FP7-SMARTCITIES-2013

STREETLIFE

Steering towards Green and Perceptive Mobility of the Future

WP2 – ARCHITECTURE & INTEGRATION

D2.2.2 – Blueprint Architecture, Security Architecture and Pilot Specific Architectures (Intermediate)

Due date: 30.09.2015 **Delivery Date**: 02.10.2015

Author(s): Yogesh Nagappa (SIEMENS), Silke Cuno (Fraunhofer), Florian Thiemer (Fraunhofer), Mika Vuorio (Logica (CGI)), Giuseppe Valetto (FBK), Stefan Schaffer (DFKI)

Partner(s): SIEMENS, Fraunhofer, Logica (CGI), FBK, DFKI

Editor: Yogesh Nagappa (SIEMENS)

Lead Beneficiary of Deliverable: SIEMENS

Dissemination level: Public **Nature of the Deliverable:** Report

Internal Reviewers: Peter Deussen (Fraunhofer), Klaus-Peter Eckert (Fraunhofer), Antii Nurminen (AALTO)

EXECUTIVE SUMMARY

In the second year of the STREETLIFE project WP2 delivered an extension to the previous year's deliverable on the blueprint, security and pilot architectures; initial. This next deliverable contains an architectural integration concept and example implementations that span across the pilots. Using the integration concept the STREETLIFE partners and pilots have been able to successfully integrate their services and their components between each other's sites. In addition, STREETLIFE has actively propagated this concept and lessons learned at various collaborative workshops within the EU project space.

In the deliverable, after describing the background and motivation, systematic steps to arriving at the STREETLIFE integration concept are listed. This provides the reader a very quick overview and a step by step guide to arriving at an integration concept. The purpose being that this could also be useful to readers from other EU projects. It could potentially be used as '*a framework for an early approach within other EU projects or pilots towards an integrated technical architecture*'.

In the chapters to follow, each of these individual steps is elaborated in detail. The STREETLIFE specific integration problem is elaborated, the search for a ready-made solution, finding the closest one that solves the problem and extending it to exactly meet the requirements is described. The two extensions; two branches of the integration concepts – a vertical and a horizontal concept and experiences with example implementations per branch are described. The lessons learned in the process and how the knowledge gathered is disseminated to other similar EU projects is mentioned. The final chapter contains detailed background developments within the EU that has influence on the developments within projects of the type of STREETLIFE along with the requirements they put forth. The evolution of these similar developments and requirements within the telecom industry in the past and its parallel with the mobility domain are detailed. Finally, the deliverable concludes with potential next steps and directions for the WP's work in the following year.

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

D2.2.2 – Blueprint Architecture, Security Architecture and Pilot Specific Architectures (Intermediate)

Table of Contents

ABBREVIATIONS

AAA	Authentication, Authorization and Accounting
CO	Confidential, only for members of the Consortium (including the Commission Services)
CIP	City Intelligence Platform (SIEMENS)
BER	STREETLIFE Berlin-Pilot
D	Deliverable
DoW	Description of Work
DPE	Distributed Processing Environment
FP7	Seventh Framework Programme
FLOSS	Free/Libre Open Source Software
GE	Gamification Engine
GUI	Graphical User Interface
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IPR	Intellectual Property Rights
MaaS	Mobility as a Service
MGT	Management
MMECP	Mobility Management and Emission Control Panel
MS	Milestone
OS, OSS	Open Source, Open Source Software
O	Other
P	Prototype
PU	Public
PM	Person Month
R	Report
ROV	STREETLIFE Rovereto-Pilot
RTD	Research and Development
TINA-C	Telecommunication Information Networking Architecture - Consortium
TMF	TeleManagement Forum
TRE	STREETLIFE Tampere-Pilot
WP	Work Package
Y1,2,3	Year 1,2,3

TABLE OF FIGURES

1. BACKGOUND

In the previous year, WP 2 delivered a study on the environments & gaps (D2.1 – Report on gap analysis and incentive models) [1], the initial STREETLIFE blueprint and security architecture documentation (D2.2.1 – STREETLIFE Blueprint Architecture, Security Architecture and Pilot Specific Architectures, initial) [2] that identified sub-systems and their components and scenarios that are essential for the realization of the goals of STREETLIFE [2]. This provided a strong theoretical foundation from which many practical components developed at the pilots.

The deliverable also identifies [2, p. 17] that the integration is always on a per site basis and the STREETLIFE system is not in control [2, pp. 45,46] of how city traffic data can be integrated with itself which implies that the STREETLIFE system has to be adaptive when considering commissioning in other cities. It also identifies the synergies between the various work packages within STREETLIFE and WP 2 that is responsible for the architecture and integration topics.

This next deliverable from WP 2 of STREETLIFE builds on these foundational aspects put forth in the work from the previous year. The information available within D2.2.1 and the current deliverable therefore are mutually exclusive and do not contain duplicate or overlapping information. Therefore this deliverable is a self-contained document with references made to the D2.2.1 as necessary.

An important note for the reader is the title has been maintained as specified in the DoW. At the outset, the contents of this document seem revolve mainly around the topic of Integration. Although, the process for arriving at an Integration concept within a project can be considered a part of the abstract blueprint architecture. Similarly, architectural security aspects and specializations of the integration concept are implemented at the pilots via scenarios and components. These are addressed in this deliverable at the relevant locations and hence the contents and the title of this deliverable are aligned.

2. MOTIVATION

2.1. Address the WP's scope of work

The objective of this work package within STREETLIFE is to define a high level architecture that is generic to be used for different cities. This definition of an integrated architecture blueprint for an urban mobility system it to be made as a service oriented architecture comprising of loosely coupled components integrated using light weight web based interaction protocols. In addition, security is of paramount importance since this could potentially be an integrated system dealing with citizen data.

WP2's work is centered on the creation and evolution of a blueprint, security and pilot specific architectures. It also specifies that these pilot specific architectures should work in an integrated fashion enabled by exchange of common components and usage scenarios. D2.2.1 defines the blueprint in detail and also the initial pilot architectures [2].

As a continuation of this work, for the second year, the focus is on extending the blueprint with an integration concept. That is, to create a strong integration concept with supporting processes that can be constituted within the blueprint processes evolving within the project.

In addition, the WP is also entrusted with the responsibility of developing synergies with other research projects in related domains and technology fields as STREETLIFE.

2.2. Technically enable the STREETLIFE project goals

It is a project goal of STREETLIFE that all pilots work in tandem and not in silos, that the collaborative nature of the project is enabled by strong foundational technical concepts.

Along the same lines an ideal project goal would be to create an extensible, flexible STREETLIFE system that can be extended to support all pilots and further in any other European Union city environment that shares the objectives of a STREETLIFE system. This realistic system should be extended with minimum or zero reconfiguration and structural changes to the system itself in an agile way.

Although, what is practically achievable within the timeframe of the project is to create components that are aligned with the STREETLIFE goals that can be customized and easily integrated to be reused within the pilots. While doing this transfer of components between pilots it would be possible to extract a practical repeatable process to integrate the STREETLIFE system (or components) when reusing it other cities that share STREETLIFE goals.

2.3. Address the need for a grounded approach

Extending the previous paragraph, the next step is a more practical and grounded approach. The previous year's work provides a plan and a template to start with. Taking it forward it is necessary to pit this plan and template against ground reality and create instances that are tangible and one can work with practically. Any lessons learnt can also be fed back into the plan and template thereby improving it in the next iteration.

2.4. **Address previous review findings**

2.4.1. On Extensibility of the STREETLIFE concepts

The previous year's review noted that the project should seek to maximize the extensibility of its solutions and architecture beyond the requirements of the 3 pilot sites in STREETLIFE. The project should seek to be extensible beyond the lifetime and scope of STREETLIFE. It should also allow extensibility in terms of new feature additions for e.g. to include features such as congestion charging etc.

2.4.2. On structure, re-usability and integration

The previous year's review also noted that there needs to be better clarity on how the environment / system / sub-systems / components within various pilot sites are integrating. There seems to be the first impression that they are working in silos. The project should make it clear that there is a concept that clarifies the re-usability and integrability of the solutions coming from the project.

2.4.3. Goals derived from review comments

> *"To include in the technical STREETLIFE system the ability to allow and accept significant extension of its capabilities without major rewriting of code or changes to the basic technical architecture."*

In D2.2.1 the blueprint architecture provided a generic, abstract and extensible template for the pilot systems, sub-systems and components. The actual specializations of this blueprint, that are the pilot architectures are also to be enabled with processes to help extensibility despite possible constraints at an implementation level.

2.4.4. Steps towards the derived goals

1. Represent and relate STREETLIFE objectives and STREETLIFE solutions in a single view.

2. Create a concept for Integration of these STREETLIFE Solutions. At the end of the first research cycle of the pilots, there is information available to enable an integration concept. Since this can be done via many mechanisms - common APIs, common data models, shared generic data objects used by specific solutions etc., we will need to evaluate the appropriateness of a selected mechanism for STREETLIFE.

3. Choose between the variety mechanisms mentioned above as applicable, it is required that there is an identification and definition of solid (meaning high impact, maximum yield for success), well scoped (to ensure feasibility) usage scenario(s) of such an integrated STREETLIFE system. This is a very important prerequisite to achieve any concrete, practical and demonstrable results. This top down approach defines specific components (or specific solutions and their concepts) within the specific pilot scenario(s). Using this it

is possible to arrive at generalizations for the same, thereby ensuring extensibility as defined in the goal above.

4. Develop a scenario based integration concept with a clear scope that deals with the following questions

   What are the exact requirements of the defined integration scenario(s)?

   What data is required to satisfy the scenario(s)?

   How is this data acquired?

   What data and concept models need to be created for these defined scenario(s)?

   How do we ensure that syntactic and semantic interoperability is defined for compatible access mechanisms?

   What compatible API's (signatures) need to be defined?

5. Implement the integration scenarios considering the technical architecture of the available pilots.

6. Define relevant extension scenarios considering data and functionality together with a corresponding extension process for extending STREETLIFE solutions and the STREETLIFE technical architecture while on-boarding new features and newer cities data/solutions (assuming that they share many of the SL objectives).

3. ARRIVING AT THE STREETLIFE INTEGRATION CONCEPT

This chapter describes the generalization of the process we followed within the STREETLIFE project to arrive at an integration concept that is agreeable to all project partners. This process is generalized and presented in this separate chapter with the vision that this might enable other similar projects to use it as a guideline and adapt it to their specific needs.

The chapters following the current chapter detail the specific results of this process for arriving at the integration concept in the instance of the STREETLIFE project.

- STEP 1: Derive a common understanding on the definition and meaning of Integration within the project team
- STEP 2: Identify the specific integration problems within STREETLIFE
- STEP 3: Perform a state of the art survey to see if there are similar problems and how they have been solved so far
- STEP 4: Based on the results arrive at an Integration concept and communicate this within the project for acceptance, development of components from the research WPs
- STEP 5: Adopt the integration concept by defining aligned activities across WPs enabled by project processes and pilot architectures supporting the integration and exchange of components from STEP 4.

