



FP7-SMARTCITIES-2013

STREETLIFE

Steering towards Green and Perceptive Mobility of the Future



WP4 - Mobility Management and Emission Control Panel

D4.2.1 – Mechanisms and tools for mobility management and emission control (initial)

Due date: 30.09.2014

Delivery Date: 07.10.2014

Author(s): Benjamin Dittwald, Silke Cuno, Jens Kohlmorgen, Kevin van Bernum, Björn Schünemann (Fraunhofer), Birgit Obst, Veronika Brandstetter (Siemens), René Kelpin (DLR), Davide Frigeri, Nicola Perri, Francesco Avesani, Alberto Merigo (Caire), Christine Schwarz (VMZ)

Partner(s): Fraunhofer, Siemens, DLR, Caire, VMZ

Editor: Benjamin Dittwald (Fraunhofer)

Lead Beneficiary of Deliverable: Fraunhofer

Dissemination level: Public

Nature of the Deliverable: Report

Internal Reviewers: Michael Bahr (Siemens), Giuseppe Valetto (FBK)

EXECUTIVE SUMMARY

We start with the State of the Art section (2) on the evaluation of comparable tools to the MMECP. It is shown how far STREETLIFE or in particular the STREETLIFE MMECP can enhance such tools. In section 3 of this deliverable the derived scenarios and use cases from deliverable D6.1 [1] for each pilot site we describe. The Key Performance Indicators (KPIs) important for the specific pilot scenarios and use cases are named. Section 4, Methodologies, is a main section of this deliverable. It describes the methodologies that will be used to achieve the goals of the MMECP. Section 5 describes the architecture of the MMECP component. It contains an architecture description including sequence diagrams for a set of main activities of the MMECP. Section 6 of this deliverable lists all technologies and frameworks that are used for the realization of the MMECP. The conclusion section (7) describes the lessons learned so far and the expectations on the further development of the MMECP.

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

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

© Copyright in this document remains vested with the STREETLIFE Partners

Mechanisms and tools for mobility management and emission control (initial)

Table of Contents

EXECUTIVE SUMMARY.....	2
LIST OF FIGURES	8
LIST OF TABLES	9
1. INTRODUCTION AND MOTIVATION	10
1.1. STREETLIFE FUNCTIONAL ELEMENTS	12
2. STATE OF THE ART	14
2.1. IMTECH TRAFFIC & INFRA - URBAN TRAFFIC.....	15
2.1.1. <i>Traffic Management</i>	15
2.1.2. <i>Parking Management</i>	15
2.1.3. <i>Conclusion</i>	16
2.2. SIEMENS TRAFFIC CONTROL CENTRE PLATFORM	16
2.2.1. <i>Conclusion</i>	16
2.3. BERLIN TRAFFIC CONTROL CENTRE.....	16
2.3.1. <i>Conclusion</i>	17
2.4. OTTAWA TRANSPORTATION SYSTEM MANAGEMENT.....	18
2.4.1. <i>Conclusion</i>	18
2.5. TRAFFIC MANAGEMENT IN QUEENSLAND	19
2.5.1. <i>Conclusion</i>	19
2.6. ECONOLITE ITS SYSTEMS	20
2.6.1. <i>Conclusion</i>	20
3. EMISSION CONTROL PANEL PILOT SCENARIOS.....	21
3.1. BER SCENARIOS	21
3.2. ROV SCENARIOS	22
3.3. TRE SCENARIOS	23
4. METHODOLOGIES	23
4.1. STREETLIFE KPI CALCULATION AND VISUALISATION.....	23
4.2. SPECIFIC EMISSION CONTROL PANEL KPIS	24
4.2.1. <i>Modal split forecasting</i>	25
4.2.2. <i>Modal availability</i>	32
4.2.3. <i>Resource Balance</i>	40
4.2.4. <i>Distance to Modes</i>	45
4.2.5. <i>Park & Ride</i>	47
4.2.6. <i>Bike sharing</i>	55
4.3. GUIDANCE.....	60
4.3.1. <i>ROV example for giving guidance</i>	60
4.4. SIMULATION.....	63
4.4.1. <i>Berlin simulation scenario</i>	63
4.4.2. <i>ROV simulation scenario</i>	68
4.4.3. <i>TRE simulation scenario</i>	72

