment of services that use OAuth for authorization of their own functionalities.

3.4.4.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Use cases should never ask for a username and password of a user for accessing external services/APIs unless absolutely necessary. Instead, the consumption of OAuth interfaces should be used.

3.4.5 Access Rights

In addition to the user driven authorization described in the last subsection, SIMPLI-CITY will also support the usage of access rights for storing and receiving information from the Cloud Storage. More precisely, the Cloud Storage will provide the possibility to create new isolated storage spaces (Buckets) and to define their access rights for allowing other components, apps or services to access or modify Bucket content.

As such, a Bucket will always have an "owner", which is the creator of the Bucket. This can be a service or a SIMPLI-CITY component. During the creation time, access rights of this

Bucket may be specified. As described in Section 6.2.5.1 and 6.2.5.2 of D3.2.2, a set of Java and a set of REST services will be provided to get, add or modify access rights of a Bucket.

Access rights may cover in two ranges:

1. General access rights: Those will be the base for defining the visibility and access control of a Bucket. This will allow the creation of Buckets that are publically readable for all users such as the key storage Bucket for digital signatures (public RSA key), described in Section 3.4.3.
2. User/Group specific access rights: Those will allow the specification of access rights for a specific user or for a group of users for accessing one specific Bucket via its Bucket id.

Access rights will be specified using the JSON schema described in deliverable D3.2.2 (Section 6.2.6). Possible values for one access right are:

- *NotSet:* will be used to delete an access right
- *Denied:* will be used to deny a user or group the any access to a Bucket
- *Read:* will be used to allow a user or group to read the contents of a Bucket
- *Write:* will be used to allow a user or group to read and write the contents of a Bucket
- *Super:* will be used to allow a user or group the same rights as the owner, this includes the right to grand or restrict access rights to other user or groups as well as to delete the Bucket. It is only restricted in the way, that it will not be possible for a user or group with this access right to remove the owner of the Bucket the super access right. (This right is granted to the owner without a specific access right, hence a restriction will not be possible).

It should be noted that this document describes the holistic approach of SIMPLI-CITY for handling privacy and security aspects. Apps, services and components may at any time extend those concepts or add own access rights methods on top of e.g. the OAuth or on top of the Access Control List (ACL) of the Cloud-based Information Infrastructure. Those may of course be task specific and may only be usable for themselves. For example, task T5.2 will provide a secure access component which will control the data access between backend services and the Context Manager. This will define which backend service may access which context data. It will manage this information within the Cloud Storage by making use of the isolated data Bucket feature.

3.4.5.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Cloud-based Information Infrastructure (Cloud Storage) will support the definition of access rights for restricting access to Buckets. This includes the restriction of reading and/or modifying content.
- All components of SIMPLI-CITY may create Buckets for storing information in the Cloud Storage. When doing so, each component should carefully select the best properties for their Buckets by making use of the Java interface or of the REST interface of the Cloud Storage as described in D3.2.2.

3.4.5.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Services created in the use cases of SIMPLI-CITY may create Buckets for storing information in the Cloud Storage. When doing so, each service should carefully select the best properties for their Buckets by making use of the Java interface or of the REST interface of the Cloud Storage as described in D3.2.2.
- When accessing data from the Cloud Storage, errors will appear in case that a service tries to access a Bucket without having sufficient access rights. Services should be able to cope with those errors, e.g. by forwarding a corresponding error message to the invoking app at the PMA side.

3.4.6 Communication Encryption

SIMPLI-CITY is a service-based system providing apps to users of the PMA. Those apps will utilize services on the server side, e.g. to request the nearest parking lot for a car. As such, SIMPLI-CITY heavily uses the internet to exchange information between the PMA and the server side. As explained in Section 3.2, the internet cannot be considered as a secure data channel as such as it involves routing messages via various third party servers. Therefore, SIMPLI-CITY will encrypt all sensitive messages that leave the PMA. For this purpose, SIMPLI-CITY makes use of the HTTPS standard [Sc08], a protocol for secure communication between two parties. HTTPS protects SIMPLI-CITY from man-in-the-middle attacks by encrypting the communication and allowing only the receiver to decrypt a message.

More precisely, SIMPLI-CITY will make use of HTTPS for any data transfer between the Application Runtime Environment at the PMA side and the Service Runtime Environment at the server side. This also means that all service calls will be secured via HTTPS.