4. DEFINITIONS OF INTEGRATION

There are different meanings of integration architecture and different approaches to arriving at one. Of these meanings and approaches, one approach or meaning is good or better "per se", but each is more appropriate and effective for a specific class/kind of system or system purpose. The objective in this chapter is to *'only collect all possibilities and not to decide on or strike out any particular definition'*.

The expertise of the partners allowed to analyse the different concepts, check their applicability for the STREETLIFE system and choose the more effective (combination of) approaches. Jointly, all team members exhaustively listed all the possible definitions that came to their minds when they thought 'integration'. This was captured, condensed and documented so that, later, this list could be used to validate the scenarios that were considered to be adaptive to all or most of these definitions of 'Integration'. ***At this stage it is assumed that an integration concept could potentially accommodate all these possibilities***. Although it should be noted, for implementations due to time/resource constraints it might not be possible to demonstrate all these meanings in a tangible way.

Here is a list of possibilities what 'integration' could mean within the STREETLIFE context.

- Common platform
  - A platform that puts all the pieces of the system together
- Common domain/concept models
  - Models that can be reused when new systems need to be developed for new cities that would also want to implement STREETLIFE objectives
- Common objectives
  - Arriving at common objectives could by itself be a first step for reusability
- Common implementations for various objectives
  - One implementation for various objectives. E.g. Carbon Emissions Calculator
- Common processes (non-software, e.g. operational processes) for achieving objectives
  - Methodologies that can be shared across pilots for achieving objectives
- Common data formats for exchange between systems
  - An agreement on data formats could make sub-systems and components within a STREETLIFE system compatible and interoperable
- Integration with existing city IT infrastructure
  - Integration with existing and functioning city IT Infrastructure and citizen sourced information systems and methodologies
- Common APIs for exchanging information
  - Re-usable, advertised and published API's that can be re-used for a new purpose with little or no modification
- Integration of providers (e.g. car sharing, bike sharing)
  - Data sources per type of data are many, putting them all together would provide a integrated view on a type of data that can be reused for various purposes
- Integration of applications
  - The ability to make all applications work together, this is a combination of various definitions above

- Integration of various STREETLIFE stakeholders
    - In the STREETLIFE context there are many stakeholders – citizens, IT admins, data providers, mobility mangers etc. This definition means to achieve an end to end integration with a feedback loop between stakeholders. For e.g. mobility managers having the ability to know and influence citizen behavior or citizens knowing that their feedback or inputs are being acted upon.

5. The Streetlife Integration Problem

At the end of year 1, STREETLIFE had developed many components that were suitable for the pilot platforms they were built for. It was already known right from the time of the proposal stage that the partners were well aligned with each other and their capabilities and ideas complemented each other very well. So within the project there was already a good starting point to collaborate and integrate components across partners and pilot sites. In some areas, such as in cases of components that could not be distributed (e.g. data warehouses) it was not clear how this was to be re-used across other pilots. In effect, there were three pilot sites, with three solution platforms and many components that were developed for pilot experiments that could be potentially re-used at other pilot sites.

At the highest level of abstraction the integration problem can be summarized as, three pilot platforms with locally evolving platform components, a need to share components between pilots and a need to describe usage scenarios that would use different pilots and their components. These components evolving at the various pilots have to be exchanged, integrated and re-used within the project at other pilot sites based on the requirements and scenarios of the experiments planned at these other pilots. The diagram below summarizes[1] the pilots and components setup in STREETLIFE without an integration approach.

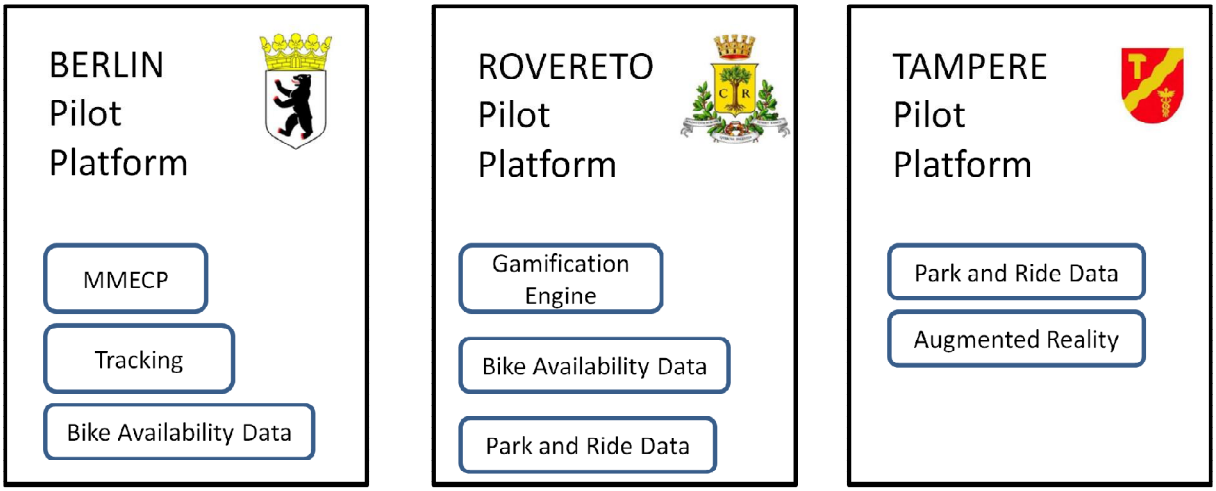

Figure 1: Sample set of STREETLIFE components per pilot

In general, it is also possible that other EU projects could also have similar problems when considering the integration context. There could be *n* EU cities with partners who are developing components and experiments in these cities' deployment environments. For increasing the total throughput of a project it would be beneficial to re-use components that are developed at one partner's pilot site with another partner in other pilot sites.

[1] Not all applications and components from all sites are listed here. It is only a subset of the applications and components developed that could be potentially used for reference implementations of the integration concept.

The above describes the need within STREETLIFE for having an integration concept. This also considers the major constraints put forth by the pilots such as '*they cannot run their experiments from systems that are not local to their pilots*'. This is due to a variety of reasons ranging from intellectual property rights of systems on which these are based to the nature of the technical systems and their service level agreements.

Some parts of the overall problem described above could be solved partially by the various ways that are also listed as various definitions of what 'integration' means to the STREETLIFE partners in the previous chapter.

In the next sections, some definitions of 'integration' are chosen as examples to discuss in detail why considering only one or a few of the definitions will not satisfy our need for an '*Integration Concept*'.

5.1. **Common Platform**

Would the use of a single common platform (common component deployment environment) at the outset of the project for all pilots/partners in a project solve the problem?
It was found to not be so, since it still leaves open how integration with various city systems, integration of applications is to be achieved. Additionally, the setup of such a common platform is not feasible due to the constraint mentioned earlier that pilots cannot run their experiments on remote systems. Even if the constraint did not exist creation of such a general purpose platform causes significant efforts and shifts the focus of the project from its main goals.

5.2. **Common Concepts and Data/Domain Models**

This seems like an attractive solution initially. Although, a deeper look into this would show this to be having similar effects at a conceptual layer as the common platform at the technical layer has.

It is attractive since, if there was a STREETLIFE domain model from which data models could be derived. Then integrating with systems outside of STREETLIFE would mean that the outside systems' domain/data models only need to be transformed into STREETLIFE's and after that the systems are compatible and can understand each other. This is similar to the reference models described in the Microsoft developer network blogs [3].

There is experience in this area in STREETLIFE from the work done within WP3 in Y1 [4] [5]. WP3 has produced STREETLIFE conceptual models and specifications for mobility data integration. WP3 provided a high-level data model that is generic enough to be used as blueprint for the pilot sites. Its purpose was not to provide any detailed level of database design. The purpose of this model was to provide an overview and common understanding of the required data entities that need to be represented, their main attributes and the main relations among them.

5.3. **Top down vs. Bottom up approach**

In essence, the problem at the highest level of abstraction would need to be addressed first and solution has to be found. It makes more sense to employ a top down approach moving from such an overall concept/solution into various specific technical solutions on a case by case basis. To start with the bottom up approach is opted out since it provides too many options and did not provide a structure that was manageable within the timeframe of the project.

6. Potential Solutions Discovered

The search for solutions with a similar problem definition as described in the previous chapter was conducted and many interesting solutions which solved wholly or partially the problem were found. The solution that best answers our problem is mentioned in the end and other intermediate findings are listed earlier.

Interoperability is a word that is closely associated with Integration. The ETSI approach [6] on interoperability was also considered as a candidate that might solve the defined problem exactly. Although, it was concluded that this was not an exact fit. It was since this approach is mainly concerned with the endpoints of applications and not with the distribution and design of these components in a project context similar to STREETLIFE.

Similarly, the integration patterns and practices whitepaper [7] from Salesforce, though insightful and very useful, when coming down to specific interface solutions was also not an exact fit.

Finally, a model that was described for technical and application integration in the book Managed Evolution [8, p. 101] was discovered. This model addressed the situation that was most similar to that within STREETLIFE.

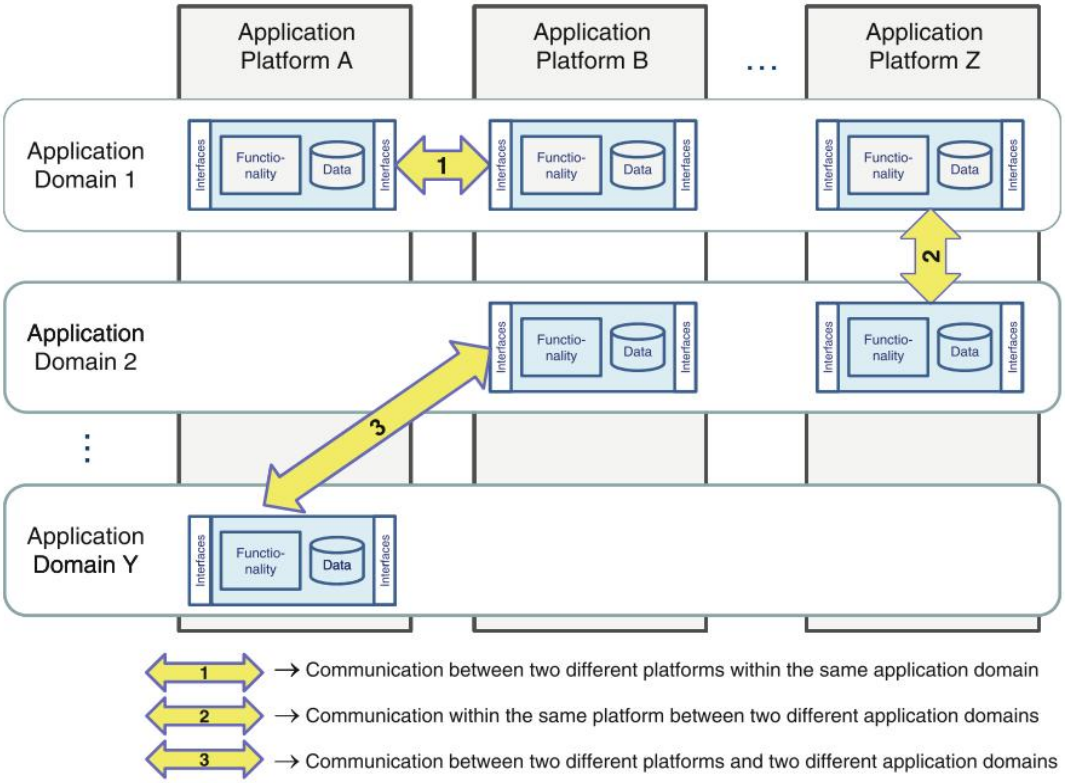

Figure 2: The closest partial solution match to the STREETLIFE Integration problem

Looking at the diagram, it is quite evident how these elements map exactly to the problem described in a diagram in the previous chapter. The application platform maps to the individual pilot platforms within STREETLIFE and the application domains map to the individual components and their domains within STREETLIFE.