5. DESIGN OF MMECP	72
5.1. MMECP COMPONENT DIAGRAM	72
5.2. GET HISTORIC DATA SEQUENCE DIAGRAM.....	74
5.3. GET FEED ENTRY SEQUENCE DIAGRAM.....	75
5.4. SHOW STATISTICS SEQUENCE DIAGRAM	76
5.5. CREATE AND POST GUIDANCE SEQUENCE DIAGRAM.....	77
5.6. A GENERIC INTERFACE SPECIFICATION FOR STANDARDIZED RETRIEVAL AND EVALUATION OF SPATIO- TEMPORAL DATA	79
5.6.1. A simple example using lists of conditions	84
6. EMISSION CONTROL PANEL TECHNOLOGIES AND FRAMEWORKS	87
6.1.1. Intelligent Test Evaluation Framework.....	87
6.1.2. VsimRTI.....	87
6.1.3. Cube.....	87
6.1.4. Carneades.....	87
6.1.5. Java	87
6.1.6. OpenLayers.....	88
6.1.7. ATOM.....	88
6.1.8. RapidMiner.....	88
6.1.9. R.....	89
6.1.10. Google Maps	89
6.1.11. OpenStreetMap.....	89
6.1.12. City Intelligence Platform.....	89
6.1.13. FIWARE.....	89
7. CONCLUSION.....	90
APPENDIX A: LITERATURE	91

ABBREVIATIONS

AGPL	GNU Affero General Public License
API	Application Programming Interface
APP	Atom Publishing Protocol
ASF	Atom Syndication Format
AtomPub	Atom Publishing Protocol
BER	STREETLIFE Berlin-Pilot
CO Services)	Confidential, only for members of the Consortium (including the Commission Services)
CIP	City Intelligent Platform
D	Deliverable
DoW	Description of Work
FCD	Floating Car Data
FP7	Seventh Framework Programme
FLOSS	Free/Libre Open Source Software
GWT	Google Web Toolkit
GUI	Graphical User Interface
HLA	High Level Architecture
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IPR	Intellectual Property Rights
ITEF	Intelligent Test Evaluation Framework
KPI	Key Performance Indicator
MGT	Management
MS	Milestone
MMECP	Mobility Management and Emission Control Panel
MC	Modal Choice

NUTS	Nomenclature of Territorial Units for Statistics
OS	Open Source
OSS	Open Source Software
OSM	OpenStreetMap
O	Other
OD	Origin / Destination
PRIS	Parking Route Information System
P	Prototype
PU	Public
PM	Person Month
R	Report
ROV	STREETLIFE Rovereto-Pilot
RTD	Research and Development
TIC	Traffic Information Centre
TMC	Traffic Message Channel
TSM	Transportation System Management
ToE	Type of Event
ToP	Type of Persons
TRE	STREETLIFE Tampere-Pilot
UTMC	Urban Traffic Management Control
VMS	Variable Message Sign System
WP	Work Package
XML	Extensible Markup Language
Y1	Year 1

PARTNER

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

LIST OF FIGURES

Figure 1: SREETLIFE functional blocks	10
Figure 2: Functional elements from the mobility management and emission control (MMECP) perspective.....	12
Figure 3: The STREETLIFE context diagram	14
Figure 4: example of Traffic Management console in Ottawa.....	18
Figure 5: Example of the Queensland Traffic Information console.....	19
Figure 6: MMECP GUI draft	24
Figure 7: Forecasting process.....	28
Figure 8: Decision tree generated by R	29
Figure 9: Decision tree generated by RapidMinder	30
Figure 10: Argumentation graph constructed by Carneades	32
Figure 11: Average daily profile pattern by rolling window	38
Figure 12: Daily profile pattern by weather condition.....	39
Figure 13: Change in availability profiles caused by event	40
Figure 14: Example for bike sharing resource balance calculation	43
Figure 15: Long term forecast for expected modal availability	44
Figure 16: Short term forecast for expected modal availability.....	44
Figure 17: Visualization example for “Mode distance calculation”	45
Figure 18: Occupancy rate map in Rovereto (Source: Urban Mobility Plan of Rovereto).....	55
Figure 19: Generic guidance process	60
Figure 20: Area of the BER simulation scenario (Source: http://www.openstreetmap.org/) ..	64
Figure 21: input data needed for the preparation of the BER simulation scenario	65
Figure 22: Traffic flow on a specific road segment (Source: Urban Mobility Plan of Rovereto)	69
Figure 23: MMECP Architecture.....	73

Figure 24: Get historic data sequence diagram	75
Figure 25: Get Feed Entry sequence diagram	76
Figure 26: Show statistics sequence diagram.....	77
Figure 27: Create and post Guidance sequence diagram	78

LIST OF TABLES

Table 1: Task affiliation	11
Table 2: Responsibilities of recommendation for urban mobility service adaptation and adjustment functional elements.....	13
Table 3 PMML transformation elements	31
Table 4: Bike sharing data.....	34
Table 5: Event data.....	35
Table 6: Weather data	36
Table 7: Data overview for resource balance evaluation	41
Table 8: Data for Mode distance calculation	46
Table 9: Data for Park & Ride	49
Table 10: Data for experimental park & ride evaluation	52
Table 11: Data for bike sharing.....	57
Table 12: Cost factor for traffic estimation	61
Table 13: Access cost indicator.....	62
Table 14: BER simulation KPIs	66
Table 15: ROV scenario key performance indicators	71
Table 16: Example result of a query	85

1. INTRODUCTION AND MOTIVATION

This is the initial version of the “Mechanisms and tools for mobility management and emission control” deliverable. It describes the current status of the concept for the Mobility Management and Emission Control Panel (MMECP), how it is planned to be realized, which methodologies will be used and a concrete software architecture that is used for the implementation.