Since HTTPS is a proven and secure protocol used by millions of sites and devices and supported by all operating systems and programming languages, SIMPLI-CITY will be able to make use of the integrated HTTPS functionalities from the underlying operating systems (e.g. Android in case of the PMA) meaning that no HTTPS stack needs to be manually implemented by the project team.

3.4.6.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Application Runtime Environment and the Service Runtime Environment will ensure that all communication between them is performed via HTTPS.
- The Service Runtime Environment will ensure that all service calls to services running not inside the local server will be performed via HTTPS if supported by the service.
- All SIMPLI-CITY components will use HTTPS for communicating with each other in case that REST calls are involved.
- All SIMPLI-CITY components will use HTTPS for communicating to any external data source whenever possible.

3.4.6.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Use cases should use HTTPS for all communication when implementing apps and services whenever possible. Data sources that provide an HTTPS interface should be preferred.
- No sensitive user data may be transferred without the usage of HTTPS connections or manual data encryption (see Section 3.4.8).

3.4.7 End-to-End Encryption

As described in the last subsection, the signing of messages as supported by SIMPLI-CITY and as described in Section 3.4.3 does not encrypt a message. Instead it only makes sure that a message has not been modified on its way and that it is really coming from correct sender. This may in many cases be sufficient and indeed it allows other SIMPLI-CITY components to analyze the content of a message. However, in many other cases, the content of a message should not be accessible by anyone but the final receiver. The encryption of communication described in Section 3.4.6 does secure the communication but only between single connections such as the connection between the Application Runtime Environment and the Service Runtime Environment. In comparison to this, the End-to-End Encryption secures the full communication allowing an app to send an encrypted message which can only be decrypted by the receiver, e.g. a backend service. All other components involved in the transaction – such as the Service Runtime Environment – will not be able to access the content of the message.

To achieve this, SIMPLI-CITY will support the encryption of JSON, XML or – more generic – text messages and will allow to pass those messages along the communication from the PMA to the service and vice versa. To realize this, SIMPLI-CITY will make use of the public and private key concept described in Section 3.4.3. Whenever an End-to-End encryption is needed, the following process will apply:

1. The sender of a message will use the private key and the RSA algorithm to encrypt the message with the public key of the receiver. This public key may be received via the key storage of the Cloud Storage as described in Section 3.4.3.
   The sender may in this case be an app at the PMA side or a service at the server side (e.g. in case of using the Push Service).
2. Once the message has been encrypted, the message will be delivered using the SIMPLI-CITY components.
   For example, an app will use the Application Runtime Environment to call a service via the Service Runtime Environment. Both, the Application Runtime Environment and the Service Runtime Environment will receive the encrypted message as payload information and will pass it to the destination service (the receiver).
3. Only the receiver will be able to decrypt the message as only the receiver has the private key which is needed to decrypt it.

This concept may be used by apps, services and also by SIMPLI-CITY components if required.

3.4.7.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Application Runtime Environment and the Service Runtime Environment will support the carrying of payload information. This payload information may be an encrypted text which will be passed to the destination as a parameter.
- The Application Design Studio and the Service Development API should at least provide one example on how to make use of this feature in order to ease the understanding of it.

3.4.7.2 Concrete Action Points for WP7 and 8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Service and apps which will be implemented in the use cases will make use of the End-to-End encryption mechanisms whenever necessary.

3.4.8 Data Encryption

In some cases, apps, services or SIMPLI-CITY components may need to encrypt data. This could be performed when passing information between different services or when exchanging information between apps and services. In those cases, the approach outlined in Section 3.4.7 will be used and provides a good base for encrypting data.

In some cases, data may also have to be encrypted when being stored permanently. This may be a storage in the cloud via the Cloud Storage or a storage on the PMA side via the Local Key Storage (see deliverables D3.1, D3.2.1 and D3.2.2).

For encrypting data, the encryption will be performed at the data source. For example, an app that wants to store data about a user in the Cloud Storage in an encrypted way will perform the encryption of the data before sending it via the Application Runtime Environment. In those cases – and in all other cases where the outlined private/public key concept of an asymmetric encryption is not suitable - AES 256 will be used as a symmetric encryption algorithm [Sc08].

As all modern programming languages support AES out of the box, the AES encryption will not have to be implemented by SIMPLI-CIY. Instead, the API of the programming language will be used for encrypting and decrypting data with a password and a salt value.

In case that user data needs to be encrypted, the UUID may in many cases be used as a password for the encryption together with an encryption ID which will be generated during user registration and which will be stored on the local PMA without being passed to any external service or component.