This model gives a good high level structure for thinking about types of integration - process integration or data integration. These two types of integration are detailed further detailed in the pattern selection guide from Salesforce's integration patterns [7] and summarized below.

- **Process-based integrations** can be defined as ways to integrate the processing of functional flow across two or more components. These integrations typically involve a higher level of abstraction and complexity, especially with regard to transactions and rollback.
    - This could include event handling, protocol conversion, mediation routing, process choreography and service orchestration

- **Data integrations** can be defined as the integration of the information used by components. These integrations can range from a simple table insert or update, to complex data updates requiring referential integrity and complex translations.
    - They could also include Extract Transform Load (ETL) jobs
    - This would involve the integration of data models & data representations

6.1. **Conclusions on potential solutions**

The search for solutions was done based on our definition of the problem. This gave us a very good start in developing the STREETLIFE '*Integration concept*'. There were some solutions that were discovered that were in too deep in the technical aspects which are bookmarked for future references when specific situations requiring them might be encountered within the STREETLIFE project lifecycle.

The technical and component integration approach and model detailed in the Managed Evolution book [8] was the best fit in the situation within STREETLIFE and provided a strong initial model to start work towards creating an overall STREETLIFE *'Integration concept'*.

The model is enhanced and extended further to accommodate the STREETLIFE specific situation in terms of integration. These extensions are detailed in the following chapter.

7. STREETLIFE HORIZONTAL AND VERTICAL INTEGRATION CONCEPTS

The best matching solution from the survey has been tweaked, mapping all the names that are relevant within the STREETLIFE context. It is also further extended to include two types of integration concepts that are relevant within the STREETLIFE context.

The diagram below shows the adapted model from the Managed Evolution book [8].

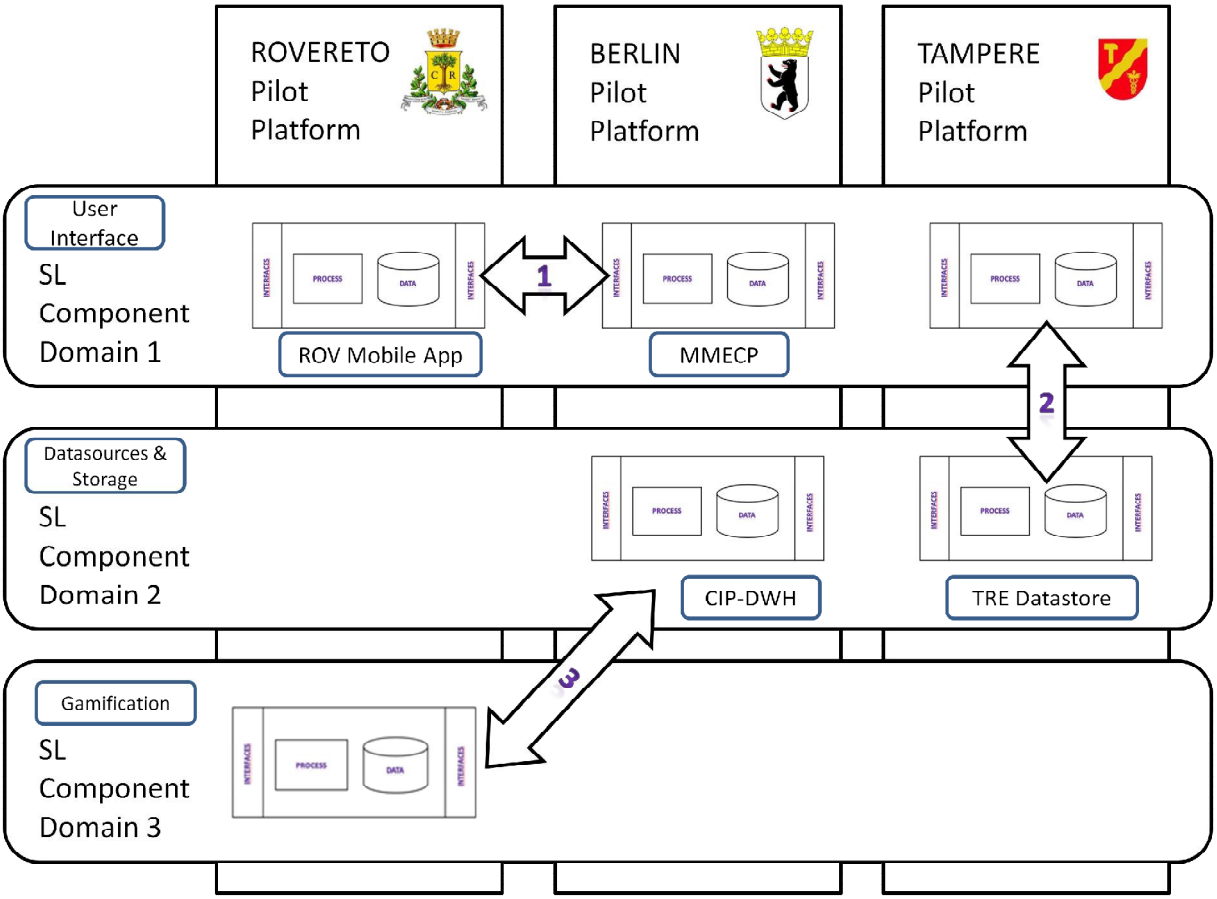

Figure 3: Adapted model to elaborate the STREETLIFE integration problem

We have three platforms one for each pilot city participating in the STREETLIFE project.

Every component that is developed within each of the pilot platforms has its own domain and it performs using the process functionalities and the data it holds. It has interfaces through which it communicates with other components

1. within the platform (<2> from the diagram)
2. between platforms but within the same component domains (<1> from the diagram)
3. between platforms and component domains (<3> from the diagram)

It is evident from the diagram above that we can now have each city independently developing components for its own experiments thereby satisfying the constraints such as inability to run experiments in non-local (outside the pilot) systems. Additionally, this

facilitates the exchange of components between pilot city platforms and also allows for pilot city platform components to communicate with each other and exchange data for usage scenarios that require orchestrations of application workflows via the participating components. This is detailed in the two types of Integration concepts in the next sections.

It should be noted that the following sections present these two concepts individually along with example implementations in each integration concept area. Since the reusability of the concept and the repeatability of the process of application of this concept is the main take away of this deliverable, one or two examples are documented for each of the horizontal and vertical integration concepts. These are not the 'only' examples of integration within the project.

7.1. **Horizontal Integration Concept**

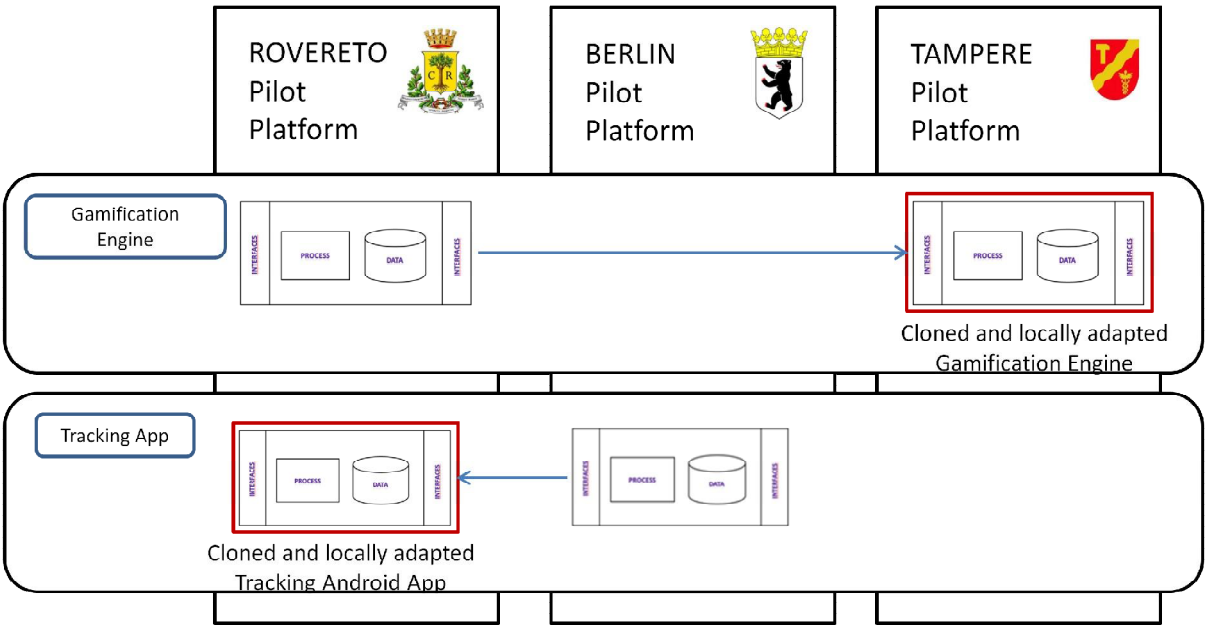

Figure 4: The STREETLIFE Horizontal Integration Concept

As depicted in the sample component exchange diagram above, components successfully developed and deployed within one from one pilot are distributed and made available to other pilot sites. The blue arrow in the diagram above indicates a component being developed at one pilot, shipped to another pilot and locally installed and used within that pilot. For example, the gamification engine developed within the ROV pilot is taken as is, configured and installed locally within the TRE pilot platform.

In the context of the STREETLIFE Integration Concept this is termed as ***Horizontal Integration***, where components are taken from one pilot site and deployed and used (as is or with some modifications) at other STREETLIFE pilot sites.

7.1.1. Process for facilitating component exchanges between pilots

7.1.1.1. Definitions

- ***Producers*** are stakeholders at pilot sites who are creating the components
- ***Consumers*** are stakeholders who are procuring, deploying and operating these components within their pilot sites

7.1.1.2. Process states of the exchanges

1) Producer provides the consumer latest version of the component that can be deployed locally or used as a service by the consumer
2) Consumer deploys the component locally or uses the service at the consumer site
3) Consumer provides feedback/enhancements/bugs (collectively known as change requests) to the producer
4) Producer evaluates these change requests and estimates efforts to develop them
5) Producer decides to incorporate the change requests into the next release
6) Producer incorporates the change requests and provides a new version to the consumer
7) Loop again from 1.
8) Final confirmation from consumer that producer's component has been incorporated into their site.