Figure 1 shows the STREETLIFE functional blocks. This deliverable describes functionalities that are belonging to the blue upper left blocks.

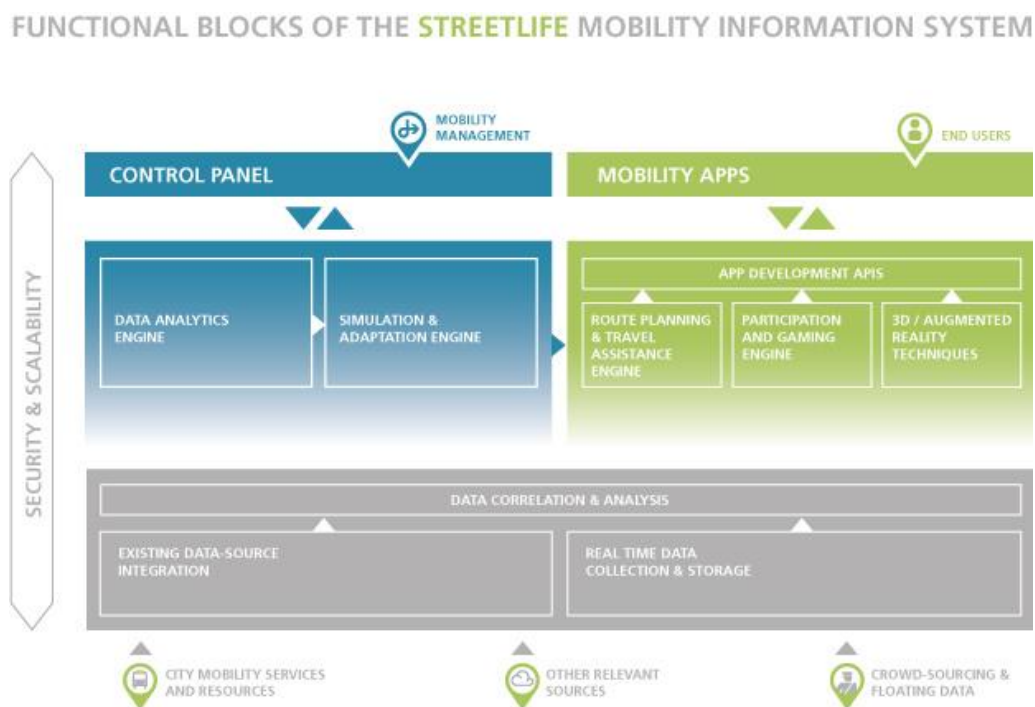


Figure 1: SREETLIFE functional blocks

The Mobility Management and Emission Control Panel WP is structured in 4 task:

- T4.1 Global data view & analytics
- T4.2 Simulation based data forecast
- T4.3 Mobility policies and strategies modelling
- T4.4 Mobility management and emission control panel

The tasks are connected to each other and have overlaps. One cannot easily decide whether a concrete to-do belongs to e.g. T4.1 or T4.3. Thus this deliverable is not structured like the defined tasks in the DoW. To provide a clear understanding that each task is handled, Table 1 gives an overview that show which section covers which tasks. The table covers each section of this deliverable and a more detailed view for section 4, the methodologies.

Table 1: Task affiliation

	T4.1	T4.2	T4.3	T4.4
2 State of the art	✓		✓	✓
3 Emission control panel pilot scenarios	✓		✓	✓
4 Methodologies	✓	✓	✓	✓
4.1 STREETLIFE KPI calculation and visualisation	✓			✓
4.2 Specific emission control panel KPIs	✓		✓	✓
4.3 Guidance			✓	✓
4.4 Simulation		✓		
5 Design of MMECP	✓	✓	✓	✓
6 Emission control panel technologies	✓	✓	✓	✓
7 Conclusion	✓	✓	✓	✓

The STREETLIFE system aims to reduce the CO₂ emission by motivating citizens to use transport modes for their daily travelling, which produce low or no CO₂ emissions. The intention of the MMECP is to give mobility managers a tool to make this possible. It is not the intention to provide a solution that reduces the CO₂ emission in a short time, e.g. a year. We think that this change must come from the citizens. The citizens need to be convinced that it is not a disadvantage to use the bike or the public transport to reach their destinations. Citizens have to confidence in that there is always a comfortable way of coming home, and it doesn't matter which time it is or how the weather is and what is happening in the city. The MMECP makes mobility management more efficient, so that the mobility services and the possibilities for multimodal mobility are objectively improved for the citizens. By combining smart technology with the urban mobility data of smart cities, we are developing a system that enables mobility manager to give this confidence to the citizens.