3.4.8.1 Concrete Action Points for WP4, WP5 and WP6

This concept has the following impact to the technical components of SIMPLI-CITY:

- The Application Runtime Environment will generate an encryption ID which may be used by apps together with the UUID as a device and user specific password for AES encryption.
- The Cloud Storage and the Local Key Storage will support the storage of encrypted data.

3.4.8.2 Concrete Action Points for WP7 and WP8

This concept has the following impact to the use cases of SIMPLI-CITY:

- Service and apps which will be implemented in the use cases will make use of data encryption whenever storing sensitive user data[12].

[12] For securing the sending and return of data, please refer to Section 3.4.6

4 Ethical and Legal Aspects

4.1 Scope

4.1.1 Definition

The most common way of defining "ethics" is: norms for conduct that distinguish between acceptable and unacceptable behaviour [Res11].

Research ethics involves the application of fundamental ethical principles to a variety of topics involving scientific research. Research ethics is specifically interested in the analysis of ethical issues that are raised when people are involved as participants in research. There are three objectives in research ethics. The first and broadest objective is to protect human participants. The second objective is to ensure that research is conducted in a way that serves interests of individuals, groups and/or the society as a whole. Finally, the third objective is to examine specific research activities and projects for their ethical soundness, looking at issues such as the management of risk, protection of confidentiality and the process of informed consent [Wal13].

Some guidelines on ethics in research are also given by a report on Research Ethics titled "A comprehensive strategy on how to minimize research misconduct and the potential misuse of research in EU funded research" [Rat10]. The aim of the report was to provide a comprehensive strategy on how to safeguard EU funded research against misconduct and misuse. In a comprehensive approach the potential role and proposed actions of relevant stakeholders were addressed:

- The EU Commission and its subsidiary institutions.
- EU Ethics Screeners, Reviewers and Auditors.
- Research project applicants, host institutions and national contact points.

The report concluded that research misconduct and potential misuse constitute an ethical issue in the context of EU funded research and should be systematically addressed in EU Ethic's oversight (Screening, Review and Audit).

In defining the scope of this ethical issue the following definitions were used as guidelines:

- "Potential misuse of research" is defined as: Research involving or generating materials, methods or knowledge that could be misused for unethical purposes.
- "Research misconduct" is defined as: fabrication, falsification and plagiarism.

Legal aspects are in some way easier to be defined: legal is what is permitted by the law.

Even if some licenses could be granted to specific research groups and for specific purposes (e.g., research on drugs or on animals), most of the research and research projects are implemented within the limits imposed by the law. SIMPLI-CITY, as well, strictly adheres to the legal provisions of European and national legislations.

4.1.2 Key Concepts

The importance of ethical issues related to ICT research and technological developments has increased rapidly in recent years. At the same time there is a large literature on ethics,

ranging from ancient Philosophy, to current Professional codes and more recently to European rules.

The two main perspectives behind the ethical aspects are:

- Deontological Ethics that produces moral obligations directly from the type of a certain action, without regard for its consequences.
- Teleological Ethics on the contrary, originates morality from the consequences of an action instead of the intentions.

Some guidelines (and also defined Code of Practices) give concrete assistance for their respective fields. Another relevant example of Ethical codification could be find at company level. In this case ethical rules could be defined by two separate aims i.e., to sets out restrictions on behaviour for the company employees (Code of conduct) and to define the company behaviour towards external stakeholders (Code of ethics). In general codes follow both deontological and teleological approaches.

The following is a rough and general summary of some ethical principles that various codes address according to [Res11]:

- Honesty - Strive for honesty in all scientific communications. Honestly report data, results, methods and procedures, and publication status. Do not fabricate, falsify, or misrepresent data. Do not deceive colleagues, granting agencies, or the public.
- Objectivity - Strive to avoid bias in experimental design, data analysis, data interpretation, peer review, personnel decisions, grant writing, expert testimony, and other aspects of research where objectivity is expected or required. Avoid or minimize bias or self-deception. Disclose personal or financial interests that may affect research.
- Integrity - Keep your promises and agreements; act with sincerity; strive for consistency of thought and action.
- Carefulness - Avoid careless errors and negligence; carefully and critically examine your own work and the work of your peers. Keep good records of research activities, such as data collection, research design, and correspondence with agencies or journals.
- Openness - Share data, results, ideas, tools, resources. Be open to criticism and new ideas.
- Respect for Intellectual Property - Honour patents, copyrights, and other forms of intellectual property. Do not use unpublished data, methods, or results without permission. Give credit where credit is due. Give proper acknowledgement or credit for all contributions to research. Never plagiarize.
- Confidentiality - Protect confidential communications, such as papers or grants submitted for publication, personnel records, trade or military secrets, and patient records.
- Responsible Publication - Publish in order to advance research and scholarship, not to advance just your own career. Avoid wasteful and duplicative publication.
- Responsible Mentoring - Help to educate, mentor, and advise students. Promote their welfare and allow them to make their own decisions.
- Respect for colleagues - Respect your colleagues and treat them fairly.
- Social Responsibility - Strive to promote social good and prevent or mitigate social harms through research, public education, and advocacy.