The goal of the process is to provide transparency within the project to all project members on the states of the exchanges. It also enables project management with a clear overview of the progress.

7.1.2. Example Implementation

7.1.2.1. Gamification Engine

The Gamification Engine (GE) is a component developed in STREETLIFE by FBK. In its current release, it is available for deployment and integration according to two different modalities [9]:

1. **Stand-alone**: the GE is released in this case as a web application, which packages all the design elements of the current release, which are discussed in detail in D5.2.2 [10]. The stand-alone deployment option can be downloaded, compiled, installed and configured within the own infrastructure of the organization that intends to adopt the GE; instructions on this deployment modality are found at the Web page mentioned above.
2. **Hosted:** this modality follows the Software-as-a-Service (SaaS) paradigm, which means an installation of the GE is instantiated and maintained within the Smart Community Open Service platform at FBK [11], on the behalf of the organization using the GE. In this case, the management and administration of the hosted GE installation remain under the

responsibility of said Open Service platform, while the user organization leverages some tools and services that we have made available for configuring the GE, and developing and launching one or more games on the hosted installation. Details on the hosting procedures are also available at the Web page mentioned above.

One major reason for providing these two different deployment options is to offer *flexibility of integration* within specific instances of the STREETLIFE platform and to respond to a diversity of integration requirements.

Clearly, the stand-alone option provides superior configurability and customizability; it can also be more easily used for local development, extension and experimentation purposes. The hosted SaaS option provides maximum convenience for organizations that intend to use the GE *as is,* in a black-box fashion, to address Smart City scenarios and applications (including sustainable mobility scenarios in STREETLIFE), which just need the super-imposition of a gamification solution.

7.1.2.2. Purpose of the gamification Engine – TRE

The gamification engine allows STREETLIFE Tampere pilot to create easily a *CityExplorer* game concept by reusing component already implemented by other STREETLIFE project members. The purpose of the *CityExplorer* game is to encourage using public transport for all sorts of trips. To get users not only to commute to work but also promote public transport for various leisure trips. They get points by visiting different locations. The scenario is described in more detail in D8.2.1 section 4.3.2 [12] second iteration's scenarios and use cases.

Integration of gamification engine into STREETLIFE Tampere pilot included communication about its configuration possibilities, communication about protocols, testing it for the *CityExplorer* game type and interoperability tests between other STREETLIFE Tampere pilot components. Tests included for example these: component unit testing, integration testing, performance testing, fault tolerance testing and Tampere IT-Platform interoperability tests.

7.1.2.3. Current status of the exchange of component

The gamification engine developed by FBK has been successfully integrated into STREETLIFE Tampere pilot by Logica and it will be used in the 2nd iteration pilot starting in October 2015. For the STREETLIFE Tampere pilot the gamification engine was provided as an installation package with associated installation and configuration documentation. It was installed locally in the STREETLIFE Tampere pilot cloud environment. With this example it can be stated that the STREETLIFE integration concept works, STREETLIFE has produced a component which is configurable and which can be reused. There are documented integration points for this component and it matches the STREETLIFE blueprint architecture [2].

It is emphasized that this component allows extending the existing IT-platforms City areas might have as in Tampere pilot one of the key scenarios is that STREETLIFE Tampere pilot builds on the investment the City of Tampere already has made in their Intelligent Transport Systems.

7.1.2.4. Tracking Component

The tracking component is developed in STREETLIFE by DFKI. In its current release, it is available in two different ways:

1. **Library:** the library bundles the functionalities for accessing sensor data from android smartphones and for the configuration of the tracking component. The library consists of java code that can be integrated in android applications.
2. **Stand-alone**: the *TrackMe* app is released as a standalone android application, which packages all the functionalities of the tracking component. It can be installed on android devices with an API level of 15 (Android version 4.0.3 IceCreamSandwich).

One major reason for providing a library and a mobile application is that other components like e.g. mode detection are in need of data before the library has been integrated in the mobility apps of the pilots wanting to make use of the tracking functionalities. In these cases the *TrackMe* app can immediately be utilized in order to experiment with and collect data in determined formats and frequencies. This allows for parallel developments of single components. Additionally the *TrackMe* app can be utilized as is in the pilot experiments.

7.1.2.1. Purpose of the tracking component – BER

In the context of the BER Gamification approach the *Bike Rider* game is realized where players get points for each kilometre driven by bicycle. The tracking component will be used within the STREETLIFE BER mobility app to gather information about the player's itineraries. A back-end component will validate that the bike was used, by processing the tracked data.

Different partners developed the back-end component for bicycle route validation (Siemens) and the *Bike Rider* game as part of the BER mobility app (DFKI) in parallel. The *TrackMe* app was used to generate tracking data for the development of the bicycle route validation, while the BER mobility app was still under further development.

7.1.2.2. Current status of the exchange of the tracking component

The tracking component developed by DFKI has been successfully tested in the STREETLIFE ROV pilot by FBK. The test revealed that the results of the android activity recognition (which is a part of the tracked data) are insufficient for the purpose of distinguishing between cars and buses. Therefore the tracking component will be used as a data supplier for an improved mode detection component, which is currently under development as back-end system by AALTO.

In order to implement the component AALTO in cooperation with the BER pilot (as the BER pilot will conduct file experiments in October) are going to perform tests and data collections with the tracking component. For the tracking component the STREETLIFE integration concept works, STREETLIFE has produced a component which is configurable and which can be reused at different pilots.

7.2. **Vertical Integration Concept**

The diagram below depicts a sample grouping of components across pilot sites for a common goal or use case that is termed as a '*scenario'*. Such scenarios would require components across sites working in cohesion for achieving the goals of the scenarios. In the context of the STREETLIFE integration concept this is termed as ***Vertical Integration***, where applications/components that are deployed and operating in various pilot sites are working together to satisfy the requirements put forth by a scenario that is attached to the goal of the STREETLIFE project.

In the diagram below, for example, let us consider the light green scenario – scenario B. This scenario requires that data sources from the three pilots (bottom datasources layer) are integrated and common services (data integrators and services layer) at the BER pilot connect them to the user interface of the BER pilot. A concrete example here is the scenario where the park and ride data from TRE, ROV are both integrated by the CIP and displayed in the MMECP. Samples of the interaction within this scenario are detailed in the following sections.

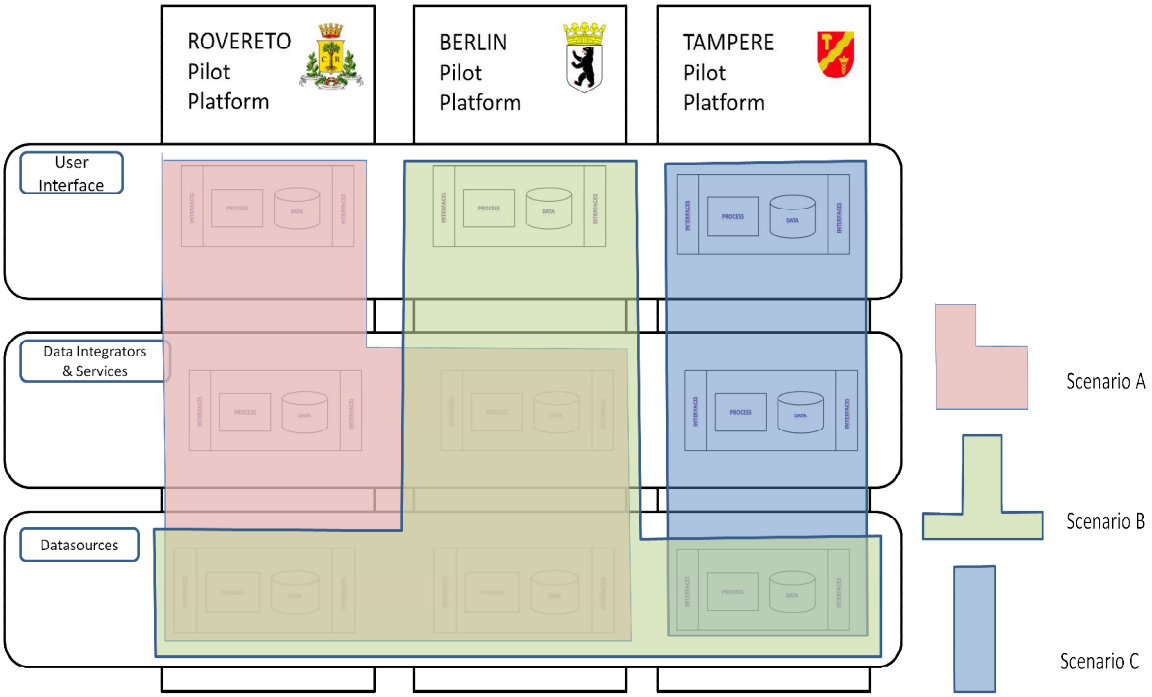

Figure 5: The STREETLIFE Vertical Integration Concept

The following section also describes how a scenario is defined, scoped and implemented.

7.2.1. A Model for defining usage scenarios

The purpose of having a model for defining usage scenarios is to

- Perpetually be aware of the goals of the scenario and the project and not going into too many details of technical implementation before specifying a purpose

- Scope the usage scenario in such a way as to allow for reasonable and feasible implementations within the time and resource limits of the project

The model below is derived from the software industry best practice for defining requirements for software solution systems [13]. This model is generic enough to be applicable for any solution system. It clearly separates the problem and solution domains and connects them. It is documented here only to demonstrate itself as a driver for identifying, integrating the components in STREETLIFE pilots that are required for satisfying a usage scenario.