To provide an overview of the MMECP the following section describes the functional elements of STREETLIFE [2].

1.1. STREETLIFE functional elements

The STEETLIFE functionality addressed in this Deliverable pertains to monitoring and analysis of mobility resources and carbon emissions. Furthermore, the functionality described in this Deliverable addresses the simulation and adjustments of mobility resources and policies by using forecasting methods.

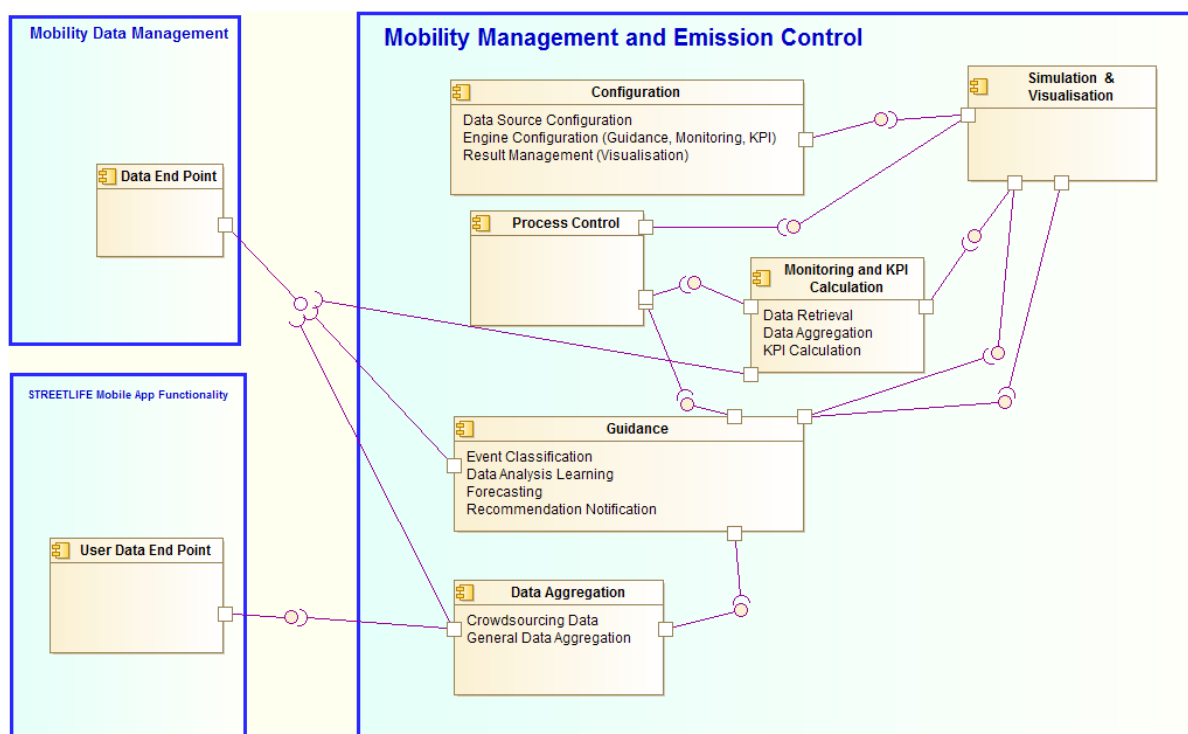


Figure 2: Functional elements from the mobility management and emission control (MMECP) perspective

The purpose of this functional element is to create forecasts that are translated into mobility recommendations (i.e. guidance) for mobility managers and service operators to adjust their

services to the actual current mobility needs. The resulting recommendations are presented in a visualised manner. The functionality residing here is event triggered and controlled by a process controller.