- Non-Discrimination - Avoid discrimination against colleagues or students on the basis of sex, race, ethnicity, or other factors that are not related to their scientific competence and integrity.
- Competence - Maintain and improve your own professional competence and expertise through lifelong education and learning; take steps to promote competence in science as a whole.
- Legality - Know and obey relevant laws and institutional and governmental policies.
- Animal Care - Show proper respect and care for animals when using them in research. Do not conduct unnecessary or poorly designed animal experiments.
- Human Subjects Protection - When conducting research on human subjects, minimize harms and risks and maximize benefits; respect human dignity, privacy, and autonomy; take special precautions with vulnerable populations; and strive to distribute the benefits and burdens of research fairly.

Most of the above principles are addressed and respected by the member of the SIMPLI-CITY consortium due to the project nature and its planning described in the Description of Work (DOW), both in respect of internal relations and with external stakeholders. Some other principles (e.g., Animal Care) are not directly addressed by the project and thus not relevant for the project purpose.

4.1.3 Main Sources

As far as European rules, the main principle on ethics is stated by the Decision 1982/2006/EC, Article 6: “All the research activities carried out under the Seventh Framework Programme shall be carried out in compliance with fundamental ethical principles”, including those reflected in the Charter of Fundamental Rights of the European Union and taking into account opinions of the European Group on Ethics in Science and New Technologies (EGE).

Even if, as stated in the Description of Works, the European Commission directives on ethical rules are primarily concerned with issues that are outside the scope of SIMPLI-CITY, the SIMPLI-CITY consortium will pay the due attention on any ethical and legal question that may arise during the project lifespan.

Other related sources are[13]:

- The Charter of Fundamental Rights of the European Union signed and proclaimed on 7 December 2000 - The European Union Charter of Fundamental Rights sets out in a single text, for the first time in the European Union's history, the whole range of civil, political, economic and social rights of European citizens and all persons resident in the EU.
- Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 - This covers the protection of individuals with regard to the processing of personal data and on the free movement of such data.
- Convention for the Protection of Human Rights and Fundamental Freedoms as amended by Protocol No. 11 on 1 November 1998 - The "European Convention on Human Rights" sets forth a number of fundamental rights and freedoms (right to life, prohibition of torture, prohibition of slavery and forced labour, right to liberty and security, right to a fair trial, no punishment without law, right to respect for private and family life, freedom of thought, conscience and religion, freedom of expression,

[13] Source: CORDIS – www.cordis.europa.eu

freedom of assembly and association, right to marry, right to an effective remedy, prohibition of discrimination).
- Directive 86/609/EEC of the European Parliament and of the Council of 24 November 1986 - This covers the approximation of laws, regulations and administrative provisions of the Member States regarding the protection of animals used for experimental and other scientific purposes.
- Directive 90/385/EEC of the European Parliament and of the Council of 20 June 1990 - This relates to active implantable medical devices.

The SIMPLI-CITY project fully recognizes the need to ensure the end user privacy / security as already described in sections 2 and 3. In this field the legal references are given by the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. The Directive lays down a series of rights of the data subject. These are:

- The right of access to his / her personal data
- The right of erasure, blocking or rectification of the data, which do not comply, with the provisions of the Directive, are incomplete or inaccurate.
- The right to be informed of all relevant details relating to the data processing and the rights granted to him/her
- The right to a judicial remedy for any breach of the above mentioned rights.