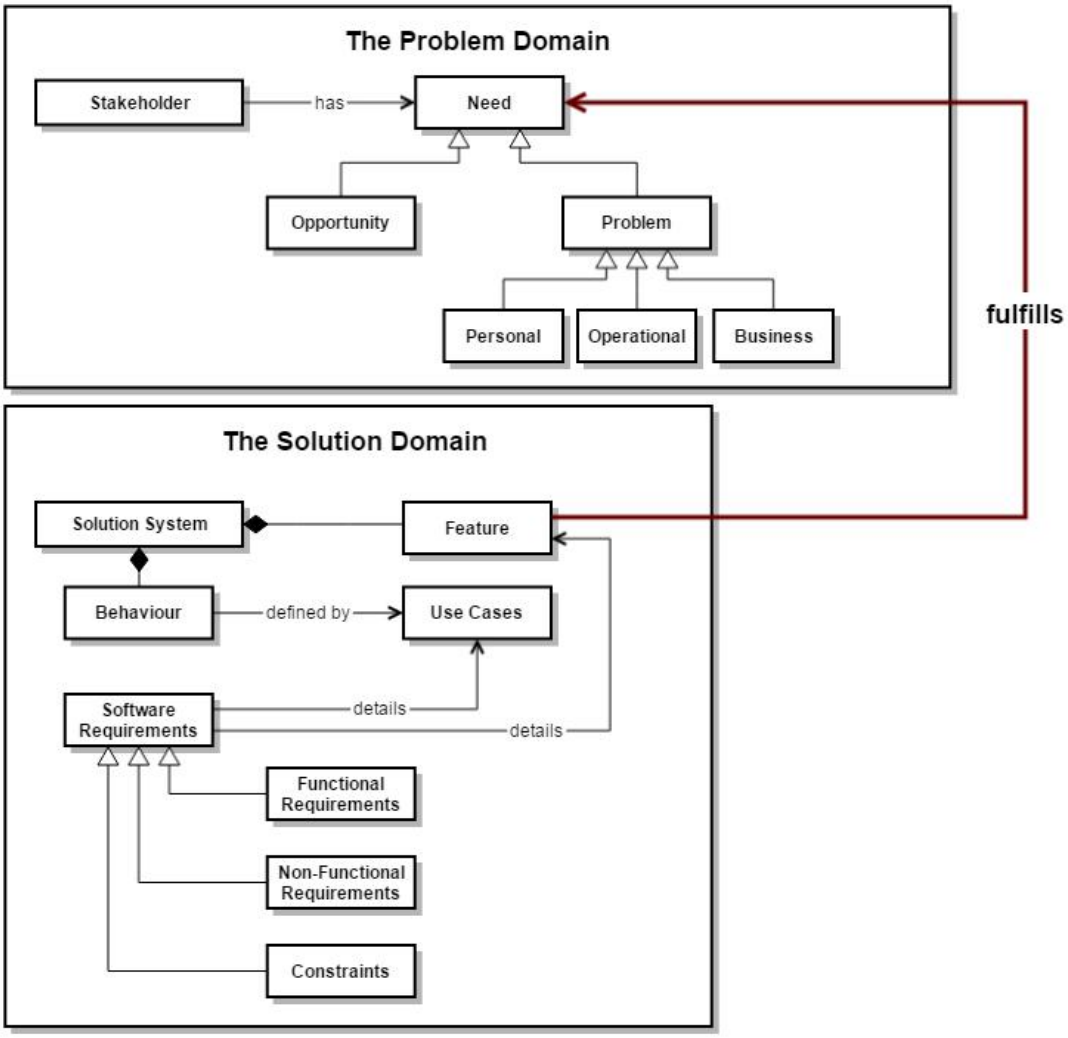

Figure 6: A model for defining scenarios

We decided to take a 'top down' approach. That is, to keep the stakeholder as our origin and build the scenario around the needs of the stakeholder. Mobility Managers, City

Administrators, Citizens of a city are some example stakeholders in the STREETLIFE context. The stakeholders have opportunities or problems that are generalized as needs.

Stakeholders' needs are fulfilled by features of a solution system. This solution system contains features and exhibits behavior. Behavior is defined by use cases. These use cases and features of a solution system if software in nature is detailed in a software requirements document. With the top down approach, there is a rationale for every feature that is to be implemented.

7.2.2. Example Implementation

7.2.2.1. Park and Ride Scenario

The '*park and ride implementation scenario*' is detailed in the WP4 use case specification document [14]. The document talks about all the requirements originating from the top down approach and also applies the model for specifying scenarios mentioned previously.

7.2.2.2. Components and their interactions

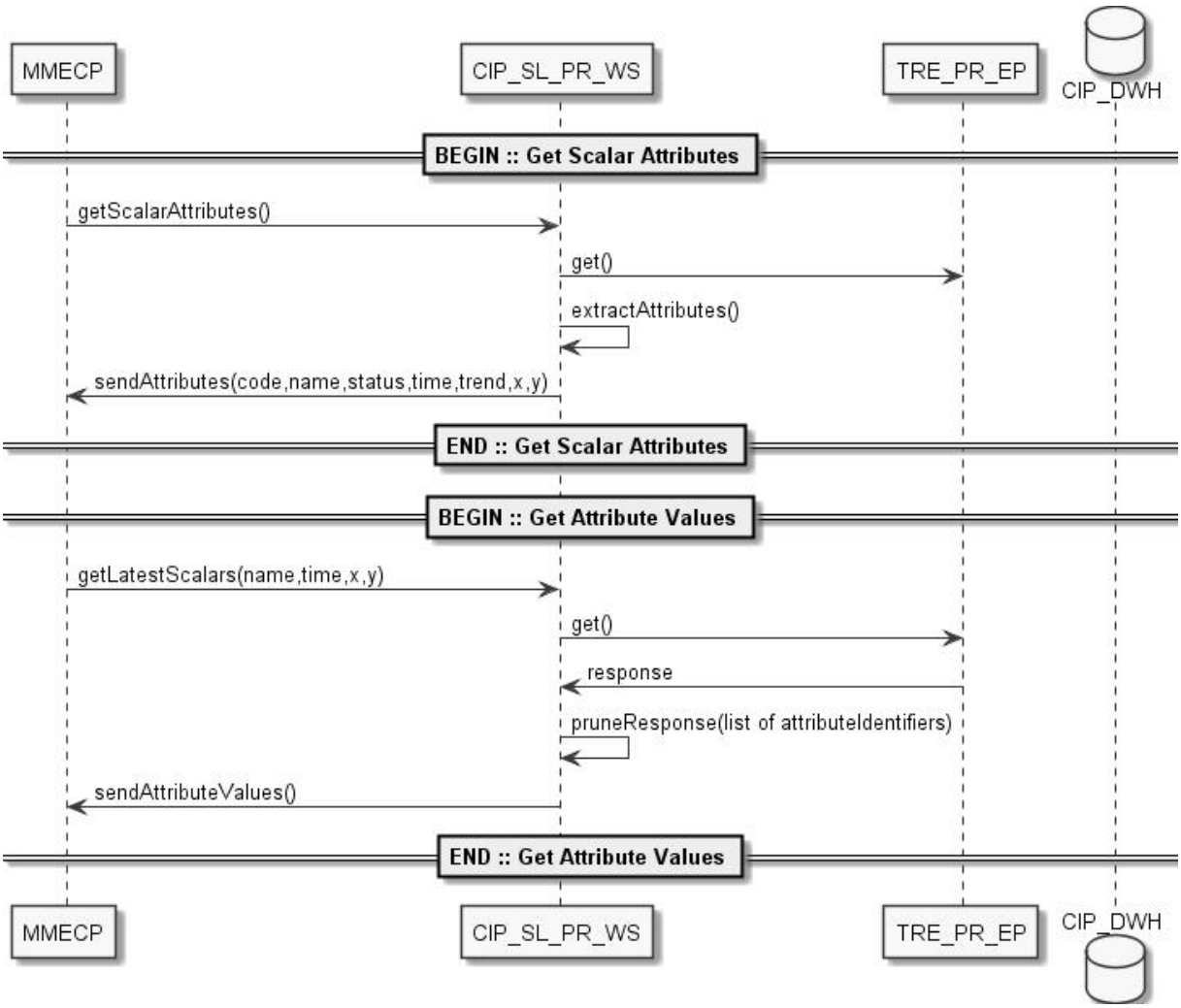

Figure 7: An example component interaction diagram

The MMECP and the CIP understand each other by the common APIs [2] [15] [16] implemented between them. The CIP knows how to access the Tampere Park and Ride Server and translate this to the common APIs [16] so the MMECP can use this information. The CIP also has access to a local storage of the inner city parking data from Rovereto.

7.2.3. Security Architecture with an example implementation

The Security Perspective of the STREETLIFE Blueprint Architecture already defines an abstract set of security architecture components [2]. These components will be refined; their interactions described and related to the use cases scenarios regarding the vertical integration. The central piece of the vertical integration is the CIP. It is used in each pilot either as data source or data endpoint. Thus a special attention on the security mechanisms of the CIP is made in the following chapter.

7.2.3.1. Secure components and their interactions

In this section the mechanism to authenticate sender and recipient for a given request are described.

The following components of the Blueprint Architecture have to be implemented:

Blueprint Component	Implemented By (or Part of)
Authenticator	CIP
User Authenticator	CIP
User DB	CIP

There are in general two ways to achieve an authenticated access in the vertical integration scenario. In a direct authentication the caller is authenticated by the service itself. In a token based authentication the caller is authenticated by a dedicated service. A token issued by this service is used in further request in order to gain access to the STREETLIFE services.

A direct authentication between MMECP and the CIP using TSL/SSL and HTTP Basic Authenticate is described in the following diagram. It shows the interaction pattern described in section from a security point of view. The MMECP establishes in a first step a secure TLS/SSL [17] connection to the CIP. Please note that as a prerequisite the certificate of the CIP has to be exchanged with the MMECP in order to guarantee a trust relationship between MMECP and the CIP. In the next step the MMECP sets the username and the password for the request. The username and password is transferred via HTTP Basic Authenticate mechanisms [18] besides the usual request parameters. The CIP itself verifies the username and password against the entries in the User DB in order to authenticate the requester. After a

[2] This has been accepted to be published as a conference paper at the KMIS conference, Lisbon, Portugal 2015

successful authentication the CIP handles the request and sends the response back to the MMECP.

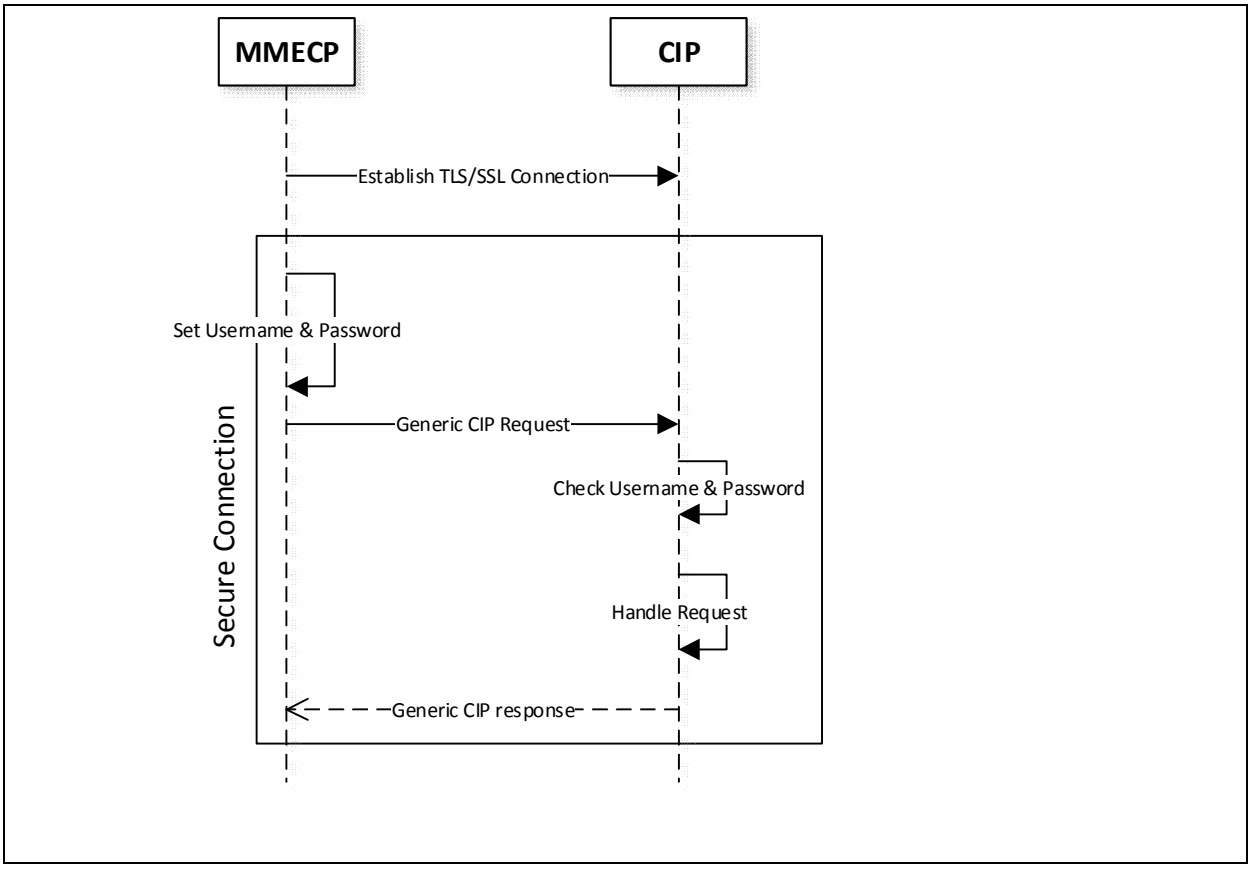

Figure 8: Secure Communication between MMECP and CIP using TSL/SSL and HTTP Basic Authenticate