Table 2: Responsibilities of recommendation for urban mobility service adaptation and adjustment functional elements

FUNCTIONAL ELEMENTS	RESPONSIBILITIES
Guidance	<p>Provides interdisciplinary data analytics combining crowdsourcing data, mobility data and additional correlated data.</p> <p>Derivation of guidance recommendation.</p> <p>Post processing for result visualisation and notification.</p> <p>Allows mobility managers to decide how compelling the alert notification is on a numeric scale.</p>
Event Classification	<p>An event is defined as a thing that happens in a certain area at a certain time. Event classification indicates the kind of event to deal with.</p>
Data Analysis Learning	<p>Facilitates machine learning mechanisms for creating guidance to be used in decision making.</p> <p>Adjusts forecasting mechanisms.</p>
Forecasting	<p>Predicts the mobility situation for a point in time based on historical data.</p> <p>Is capable of creating and editing model policies.</p> <p>Gives the possibility to inject policies affecting the planning of journeys.</p> <p>Transport mode availability forecast.</p>
Recommendation Notification	<p>Notifies the mobility management office for further analysis and possible action.</p>

2. STATE OF THE ART

In the Smart City intermodal and ecological oriented traffic control management is thought to take place in a city cockpit. It is seen as a fixed instance within the city governance. The interrelated activities performed here should follow an integrated urban management approach by bringing together different urban domains for example traffic management and environment protection. The cockpit includes an emission control panel and provides the requested context sensitive view to the mobility managers, which they need to track on the performance of the mobility in the city. Such management activities can be termed as an “urban process”. The term derives from the recognition of the city as an organisation and complex system that needs to be managed correctly for example regarding its mobility demands and its traffic flows. Such a process is divided in phases: initiation, planning, execution, monitoring and controlling, optimization and closing. In case of the STREETLIFE concept the process of mobility management is organised in form of a closed control loop of analyses, integration, simulation, evaluation and improvement.



Figure 3: The STREETLIFE context diagram

This is done to ensure that the mobility management fulfils all tasks required to achieve its objectives and KPIs.

Visualization and analyses of the data that supports the process takes place in Smart City cockpit, where data from the database lying below converges. It is characterized through the collection and integration of the data of different application areas like air and water infrastructure, waste management, mobility, finances, health, economy, environment, demography etc.

The results of the analysis and simulations of that data are used to address fundamental problems and challenges of all city operations, planning, and development. It is envisioned that the cockpit discovers effects of different precautions in cause of events and learns to anticipate and avoid them. It is envisioned that it is capable of generating proposals for decisions – possibly based on rule-engines. Also with the help of the cockpit enhancements of the infrastructure as well as calculations and anticipations of the future are developed.

The capability of such cockpits towards partially automatic conflict management is in discussion. Fully automatic conflict resolutions are not envisioned as good, since conflict solution is always a matter of fair human consideration and cannot be left to rule based systems. At this point human consultation and discussion running in parallel is of importance and included in the urban process.

The market for Smart City Cockpits is currently developing [3]. Global companies like SAP, Cisco, IBM and Siemens and others are working at it and will offer sooner or later intelligent solutions for the Smart City analysis.

This section provide the results of the evaluation that was undertaken from MMECP partners to indicate the city cockpits currently available, how they work and what functionalities they provide. This is an exemplary list and is not complete. The intention of the MMECP is not to reinvent the wheel by re-implementing already in-use technologies. So each sub-section concludes emphasising what the MMECP delivers compared to the evaluated cockpit or where enhancements on side of the MMECP are seen.

2.1. Imtech Traffic & Infra - Urban Traffic

A big European actor for urban mobility management is Imtech Traffic & Infra [4]. It provides technologies for traffic- and parking management.

Regarding the city cockpit Imtech Traffic & Infra makes software solutions and a set of tools for traffic and parking management available.

2.1.1. Traffic Management

Imtech Traffic & Infra provides software and hardware components to control and monitor the traffic in urban areas. They have developed peripherals like city lights and sensors to be integrated into the road. The main two software components within the Urban Traffic Management Control (UTMC) System are called “ImFlow” and “Scoot”. ImFlow literally translates policies into operational measures. This is done by changing mobility policies in real-time, to react on certain situations. One can create a set of policies, which is then called a scenario. For each policy a level of importance can be assigned. Scoot is an algorithm with real-time capabilities for adaptive traffic control. They also provide a cooperative ITS Platform to improve safety, air quality and traffic flow. This is done by exchanging information between vehicles and infrastructure. That platform is not only addressing the traffic manager, but citizens as well. An example is an assistant for drivers that recommends the driving speed to avoid stops at red traffic lights.

2.1.2. Parking Management

Imtech Traffic & Infra provides a solution to shorten the time to find parking lots. To achieve this, they developed a software component named Parking Route Information System (PRIS). PRIS uses data from the parking locations or data that is combining traffic situations and time to the parking location. This helps municipalities managing the parking situation during busy periods in the city.