4.2 Impact to SIMPLI-CITY and Requirements to the Project

While requirements based on privacy and security consideration are already described in sections 2 and 3 respectively, this section contains requirements to the SIMPLI-CITY project based on ethical and legal considerations:

4.2.1 End User Requirements

The consideration of ethical and legal aspects from an end user viewpoint lead to the following requirements which have to be considered during the prototype implementations:

- The PMA should be accessible to people with disabilities avoiding any discrimination[14].
- Provision of useful information to drivers that must be legal, according to national regulations on traffic (e.g. speed camera location warning) and ethical avoiding unnecessary and misleading information.
- Non-distraction of the driver must be ensured for safety reasons and in order to be aligned to some provisions of traffic codes that prohibits the usage of handheld IT devices while driving. In general, a driver may use the PMA for navigation or for other non-distracting purposes while driving if it is programmed before the start of the trip or if it can be programmed in a voice-activated manner. Furthermore media data streams could be only for audio entertainment purposes. Video streaming or Television must be avoided.
- Self-Determination: ICT systems such as SIMPLI-CITY are developed in order to assist people. But in any case, the final decision needs to stay at the end user. For example, SIMPLI-CITY may suggest that the end user drives slower to improve his

[14] This may only be partly realized due to the research nature of this project.

environmental footprint but it should not enforce it or decide it on behalf of the end user.
- Malware protection: End users need to be protected from malware and other harmful apps and services. This could be achieved – up to a certain extend – by realizing the review process for apps and services as described within D3.1 and D3.2.1.

4.2.2 Developer Requirements

The following list shows requirements from a developer's viewpoint:

- Local storage of data and storage of data in the cloud must be duly related to the project purpose, avoiding unnecessary storage of data not strictly research related. Indeed, one of the main areas of concern regarding potential research misuse is related to data mining and profiling technologies [Rat10].
- Services and apps developer must be identifiable in order to link the software author in the case of legal or ethical problems.
- Marketplace must be accessible to developers without any discrimination, according to international competition rules.
- Compliance to regulation: Data could be subject to licences. It is important for SIMPLI-CITY to make sure that data which is processed and stored in the Cloud-based Information Infrastructure is compliant with regulations and licensing rules. SIMPLI-CITY has been designed and adapted to be used within the European Union. It should, however, keep the possible usage outside the EU in mind and therefore consider regulative points from a global perspective.

4.3 Possible Standards and Approaches

Standard in ethics are not easy to be found, since the definition of ethics differs according to the field of action and to the actors and stakeholders involved. To narrow the field, as standard for SIMPLI-CITY, could be intended what is foreseen by the DOW, by the Consortium Agreement and by a set of general principles that are commonly accepted in the research field.

As general principles, project partners and developers must avoid research misconduct and especially misuse, such as:

- Falsification, defined as the misrepresentation of results.
- Fabrication, defined as the reporting on experiments never performed.
- Plagiarism, defined as taking the writings or ideas of another and representing them as one's own [Rat10].

Furthermore, from a broader point of view, avoidance of exploitation, just distribution of benefits and burden, beneficence, respect for persons, respect for human dignity, scientific validity, social value, the rights and interests of research participants are overarching ethical principles of any scientific research [SF10].

Concerning legal aspects of SIMPLI-CITY, national and European sources have to be taken into account. In exploitation of project results, it could be necessary to overcome the boundary of partners' countries and to comply with other national or international rules.

4.4 Ethical and Legal Concept of SIMPLI-CITY[15]

Being an ICT RTD project, SIMPLI-CITY is focused on legal issues and does as such not touch any sensitive ethical topics. However, for exploiting the project work, it is indeed necessary to ensure that SIMPLI-CITY does not break European laws and that SIMPLI-CITY does not develop in an unethical direction. For achieving this, the project will follow the following concepts:

4.4.1 Legal Development

During its lifetime, the project consortium will carefully monitor legislation for detecting any legal changes and any European developments that may have an impact to the project. In those cases, legal issues will be raised to the Board of Partners (BOP) which will decide on how to handle the situation and on which actions to undertake.

4.4.2 Ethical Discussions

The project will actively participate in ethical discussions connected to SIMPLI-CITY and will raise those topics during its regular plenary meetings. The involvement of external persons via the project collaboration efforts in WP9 will also allow the project to get immediate feedback from external audiences. This will create an on-going reflection of SIMPLI-CITY and its role in society and will quickly show any ethical issues that might impact the project.

4.4.3 User Data Handling

Users will have the right to request information about which data is stored about them within SIMPLI-CITY. For this purpose the PMA will be able to show the UUID for each app to the user allowing him to request the deletion of data for his/her UUIDs.

Users will also have the right to request the full deletion if their user specific data including entries in the Cloud Storage.

4.4.4 Privacy Statement

The SIMPLI-CITY consortium has created a privacy statement which will be accessible for its users:

Listing 1: Privacy Policy for SIMPLI-CITY

PRIVACY POLICY for SIMPLI-CITY

This privacy policy describes how SIMPLI-CITY uses and protects any information that you transfer when you use its applications and services.