The Figure 8: Secure Communication between MMECP and CIP using TSL/SSL and HTTP Basic Authenticate demonstrates a secure communication at protocol level using TSL/SSL and HTTP Basic Authenticate. As you can see in a first step a secure connection using TSL/SSL is established. The regular request is extended by the Authorization Header-Element in the HTTP request.

```
*   Trying 194.138.248.80...
* Connected to www.cityintelligenceplatform.siemens.com (194.138.248.80) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* TLSv1.2 (OUT), TLS Unknown, Certificate Status (22):
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / AES256-SHA
* Server certificate:
```

```
*        subject: C=DE; ST=Bavaria; L=Munich; O=Siemens AG; OU=Siemens;
CN=www.cityintelligenceplatform.siemens.com
*        start date: 2014-09-30 00:00:00 GMT
*        expire date: 2015-10-01 23:59:59 GMT
*        issuer: C=US; O=VeriSign, Inc.; OU=VeriSign Trust Network; OU=Terms of use at
https://www.verisign.com/rpa (c)1
0; CN=VeriSign Class 3 Secure Server CA - G3
*        SSL certificate verify result: unable to get local issuer certificate (20),
continuing anyway.
* Server auth using Basic with user 'mobility'
> POST /app-sl-pr/rest/streetlife/parkride/getScalarAttributes HTTP/1.1
> Host: www.cityintelligenceplatform.siemens.com
> Authorization: Basic bxpdHk6iaWxpuY2llusQDEyMz
> User-Agent: curl/7.44.0
> Accept: Application/json
> Content-Type: application/json
> Content-Length: 1
>
* upload completely sent off: 1 out of 1 bytes
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: Apache-Coyote/1.1
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, OPTIONS
< Content-Type: application/json
< Content-Length: 67
< Date: Wed, 09 Sep 2015 13:30:20 GMT
< X-Cache: MISS from ITRPXY.SIEMENS.CO.IN
< X-Cache-Lookup: MISS from ITRPXY.SIEMENS.CO.IN:443
< Via: 1.0 ITRPXY.SIEMENS.CO.IN (squid/3.1.10)
* HTTP/1.0 connection set to keep alive!
< Connection: keep-alive
<
{"attributeIdentifiers":["time, trend, status, code, name1, y, x"]}* Connection #0 to host
www.cityintelligenceplatform.
siemens.com left intact
```

Figure 9: Secure communication at protocol level using TSL/SSL

The technical interface specification of the direct authentication via HTTP Basic Authenticate is described in Table 1. Keep in mind that this specification is an addition to the regular Web-Service specification.

Description	
Protocol	HTTPS
Port	443
REQUEST PARAMETERS	
Path	<path/to/service>
HTTP Method	POST or GET
GET Parameters	<From Web-Service Specification>
HTTP Headers	Authorization [Basic <username:password base64 encoded as

	described in [17]>]
HTTP Body	<From Web-Service Specification>
Examples	-
RESPONSE (Successful)	
HTTP Status Code	200
HTTP Headers	<From Web-Service Specification>
HTTP Body	<From Web-Service Specification>
RESPONSE (Exception)	
HTTP Status Code	401
HTTP Body	Please provide the valid username and password

Table 1: Technical interface specification for the direct authentication to a generic CIP Service via POST

The token based authentication is realized with a Session Cookie between the MMECP and the CIP and is described in the following diagram. It shows the interaction pattern described in section that details components and their interactions from a security point of view using token based authentication. Here the authentication process is decoupled from the application logic provided by the CIP. Thus a Login Service is introduced. This Service is called by the MMECP in advance in order to verify the provided credentials and issue a session cookie which can be used by the MMECP to access any additional service provided by the CIP. The credentials and the session cookies need to be protected against disclosure attacks. Since the session cookie is usually sent in the HTTP Header a secure connection via TSL/SSL mechanisms should be used.

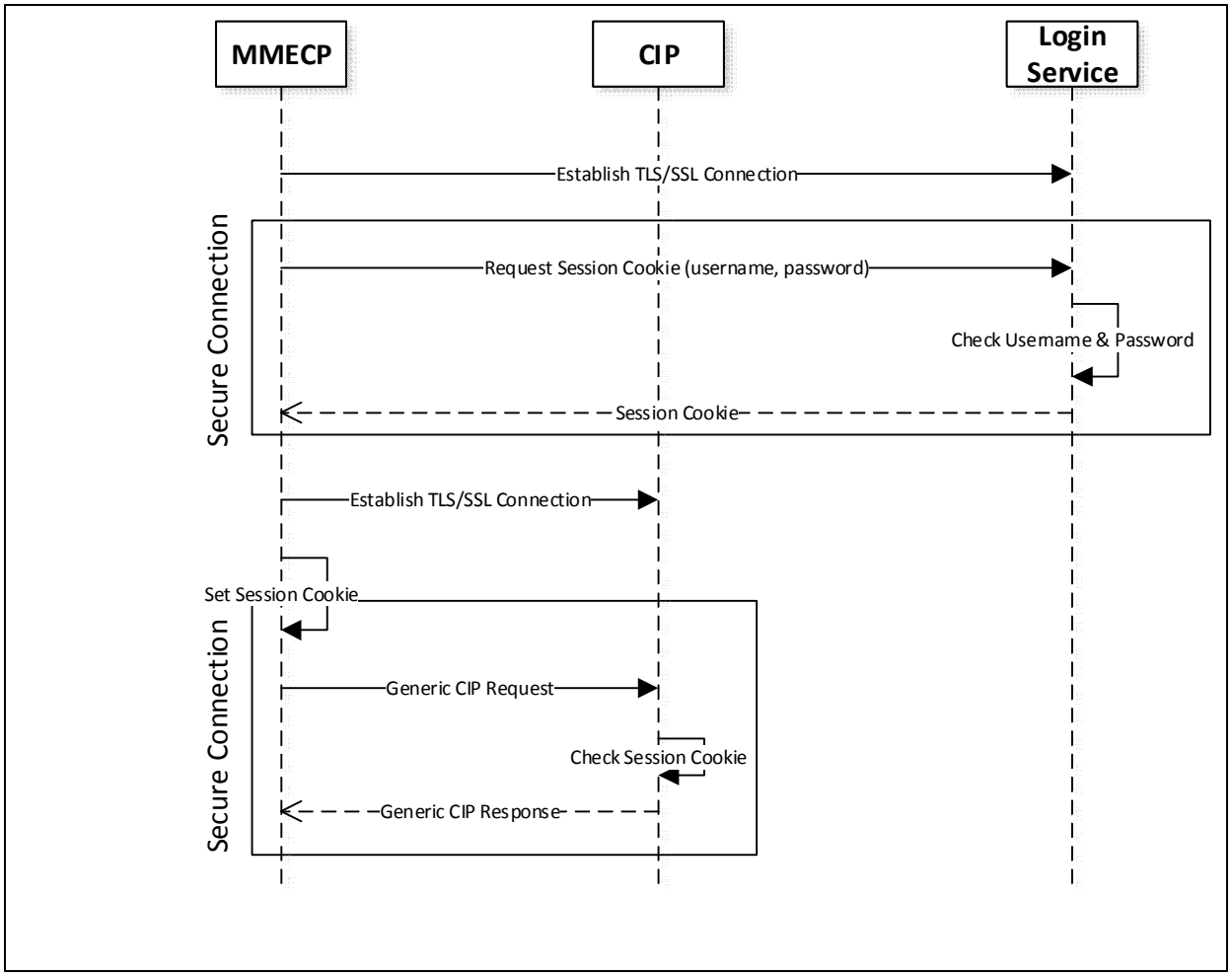

Figure 10: Secure Communication between MMECP and CIP using TLS/SSL and Session Cookies

The technical interface specification for the login service is given in **Table 2** for the HTTP POST method.

Description	
Protocol	HTTPS
Port	443
REQUEST PARAMETERS	
Path	/cip-mobilityguidance/login
HTTP Method	POST
GET Parameters	None

HTTP Headers	Content-Type [application/x-www-form-urlencoded]
HTTP Body	u_name=<username>&u_passwd=<userpassword>
Examples	https://localhost/cip-mobilityguidance/login
RESPONSE (Successful)	
HTTP Status Code	302
HTTP Headers	Content-Type [text/plain]
HTTP Body	Login succeeded. Redirect to index.html
RESPONSE (Exception)	
HTTP Status Code	400
HTTP Body	Login failure. Loop back to login page with error message.

Table 2: Technical interface specification for the login service via POST

8. LESSONS LEARNED

The STREETLIFE project has integrated into multiple existing systems so far. There were quite some obstacles in the way to achieving this. At the end of Year 2 of STREETLIFE considerable material has been collected as lessons learned from this integration especially which kind of approaches promise to be feasible or rather bulky in this context. The approaches explained in the example implementations are considering these lessons learned.

In addition, this chapter addresses some of our approaches in disseminating this knowledge with other EU projects that are working in the same domain as STREETLIFE.

8.1. **STREETLIFE Architecture Integration Concept – applicability to EU Projects**

During the process of the evolution of the STREETLIFE Architecture Integration concept, it became evident to the partners that this could be generalized to a state where it could be reused within other EU projects. The general consensus amongst the STREETLIFE partners is that this could be used as *"a framework for an early approach within EU mobility projects towards an integrated Technical Architecture"*.

At the end of Year 2, the STREETLIFE partners opine that other EU projects (especially in the mobility domain with a similar partner setup as STREETLIFE) could use the steps that were followed and generalized in the chapter "Arriving at the STREETLIFE integration concept" as a reference guide between partners and pilots to quickly conceptualize their own extension and specialization of the STREETLIFE concept.