2.1.3. Conclusion

Imtech Traffic & Infra provides functionalities for a modern smart city cockpit, covering the traffic management and parking management. As it is not hosting a public accessible instance this evaluation concentrated on the information that is published on the Imtech traffic & Infra website. These solutions rely on the presence of extensive and expensive infrastructure and hardware in the city system. Avoiding such expensive deployments is one of the main motivations of STREETLIFE, which tries to obtain similar results only with SW technology and what is already available to the cities

2.2. Siemens Traffic Control Centre Platform

Siemens' Traffic control center platform Sitraffic Concert, Sitraffic Scala and Sitraffic Guide integrates the formerly separate worlds of traffic management, traffic control and parking guidance. This enables customized system solutions to be developed, which provide the flexibility and functions that are required for the specific traffic situation of a given city.

Sitraffic Concert offers a traffic control center for integrated urban traffic management. It collects precise, important traffic data and provides a reliable overview.

Sitraffic Scala is Siemens' modular traffic computer system. It facilitates the daily tasks of the traffic control center by efficiently mapping the operating processes of traffic lights based on the relevant application.

Up to 40% of inner-city traffic consists of vehicles looking for somewhere to park. The intelligent Siemens parking guidance systems help to improve in this area. Sitraffic Guide monitors the occupancy of parking lots and multi-story car parks and passes the information on on-street guidance panels and other information platforms such as the city's Internet portal. There are numerous systems and components in road traffic technology to ensure a smooth exchange of data between different systems and components. The Siemens control centers comply with the required standards and thus offers maximum integration.

2.2.1. Conclusion

Sitraffic product family from Siemens offers compatible modules to be combined in many ways on top the same database. Equipped with standard open interfaces, the traffic control centres can also be networked very well with existing technologies. For the STREETLIFE project the access and integration to these technologies will be realized and evaluated via Berlin traffic control centre for BER pilot. Thus STREETLIFE provides a common API that connects STREETLIFE with data mobility data, which is stored in the Berlin traffic control centre.

2.3. Berlin Traffic Control Centre

With its traffic control and traffic information centre the city of Berlin distinguishes itself as one of the leading cities for traffic and mobility management in Europe. The Berlin Traffic Control Centre is one of the largest state-of-the-art traffic control centres in Europe. The Berlin Traffic Control Centre monitors and - when indicated - manually activates the traffic lights (also "light signal systems") in Berlin at about 2.000 intersections, as well as- nine Variable Message Sign Systems (VMS) mainly on the Berlin motorways. It monitors the

traffic situation on more than 1.500 km of roads, and transmits traffic information of the regional reporting office (TMC procedure).

Traffic information is provided by the traffic information centre (TIC), which is an incorporated unit of the Traffic control centre. TIC's web portal is a comprehensive mobility information service for the city of Berlin. www.viz-info.de offers a wide range of cost-free information services for the general public, the media, and the economy with real-time information on road and rail traffic in Berlin.

These services are based on the current car traffic situation that is detected by means of a combination of Floating Car Data (FCD) and permanent detectors in the main street network of Berlin. The data and the resulting traffic map are updated every five minutes. The TIC collaborates with the Berlin traffic control centre in order to provide the media with reliable traffic information. radioBerlin 88.8 radio station is broadcasting live from the traffic information centre. The goal is to relay reliable and timely information on congestion and incidents to the public. In case of disruptions, the information services show alternative ways of reaching your destination faster and stress-free.

In the past: traffic management was focused on safety and flowing car traffic

New Approaches:

- The recent focus is on multimodal information including all transport modes (e.g. multimodal mobility panels, information on real time travel time in public transport in VMS in streetnetwork)
- Monitoring the quality of traffic and the subsequent impact on air pollution, noise and safety (iQ traffic), limited to monitoring, no control or management strategies
- The environment-oriented Traffic Management for hotspots (traffic control measures are triggered when air pollution limit values are exceeded) is currently being tested, limited to air pollution and noise, not focussing CO₂
- Information strategies target on up-to-date information of the public via internet and radio. Mobile multimodal information services are being tested
- Intermodal mobility platforms for intermodal mobility information with integrated booking and accounting functionalities are presently developed and await market entrance.

2.3.1. Conclusion

First steps towards an environmental traffic and mobility management are being implemented in BER, monitoring tools are available for air pollution. Measures to reduce traffic induced CO₂ emission are mainly strategic and focus on a long-term change of the mobility habits towards public transport and zero-emission mobility. These will be addressed by the STREETLIFE system, by monitoring the CO₂ emissions and providing tools to motivate citizens to use transport modes that produces less or no CO₂ emission. As the VMZ is a partner of the STREETLIFE project, the collected data from the Traffic Control Centre can be used in the BER pilot for the forecasts and simulations.

2.4. Ottawa Transportation System Management

The Ottawa TSM was developed with the goal of providing residents with sustainable transportation management and traveller information systems that supply safe, reliable, and accurate information in a timely and cost effective manner, see [5]. An important part of this strategy consists of an interactive Traffic Web site where citizens can retrieve real time information about events, constructions, parking spots, Park & Ride solutions, etc. on the city map.