SIMPLI-CITY is committed towards ensuring that your privacy is protected. Should we ask you to provide certain information by which you can be identified when using this system, then you can be assured that it will only be used in accordance with this

[15] Please note: This section is significantly shorter than the other sections. This is not because the project does not value legal or ethical aspect but rather because the project has a strong technical core by its nature. As such, the focus of this ICT RTD project is more on the technical side and as such closer to privacy and security aspects discussed in section 2 and 3.

privacy statement.

SIMPLI-CITY may change this policy from time to time by updating this page. You should check this page from time to time to ensure that you are happy with any changes. This policy is effective from 05.09.2013.

What we collect

We may collect the following information:

- name (given and surname)
- contact information including email address, address, fax and telephone number
- demographic information such as postcode, preferences and interests
- other information relevant to customer surveys and/or offers
- in the case of transactions also payment information

SIMPLI-CITY allows you to install apps and to make use of services which are not developed by the SIMPLI-CITY consortium but by third parties. Those apps and services may collect additional information from you and are not covered by this privacy policy.

What we do with the information we gather

We require this information to understand your needs and provide you with a better service, and in particular for the following reasons:

- Internal record keeping.
- We may use the information to improve our products and services.
- We may periodically send promotional emails about news related to SIMPLI-CITY. This may include the introduction of new features, products, special offers, project newsletters or other information which we think you may find interesting using the email address which you have provided.
- From time to time, we may also use your information to contact you for market research purposes. We may contact you by email, phone or mail. We may use the information to customize our applications according to your interests.

Security

We are committed to ensuring that your information is secure. In order to prevent unauthorized access or disclosure, we have put in place suitable physical, electronic and managerial procedures to safeguard and secure the information we collect online.

How we use cookies

A cookie is a small file which asks permission to be placed on your computer's hard drive. Once you agree, the file is added and the cookie helps analyze web traffic or lets you know when you visit a particular site. Cookies allow web applications to respond to you as an individual. The web application can tailor its operations to your needs, likes and dislikes by gathering and remembering information about your preferences.

We use traffic log cookies to identify which pages are being used (e.g. within the web UI of the Services and App Marketplaces). This helps us analyze data about web page traffic and improve our applications in order to tailor it to customer needs. We only use this information for statistical analysis purposes and then the data is removed from the system.

Overall, cookies help us provide you with a better service, by enabling us to monitor which pages you find useful and which you do not. A cookie in no way gives us access to your computer or any information about you, other than the data you choose to share with us.

You can choose to accept or decline cookies. Most web browsers automatically accept cookies, but you can usually modify your browser setting to decline cookies if you prefer. This may prevent you from taking full advantage of our services.

Links to other websites

SIMPLI-CITY may contain links to other websites, apps or services (hereafter referred to as “sites”) of interest. However, once you have used these links to leave our site/apps, you should note that we do not have any control over that other site. Therefore, we cannot be responsible for the protection and privacy of any information which you provide whilst visiting such sites and such sites are not governed by this privacy statement. You should exercise caution and look at the privacy statement applicable to the website in question.

Controlling your personal information

You may choose to restrict the collection or use of your personal information in the following ways:

- whenever you are asked to fill in a form within an application, look for the box that you can click to indicate that you do not want the information to be used by anybody for direct marketing purposes
- if you have previously agreed to us using your personal information for direct marketing purposes, you may change your mind at any time by writing to or emailing us at contact@simply-city.eu

We will not sell, distribute or lease your personal information to third parties unless we have your permission or are required by law to do so. We may use your personal information to send you promotional information about third parties which we think

you may find interesting if you tell us that you wish this to happen.

You may request details of personal information which we hold about you. If you would like a copy of the information held on you please write to contact@simply-city.eu

If you believe that any information we are holding about you is incorrect or incomplete, please write us a postal mail or email as soon as possible, at the above address. We will promptly correct any information found to be incorrect.

Acknowledgement
This privacy policy used by SIMPLI-CITY is partly based on text from www.businesslink.gov.uk and is published under the Open Government Licence v2.0.

4.4.5 Terms and Conditions

The SIMPLI-CITY consortium has created terms and conditions which will be accessible for its users:

Listing 2: Terms and Conditions for SIMPLI-CITY

TERMS AND CONDITIONS for SIMPLI-CITY

These terms of use govern your use of SWILI-CITY. Please read these terms in detail before you use SIMPLI-CITY or any part of it. If you do not accept these terms of use, please do not use it. Your continued use of SIMPLI-CITY confirms your acceptance of these terms.