Further, to disseminate this knowledge STREETLIFE has taken proactive steps to attend and share this Integration concept and approaches along with other contributions from other WPs in STREETLIFE at the following EU project collaboration workshops

- 2015 May 20th - 1st Joint Collaboration Workshop organized by MyWay [19], Berlin
- 2015 September 23rd - SMART MOBILITY SERVICES FOR THE SMART CITY // Architectures and Solutions towards a Service Market Place [20], Frankfurt IAA
- 2015 October 14th – 2nd Joint Collaboration Workshop organized by STREETLIFE [21], Trento

9. INFLUENCING DEVELOPMENTS OUTSIDE OF STREETLIFE AND OUR VIEWPOINTS

This chapter details the recent developments within the European Union within the mobility sector. We consider this as a significantly influencing factor for projects such as STREETLIFE. Further, we will look on the challenge of integration with a bigger perspective and observe these integration problems with lessons learned from the telecom domain.

9.1. **STREETLIFE's Strategic Approach towards a Smart City Reference Architecture**

The STREETLIFE strategic approach is the acceleration of the adoption of a common approach and standard for Smart City interoperability platforms starting with a blueprint and ending with a reference architecture for the mobility domain fostering sustainable solutions! Means – STREETLIFE integrates, within the Smart City architecture concept, the mobility domain (which can be regarded as vertical) into a horizontal smart city IT-infrastructure.

It is clear; a modern "traditional", technical infrastructure covers various city sectors (mobility, energy, water, health, IT, etc.) and is commonly regarded as an important location factor for the economy. Progress is seen with the additional installation of an intelligent horizontal IT Infrastructure connecting all these traditional (vertical) sectors with each other. This way enabling cross-sector data flows, data analysis, better monitoring and control, efficient use of resources, predictions etc. It is expected that new intelligent services arising will strengthen the basis for innovation and creative ideas. A modern information society is seen as a pre-requisite to support fast take-up of smart solutions in cities. An intelligent IT-infrastructure has to potential to enable various stakeholders of a city to participate and facilitate the different vendor solutions to be integrated.

9.1.1. What steps are needed to achieve this?

The transmission of a standard European City with its existing infrastructures and legacy systems into a "Smart City" is a complicated process and needs to be performed in practice (in pilots) and in terms of pilot experiments checking step by step, for each domain where the obstacles are, for example, regarding integration of existing legacy systems in terms of interoperability. In preparation of the standardization, stakeholders of all application sectors (domains) should commonly push standard and reference implementations to identify integration approaches. And this process can be started normally with small cities scaling up to larger ones and the process needs different parallel efforts of above kind to put the solutions in completion. STREETLIFE covers in its pilot concept three sizes of cities.

9.1.2. Smart City Reference architecture to realise "Mobility as a Service"

Above all city sectors the mobility domain especially produces plenty of data-sets and has the potential to more easily exploit data than all other sectors (health, environment, weather,..) so value-added services can easily be created. Also the mobility data and the environmental data is more easily to be used/exploited in terms of privacy than data-sets from other domains like health for example. This is the reason why, for example, Germany has identified especially the mobility sector as one of the initial urban domains to be integrated within a smart city infrastructure. It seems the idea is to roll out the entire digital infrastructure of the smart city from the mobility side, use the IT mobility infrastructure as blueprint for the other domains, exploit the mobility sector as precursor/pioneer/test domain for preparing/rolling out the

entire (including all domains) digital infrastructure of a city/nation. Other domains are planned to follow and integrated later in the horizontal IT-infrastructure. For this reason, for example, a short time ago Germany has united the two historically/formally separated domains of Traffic and Digital Infrastructure into one Federal Ministry called "Federal Ministry of Transport and Digital Infrastructure" promoting Standards, Interoperability, New Services and Open data etc. Another proof of this is that the "*Round Table on Integrated urban Platforms*" promoting a Smart City Reference Architecture [22] organized by the ***"Directorate-General for Communication Networks, Content and Technology Smart Cities and Sustainability"*** taking place September 2015 in the context of the IAA International Car Exhibition Fair in Frankfurt. The same directorate is financing the Integrated Mobility projects that STREETLIFE is in.

Focusing on the mobility sector: In the beginning of 2015 the trend for "*Mobility as a Service*" (MaaS) as a new market approach in the mobility sector came up. This movement was promoted initially in Finland by companies (some of them STREETLIFE partners) like CGI, or the city of Helsinki, and TEKES.

According to TEKES [23] the idea of "Mobility as a Service" goes along with a social change in Europe: "*Urbanisation*", "*Sharing economy*", "*Servicizing*", decreasing car ownership among young, even decreasing willingness to achieve drivers licenses (the young have other priorities and want to be multimodal, spontaneous and continuously connected), demand for sustainability, environmental objectives, disgust of pollution and congestion, technological developments etc.

The objectives constituting the MaaS are

- Global scalable door-to door mobility services without owning a car
- Seamless and efficient flow of information, goods and people both locally and through long distances
- Open ecosystem for information and services

The idea is to sell "personal mobility packages" to people for example on monthly basis and guarantee them transportation from A to B according service level agreements (SLA). These SLA should best include all transportation modes on regional, public transport, taxi, car and ride sharing, bike sharing etc. They should include transport related services like city logistics and home deliveries and include roaming in other countries. In the context of MaaS, the "mobility operator business" is taking over offering users individual door-to-door mobility services, integrating several transport services by utilizing open interfaces to schedules, real-time location data and payment systems for transport services both locally and in long distance transport. Through these digital solutions, users can search and compare the length or other attributes of the routes like secure routes, less-polluted routes, green routes, touristic routes etc., book an pay, follow and get instructions or itinerary adjustments. The digital solutions are scalable to different user profiles and internationally.

We want to hold on that MaaS demand goes very well along with the demand for Smart City IT-architectures – here overarching Reference Architectures with open interfaces to transport services including timetables, real-time location information and payment systems so mobility operators can seamlessly integrate transport services worldwide. This reference architecture for Smart Cities and MaaS should include especially the telecommunication sector's service

architecture as a precursor.

9.1.3. Telecommunication sector service architecture as precursor

After completing a system one knows about the strengths and weaknesses. Often after finalising a system one would like to restart and build it new and differently - this is a natural learning process. It would make sense therefore, certainly, to benefit from the learning process of others and at least not to repeat their mistakes. Let's take a look at the developments the telecommunications sector have gone through in the past 25 years to draw some similarities and extract lessons learned.

Already, in early nineties, the Telecommunication stakeholders asked themselves similar questions as coming up now in the mobility sector: How can the mobile communication provide versatile multimedia services over the internet and the traditional telecommunication networks? How can they keep in check the potentially increase in cost of creating, deploying and managing these services? How can they incorporate flexibility and personalisation in the service software for current and future network technologies? Their answer was, in the early nineties, to define a new software architecture that capitalizes on the latest advances in computer and telecommunications technologies to rationalize the organisation of complex software for services and network management. This process took more than 15 years. The final service architecture they created was based on the four principles – the same are still valid for the Smart City: ***Object-oriented analysis and design, distribution, decoupling of software components & separation of concerns***. The purpose of these principles was similar to the Smart City Reference Architecture: to ensure interoperability, portability, reusability of software components, independence from specific technologies and to share the burden of creating and managing a complex system among different business stakeholders such as consumers, service providers and connectivity providers. [3]

In detail the development was as follows:

- In the beginning of mobile communications, the customer closed a contract with a provider of telecommunications services. This provider could store data on the customer, its contracts, connections, service usage, billing data, agreed service quality, etc. in specific formats. If a customer needed to use a service, he signed up (more precisely did his phone), chose the service and used this then. All services were proprietary and not really secure (only one PIN for the phone and who else was listening in the transmission).
- Next the users got mobile and then the vendors started cooperating. Users needed to use the services from abroad or through foreign providers use (roaming). There was the need to make a minimum of technical agreements. The providers had to trust each other, they had to offer acceptable possible / similar services and they had to create binding, technical interfaces between the IT systems of suppliers.
    - Therefore, on the IT side a reference architecture was defined with well-defined interfaces between the systems of the provider. Providers could exchange information about the users, their contracts and services, usage and billing data.

[3] See: Principles of TINA-C at: http://www.tinac.com/.

  - Since the standardization of data structures in the IT systems of suppliers and existing systems would have been too drastic, emphasis was placed on the identification and specification of mandatory interfaces, so-called reference points, between the IT systems of the various providers (Service Architecture)
  - To ensure the communication taking place within distributed IT systems, a communication middleware was defined, that at least ensures technical interoperability (Computing Architecture, DPE).
- From the perspective of the provider of telecommunication equipment uniform interfaces between IT (software) and the provided hardware and network infrastructure was necessary (Networking Architecture).
- With rising performance of hardware and software the services were getting always more comfortable during the years. This meant that over time the service usage contracts and associated user profiles moved into the focus. Users were enabled to determine what kind of service they like to use, and they could determine over profiles how this specific service shall be used in which kind of kind of specific environment. Consequently, later on these profiles were also used accessing the services over foreign suppliers, so that further adjustments in terms of content, access and safety in particular were required.
- While more players included, the importance of the definition of an associated roles and rights system raise. So in the later time of the development process the “User management”, federated identity, authorization management, etc. became increasingly important.
- In order to offer efficient and resource-saving services, providers started to evaluate usage data on larger scale. Content delivery networks can be cited as an example here. Users like to e.g. stream videos on their devices. Knowing the preferences of videos of users in a specific region, providers are likely to host a video already as close as possible to the customer, at the "Edge" of the network so he does not need to stream it from distant server farms from halfway around the world.
- Related technical architectures and conventions are drafted and by interested parties and organizations (in this example, TINA-C, ITU-T, TMF).

The extract of this example is similar to our situation in the mobility domain now:

- In the beginning the data is kept as proprietary data.
- The cooperation of service providers requires agreements on common IT architectures, interfaces and optionally data structures
- This is followed by increasing personalisation of services – Profiles, increasing need for protection / security
- This is followed by predictive optimization of resources needing further arrangements and adjustments in the infrastructure
- Liability of the agreements requires associated syndicated / standards bodies

Comparing the architectural approach followed by TINA-C and the integration approach described in this chapter, some similarities can be detected. A TINA-C platform provides support for the secure execution of arbitrary telco-services. It factorizes AAA functionality

and provides associated building blocks offering their functionality via standardized interfaces. Every telco-service can offer its functionality via service-specific interfaces. In addition it has to provide a standardized management interface that allows to start, stop, interrupt, resume, etc. the service. Thus the platform mainly focuses on AAA-support and on service management. This approach allows the easy integration and execution of new telco-services. In addition it provides a standardized communication infrastructure (the DPE) that is a kind of distributed operating system for the platform and the hosted telco-services. This approach can be somehow compared with the vertical integration concept in STREETLIFE, or, with other words, STREETLIFE can check if their blueprint and integration architectures comprise all necessary and transferable building blocks identified by TINA-C.