Figure 4: example of Traffic Management console in Ottawa

Coupled with the Web site there's also a Traffic Management console where the user is enabled to send notifications, add navigation and other features like service updates, and launch mobile applications. In the menu on the left side there are links to other very interesting features: "Ottawa Nav", that is the routing app for residents of the area, "Ottawa Ride Match" that is a link to a website aimed at helping and improving car pooling in the City, and "OC Transpo", a link to the Public Transportation Company's website and planner.

2.4.1. Conclusion

According to the Web site of the system, there are some things that can be very interesting for the STREETLIFE project to look at: the visualization on the city map of accidents, construction works, and Park & Ride structures are features that will be part of the

STREETLIFE cockpit also. The Ottawa website has put links for other functionalities like car pooling, the routing app and public Transport, while in STREETLIFE the goal is to visualize every available piece of information on the same map in order to improve the performance of other means of transportation different from private cars.

2.5. Traffic Management in Queensland

Queensland, Australia, is another region with a website City Cockpit, see [6]. Its Web site has various features: a router, and the possibility to add to the map the location of events, accidents, closures and restrictions, road works and building sites.

It is not clear if the Web site for the citizens is coupled with a similar system for traffic management aimed at the traffic manager's needs, and with some more functionalities.

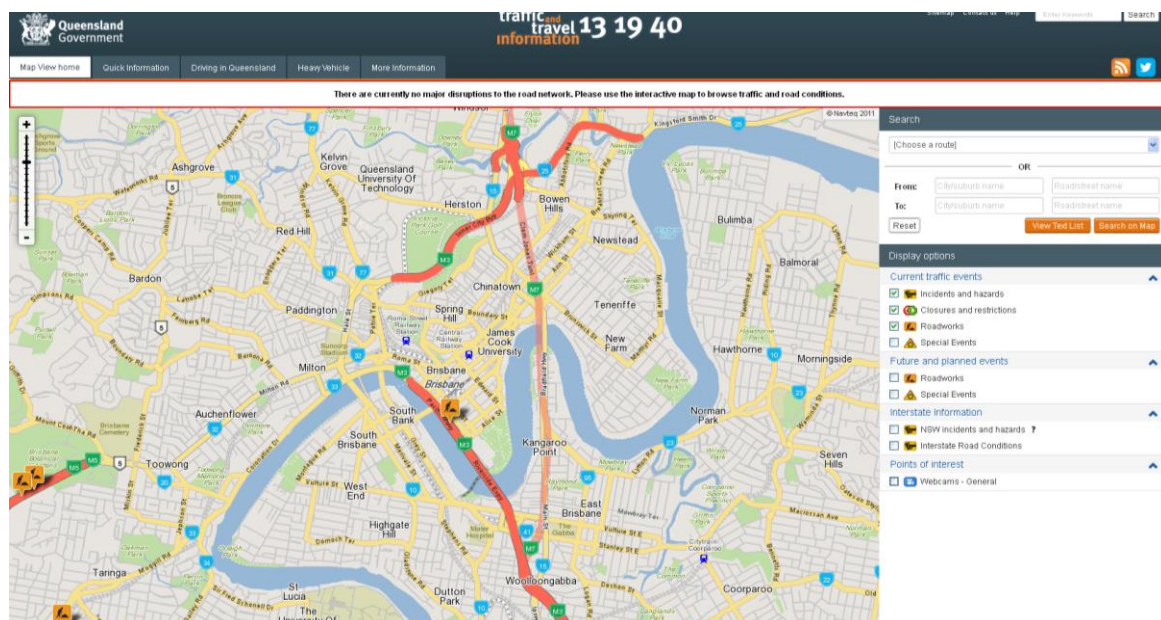


Figure 5: Example of the Queensland Traffic Information console

2.5.1. Conclusion

This website is interesting for the functionalities that involve visualization, and the structure with a column on the side where the features can be switched on or off is something that will be part of the STREETLIFE control panel. The functionalities about the “social” side of mobility (for example car pooling, or bike sharing) and the green aspects of different kinds of mobility seem to be lacking in this website. As said before, we have access only to the cockpit destined to citizens, we don't know if there's a similar instrument reserved for mobility managers, and if this second cockpit has also some more features. Anyway, the STREETLIFE control panel will have to consider also the green mobility features (emissions saved, gamification points) and the social aspects of mobility.

2.6. Econolite ITS Systems

The US brand Econolite since 2009 produced a software for Advanced Transportation Management System called CENTRACS, see [7]. This software is different from the ones listed before because there isn't an application for citizens, but it's designed only for mobility management purposes. CENTRACS is divided into modules that the city can pick and choose.