Use of SIMPLI-CITY

- It is not necessary to register in order to use most parts of SIMPLI-CITY. However, particular areas of SIMPLI-CITY will only be accessible only if you have registered.
- SIMPLI-CITY may be used for your own private purposes and in accordance with these terms of use.
- You may print and download material from SIMPLI-CITY provided that you do not modify or reproduce any content without our prior written consent.

Up Time

- All reasonable measures are taken by us to ensure that SIMPLI-CITY is operational all day, every day. However, occasionally technical issues may result in some downtime and accordingly we will not be liable if SIMPLI-CITY is unavailable at any time.
- Where possible we always try to give advance warning of maintenance issues that may result in SIMPLI-CITY down time but we shall not be obliged to provide such notice.

User Generated Content or Content from Third Parties

- Any material that a visitor to this Website sends or posts to SIMPLI-CITY shall be considered non-proprietary and non-confidential. We shall be entitled to copy, disclose, distribute or use for such other purpose as we deem appropriate all material provided to us, with the exception of personal information, the use of which is covered under our Privacy Policy.
- When using SIMPLI-CITY you shall not post or send to or from it any material:
  (a) for which you have not obtained all necessary consents;
  (b) that is discriminatory, obscene, pornographic, defamatory, liable to incite racial hatred, in breach of confidentiality or privacy, which may cause annoyance or inconvenience to others, which encourages or constitutes conduct that would be deemed a criminal offence, give rise to a civil liability, or otherwise is contrary to the law in the European Community;
  (c) which is harmful in nature including, and without limitation, computer viruses, Trojan horses, corrupted data, or other potentially harmful software or data.

Links to and from other websites/applications

- Throughout SIMPLI-CITY you may find links to third party websites/applications. The provision of a link to such a website does not mean that we endorse that website. If you open any website/application via a link on SIMPLI-CITY you do so at your own risk.
- Any party wishing to link to SIMPLI-CITY is entitled to do so provided that the conditions below are observed:
  (a) you do not seek to imply that we are endorsing the services or products of another party unless this has been agreed with us in writing;
  (b) you do not misrepresent your relationship with this website/application; and
  (c) the website/application from which you link to SIMPLI-CITY does not contain offensive or otherwise controversial content or, content that infringes any intellectual property rights or other rights of a third party.

Disclaimer

Whilst we do take all reasonable steps to make sure that the information on SIMPLI-CITY is up to date and accurate at all times we do not guarantee that all material is accurate and ,or up to date.

All material contained on SIMPLI-CITY is provided without any or warranty of any kind. You use the material on SIMPLI-CITY at your own discretion.

Exclusion of Liability

- We do not accept liability for any loss or damage that you suffer as a result of using SIMPLI-CITY.
- Nothing in these Terms of Use shall exclude or limit liability for death or personal injury caused by negligence which cannot be excluded or under the law of the European Community.

Law

These terms of use are governed by EU-law. Any dispute arising in connection with these terms of use shall be subject to the exclusive jurisdiction of the Courts of the EU.

Acknowledgement

This privacy policy used by SIMPLI-CITY is partly based on text from http://www.diylegals.co.uk/ecommerce/website-terms-conditions/ and is published under the Creative Commons License.

5 Conclusion and Outlook

This deliverable has described the security and privacy concept of SIMPLI-CITY. It has split this topic into privacy issues, security issues and ethical and legal issues. Each of them has been described in an own section starting with a short definition and with a discussion about the basic concepts. A short summary of standards and approaches has been performed before addressing the impact and requirements that is caused by privacy, security, ethical and legal aspects to the project. Those requirements are completing the D2.1 requirements and have formed the base for a description of concepts. Those concepts have been outlined in Sections 2.4, 3.4 and 4.4.

Among a large amount of smaller decisions, this deliverable defines the following key concepts for the project:

- Whenever possible, users will be able to make use of SIMPLI-CITY without the need to transfer any personal data outside the PMA.
- If a user has to be identified, then SIMPLI-CITY will use a unique ID for a user, i.e. a UUID. This allows apps and services to recognize a user without being able to identify his real identity. A single unique UUID could lead to a security problem in case of that UUID was used to correlate data from different services and then subvert (to some degree) the pseudonym. In order to avoid this, SIMPLI-CITY will return a separate UUID *per app* so that each app of the PMA will have an own UUID.
- The OpenID standard will be supported for any web UI created by the project. This includes OpenID support for developers when accessing the Services and App Marketplaces.
- Messages exchanged between different parts of SIMPLI-CITY may be signed by making use of RSA in combination with SHA-256. This is especially useful when considering the communication between apps and services.
- End-to-End encryption is supported by SIMPLI-CITY by allowing the transfer of encrypted information, which will be embedded in the payload of a service call.
- AES will be supported for symmetric encryption of data in combination with the aforementioned UUID.