The second major concept introduced by the TINA-C reference architecture is the support for federated platforms utilizing dedicated APIs, the so called reference points. This concept allows secure interworking between services hosted in different but federated platforms. Again, AAA-functionality and service management are performed utilizing general components with standardized APIs and behaviour, and by service-specific components offering standardized management interfaces. This approach can be compared with STREETLIFE's horizontal integration concept, although it provides additional secured service access.

While TINA-C additionally standardizes access to the telco-infrastructure, the hardware that is governed by the telco-services, STREETLIFE probably has to standardize access to the data that are the major input for the STREETLIFE applications.

Summarizing, the telco-community has defined a platform that supports the secure execution of telco-services by the introduction of the minimum run-time environment necessary to satisfy the requirements in the telco-world. The same should be true for the STREETLIFE platform considering the requirements of Smart Cities.

9.1.4. Lessons learned for the Maas and the Smart City - Reference Architecture

- Uniform services are a pipe dream. Service provider in the real world will always be in a competitive situation and collaborate as less as necessary.
- The coordination processes often take long, and not all potential stakeholders have the funds to participate in it. Therefore also here the note: to specify only on what is really required.
- The innovation cycles in ICT are becoming ever shorter. One always has to strongly watch out that no new technology or even proprietary but sexy solutions overtake large market penetration and push comprehensive, scientifically sound solutions against the wall.
- Often, the current 80% solution wins against the never finished 99.9% solution. The developers and users have to be aware that in reality 20% of the possible functionality will not be available.

9.1.5. Reality of the city

In the reality of the city STREETLIFE has discovered many bottlenecks between ICT and "reality of the city" regarding the mobility domain:

- We need more easy to find open data! Mobility data and data from other domains that could enrich the mobility services are not open data so far.
- In the moment the access to existing valuable mobility data of a city is in reality very difficult and time costly and often restricted. The obstacles are: First of all the host of the data needed has to be found. Usually the ownership of the data is extremely unclear and confusing, finding the datasets is very costly in terms of time to receive the data. Even if the data seems to be public, the authorised institution that could provide the data and within the institution and the authorised person has to be searched at length. Reasons for usage even in a research project context have to be very lengthily explained in written forms often multiple times, signatures have to be gained, often data privacy statements have to be signed by other authorised people, licenses have to be cleared and suitable formats and interfaces for data transfer have to be existing and constructed, before a first usage is possible. Often after that procedure it is found that for some reason or the other the data is not suitable for the dedicated research question. This can in most cases first be determined, when the data is there.
- Often there is no tailored, and little real-time information
- Apart from that, there are many interoperability issues through the diverse legacy systems of all domains
- And there are severe privacy issues regarding protection of personal identifiable information and provision of the right to denial to citizens to be part of the smart city.

9.1.6. Clearing obstacles in our way

We can draw from the lessons learned and the reality of the city following obstacles:

a) Various stakeholders from the various domains have possibly differing/conflicting interests that need to be synchronized, prioritized by the Smart City. Since the core processes of Smart Cities require large investments, in can be expected that a small number of big players will try to control the market, using competing “de-facto” standards. Hence, similar to cloud computing, the danger of vendor lock-in situations becomes imminent. In many cities, large investments already have been made. Any selected approach will have to consider those.
b) Protection of Personal Identifiable Information (PII) is an issue since city platforms collect large quantities of behavioral data. It is not clear whether existing standards are compliant with national regulations, nor do standardization bodies’ appropriate competence and ambition to address this issue. Similar issues are related to other regulatory domains (collection of data related to public health, etc.).

9.1.7. What would be the enabling factors?

- Open data, open formats, open protocols, services and interfaces for the vertical and horizontal integration of domains and interoperability of the ICT components. This would help to prevent vendor lock-ins. Development and maintenance of clear standards that map regulatory requirements to the Smart City platforms help

  - standardization bodies to develop standards that take these into account, and
  - vendors and applicants of Smart City solutions to achieve regulatory compliance.
- Screening, recommendation, development and maintenance of domain specific standards would support the design, development and port of STREETLIFE-applications between different instances of the STREETLIFE-platform.

For example within STREETLIFE the Tampere pilot has been actively using CEN standards like SIRI and DATEX [24]which has enabled this pilot to integrate with the existing mobility IT infrastructure within Finland.

- Scalable concepts for the semantic interoperability of urban data (of different domains such as open environmental data, open climate data, health data etc. and access to data of research projects) and information supporting near real-time data analytics and information exchange in heterogeneous environments
- Concepts for data-driven innovation based on trust and ownership of data while at the same time assuring security and privacy in complex urban environments
- Concepts for the resilience and safety of ICT-based urban infrastructures including energy, water and other supply networks while at the same time assuring flexibility in the timely evolution of the ICT components
- Availability of a RA, standardized APIs, OS reference implementations and interoperability testing suites will open the market for SMEs, which cannot provide a full solution, but are able to provide interoperable components covering some elements of the architecture.

To make this reference architecture applicable to many cities and domains it would make most sense to do this best as Open Standard – that in the end allows to compile a map of available standards and in particular identify gaps. The RA ideally should comprise functional key components with all relevant domain interfaces. This is a step to increase interoperability and domain integration within a city and across multiple cities. And the Reference Architecture is the step needed to realize efficient, value-added cross-domain approaches for relevant urban processes and solutions. Since smart city platforms are partially in the scope of the government (public security, safety, privacy) the public sector is in a prime position to enforce the adoption and use of open standards. The very best would be even to foster Open Source (OS) as reference implementation of the platform basis components and interoperability testing suites.

The topic of city interoperability platform should not be limited to the scope of the city. Instead of that, cities have to be ready for integration of infrastructures and services on national and pan-European level. That requires clear national and European roadmaps [22].

10. CONCLUSIONS AND NEXT STEPS

To conclude, the work conducted by WP2 in this year has yielded a Technical Architecture Integration concept that has been applied within the project by exchanging components and describing scenarios that create an orchestrated workflow and execution or components from various pilot sites.

The results of the work and the experiences are also generalized so that these could be used by other EU projects operating in a similar domain and partner/pilot setup such as STREETLIFE to arrive at *"a framework for an early approach towards an integrated Technical Architecture"* within their projects.

For the next steps, in particular for defining the work for the following year in the project, we consider our current progress within the project/pilots and our lessons learned through our experiences so far. Interoperability within the mobility domain is a major topic and we are in the process of defining a roadmap to detail the tasks that are described in the DoW and adapting it to our current progress, know-how and experiences. At a very abstract level, the current Y2's work focussed on *"Integration"* and we foresee that the future Y3 work will focus on *"Interoperability"*.

BIBLIOGRAPHY

[1] WP2, "D2.1 Report on Gap Analysis and Incentive Models," STREETLIFE, Munich, 2014.

[2] WP2, "STREETLIFE D2.2.1 Blueprint Architecture, Security Architecture & Pilot Architectures - Initial," STREETLIFE, Munich, 2014.

[3] J. Meier, "MSDN Blogs > J.D. Meier's Blog > Reference Models, Reference Architectures, and Reference Implementations," Microsoft, 16 February 2011. [Online]. Available: http://blogs.msdn.com/b/jmeier/archive/2011/02/16/reference-models-reference-architectures-and-reference-implementations.aspx.

[4] WP3, "D3.1 - Conceptual models and specifications for mobility data integration," STREETLIFE, 2014.

[5] WP3, "STREETLIFE-D3.2.1-Mobility Data Integration Models and Techniques (initial)," STREETLIFE, 2014.

[6] H. van der Veer and A. Wiles, "ETSI White Paper No. 3 Achieving Technical Interoperability - the ETSI Approach," European Telecommunications Standards Institute, Sophia Antipolis Cedex, 2008.

[7] Salesforce, "Integration Patterns and Practices, Version 34.0, Summer '15," 30 June 2015. [Online]. Available: https://resources.docs.salesforce.com/sfdc/pdf/integration_patterns_and_practices.pdf.

[8] S. Murer and B. Bonati, Managed Evolution - A Strategy for Very Large Information Systems, Springer-Verlag Berlin Heidelberg, 2011.

[9] Fondazione Bruno Kessler, "Smart Campus Gamification Setup," 15 April 2015. [Online]. Available: https://github.com/smartcommunitylab/smartcampus.gamification/wiki/Setup.

[10] WP5, "D5.2.2. End-user Applications, Techniques and Tools : Intermediary," STREETLIFE, Tampere, 2015.

[11] Fondazione Bruno Kessler, "SMART COMMUNITY LAB," 2015. [Online]. Available: http://www.smartcommunitylab.it/.

[12] WP8, "D8.2.1 Achieved Impacts (Initial)," STREETLIFE, 2015.

[13] D. Leffingwell, "IBM Developerworks Rational Library," 2000. [Online]. Available: http://www.ibm.com/developerworks/rational/library/content/RationalEdge/dec00/FeaturesUseCasesRequirementsOhMyDec00.pdf.

[14] WP4, "STREETLIFE UC-P&R - Use Case Park and Ride," May 2015. [Online]. Available: https://extsvnsrv.fokus.fraunhofer.de/svn/cc/elan/streetlife/documents/.

[15] WP4, "Work Package 4 - Interface Specification," May 2015. [Online]. Available: https://extsvnsrv.fokus.fraunhofer.de/svn/cc/elan/streetlife/documents.

[16] J. K. e. al., "A Generic Interface Specification for Standardized Retrieval and Statistical Evaluation of Spatial and Temporal Data," in *KMIS, The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Lisbon, 2015.

[17] T. Dierks, “The Transport Layer Security (TLS) Protocol,” August 2008. [Online]. Available: https://tools.ietf.org/html/rfc5246.

[18] J. Franks, “HTTP Authentication: Basic and Digest Access Authentication,” June 1999. [Online]. Available: https://tools.ietf.org/html/rfc2617.

[19] The MyWay EU Project, “First Collaboration Workshop STREETLIFE, MyWay, PETRA, TEAM, MoveUs, MoveSmart & Simpli-City @ Berlin,” 20 May 2015. [Online]. Available: http://myway-project.eu/images/1st_Collaboration_Workshop_Agenda_v01.pdf.

[20] The Four European Union Projects, “INTERNATIONAL WORKSHOP,” 23 09 2015. [Online]. Available: http://workshop.fgm.at/docs/workshop_agenda.pdf.

[21] The STREETLIFE EU Project, “STREETLIFE News & Events,” 14 10 2015. [Online]. Available: http://www.streetlife-project.eu/news-events.html.

[22] E. 2. Initiative, “MOU Towards Open Urban Platforms Smart Cities and Communities,” 05 06 2015. [Online]. Available: https://ec.europa.eu/digital-agenda/en/news/memorandum-understanding-towards-open-urban-platforms-smart-cities-and-communities.

[23] S. Heikkilä, “Mobility as a Service,” Tekes, the Finnish Funding Agency for Innovation, 2015. [Online]. Available: http://www.eltis.org/sites/eltis/files/case-studies/documents/mobility_as_a_service_heikkila.pdf.

[24] Intelligent Transport Systems (ITS), “The official CEN/TC 278 website,” 2015. [Online]. Available: http://www.itsstandards.eu/.