Each subsection has given a set of concrete action points for the technical components (realized by WP4, WP5 and WP6) and for the use cases (realized by WP7 and WP8). Those actions points will be addressed during the prototype implementations of SIMPLI-CITY. Task T3.3 does not foresee any own development efforts but it will monitor the developments of WPs 4-8 in order to ensure that the concepts described in this deliverable have been followed.

References

[AR03] Anderman, G. M.; Rogers, M.: Translation Today: Trends and Perspectives, Multilingual Matters Ltd, 2003

[Be09] Behringer, M.H.: End-to-End Security, The Internet Protocol Journal, Volume 12, No.3, 20, Cisco, 2009

[Bun11] Bundesamt für Sicherheit in der Informationstechnik (BSI): Sicherheitsempfehlungen für Cloud Computing Anbieter: Mindestsicherheitsanforderungen in der Informationssicherheit / Bundesamt für Sicherheit in der Informationstechnik (BSI). 2011. – Forschungsbericht

[CC12] Chapman, N.; Chapman, J.: Authentication and Authorization on the Web, Macavon Media, 2012

[CDG13] Choudhary, S. K.; Dubey, S. K., Gupta, R.: Wimax Technology: A Secure Broadband Connectivity for Governments, Military Services in Rural/Strategic Isolated Locations, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, 2013

[CL08] Conger, S.; Landry, B. J. L.: The Intersection of Privacy and Security, University of Dallas, USA . Sprouts: Working Papers on Information Systems, 8(38), 2008

[CMF11] Conrad, E., Misenar, S., & Feldman, J. (2011). CISSP study guide.

[Gen09] Gentry, C., „A Fully Homophorphic Encryption Scheme“, Ph.D. dissertation, Department of Computer Science, Stanford University, 2009.

[EEH+97] Ernestus, W.; Ermer, D.; Hube, M.; Köhntopp, M.; Knorr, M.; Quiring-Kock, G.; Schläger, U.; Schulz, G.: Datenschutzfreundliche Technologien / "Technische und organisatorische Datenschutzfragen" der Datenschutzbeauftragten des Bundes und der Länder. 1997. – Forschungsbericht

[FK07] Feruza, S. Y., Kim, T.: IT Security Review: Privacy, Protection, Access Control, Assurance and System Security, International Journal of Multimedia and Ubiquitous Engineering Vol. 2, No. 2, April, 2007

[Got02] Gottschalk, S.: Anonymisierung von Unternehmensdaten: ein Überblick und beispielhafte Darstellung anhand des Mannheimer Innovationspanels / ZEW Discussion Papers. 2002. – Forschungsbericht

[ITS91] ITSEC: Information Technology Security Evaluation Criteria (Provisional Harmonised Criteria, Version 1.2, 28 June 1991)

[MAO96] Menezes, Alfred; van Oorschot, Paul C.; Vanstone, Scott A. (October 1996). Handbook of Applied Cryptography. CRC Press

[OE02] Organisation for Economic Co-operation and Development (OECD): Security of Information Systems and Networks, http://www.oecd.org/sti/ieconomy/37418730.pdf, 2002

[Pe11] Perrin, C.: The CIA Triad, TechRepublic, http://www.techrepublic.com/blog/it-security/the-cia-triad/, 2011

[Per09] Pearson, S.: Taking account of privacy when designing cloud computing services. In: Software Engineering Challenges of Cloud Computing, 2009. CLOUD '09. ICSE Workshop on, 2009, P. 44-52

[Sc08] Schneier, B.: Schneier on Security, Wiley, 2008

[SCG12] Stewart, J. M., Chapple, M., & Gibson, D. (2012). CISSP: Certified Information Systems Security Professional Study Guide. John Wiley & Sons.

[Sm94] Smith, H.J.: Managing Privacy: Information Technology and Corporate America, The University of North Carolina Press, 1994

[US05] US Department of Defense, Dictionary of Military and Associated Terms, INFOSEC, 2005

[We67] Westin, A.: Privacy and Freedom, Atheneum, 1967