There's a module for controlling streets, a module used to see what's going on in the monitoring cameras along the city, a module for traffic data collection and solution to the problems, a module to get reports on the mobility situation, and also a module to schedule roads maintenance and planning. The module that seems more similar to a city mobility control panel is CENTRACS MMS, which can also be coupled with an optional traffic management module

This software has had great results on the management side, and has been sold in more than 100 US cities, but doesn't provide contents designed for citizens. Indeed, the citizens can see the improvements on the mobility system through time, and one of the reasons of these improvements is the CENTRACS software, but they are part of the process only indirectly, and that excludes all the social features.

2.6.1. Conclusion

The difference of the CENTRACS software from a STREETLIFE project point in respect to the other cockpits showed before is that the CENTRACS control panel has been designed for mobility managers,. Originally CENTRACS was designed to improve and help road and infrastructures' maintenance, and not as an instrument for monitoring policies, so many of the features showed on screen will be different from the STREETLIFE ones.

3. EMISSION CONTROL PANEL PILOT SCENARIOS

This section describes specific pilot scenarios for the Mobility Management and Emission Control Panel. They are derived from the deliverable D6.1 [1] and presented here in a more detailed and pilot specific way. The scenarios that are described here are going to be implemented for the MMECP. This section is structured into three sub-sections concerning the three city pilots. The requirements for MMECP have come from scenarios and use cases of BER and ROV only, but a main goal of the MMECP in Y2 is to integrate TRE as well. Thus a scenario for the usage of the MMECP in TRE needs to be evaluated.

3.1. BER scenarios

The BER Use Case “Guidance Bike & Car Sharing” is a specific elaboration of the original STREETLIFE Use Case BER-MGMT-4 described in Deliverable D6.1 [1].

Regarding a list of miscellaneous social events, the STREETLIFE system will collect information about upcoming events from available public event portals via an API. In the first step these calendar of events will be provided by VMZ for social events of “Tempelhofer Freiheit”, see [8]. The STREETLIFE System then classifies the event audience and dimension to deduct the mobility demand for visiting the social event and for travelling back home. Using historic data and statistics about bike availability at provider’s bike sharing stations (for BER availability data from CallABike and NextBike will be available), especially the usage of shared bikes as transportation mode to and from events will be examined, to define the impact on event mobility demand and on mobility resources. This will be combined with additional influencing factors like weather or expected consumption of alcoholic beverages. The availability of bicycles from bike sharing services and cars from car sharing services as well as the parking space availability and the traffic situation (road works, congestions, etc.) for a specific area will be aggregated to receive the mode of transportation. In order to calculate the coverage of the mode of transportation for traffic guidance with respect to a specific event and thus a specific type of audience, the use case will put the modal availability and the demand into relation. Furthermore the modal history regarding an area for a certain period of time can help to validate the modal split calculation. The STREETLIFE system then will produce special guidance information in case of specific events for the usage of the mobility manager at the traffic centre. For each supported event, such guidance information will be calculated automatically and presented to the mobility manager - in conjunction with the event notification itself. The STREETLIFE system will also support a recommendation system to inform service operators, especially bike sharing provider, about critical areas where their means of transport are required.

All KPIs for this scenario, which are based on aggregated and evaluated data, will be presented in a control panel on a city map or as recommendation text for guidance. These KPIs are:

- **Event Mobility Demand:** Includes the modal split of classified audience for a specific event located on the city map, to describe the preferred means of transportation related to visitor’s and environmental oriented circumstances.

- **Modal Availability:** Shows the actual and historical availability of transportation modes, especially of shared bike at bike stations within the area of the event location as well as for the whole city. In case of future events a forecast of mode availability will be related to each event type, location and environmental circumstances.
- **Resource Balance:** For recommendation of transportation coverage a traffic light icon for all relevant traffic modes will give guidance for transportation services and mobility management.
- **Distance to Modes:** The distance from the event location to the transportation mode access point will give hints on how comfortable its use will be for the classified audience.

Within the first iteration the control panel will only realize the guidance for bike sharing.

3.2. ROV scenarios

The ROV pilot distinguishes between a) strategic KPI's – which represent high-level goals that the city of Rovereto has defined for itself in its Urban Mobility Plan and which must be pursued for the improvement of the urban mobility – and b) scenario-level KPIs which are directly linked to the goals defined for the STREETLIFE project concerning the city of Rovereto.

Strategic KPIs:

- **Modal split:** The modal split describes the trip distribution by different means of transport. The idea is to show the modal split for the city of Rovereto with pie graphs depicting the shares of different ways of transportation. The control panel should also enable the mobility manager to visualize the modal split not only for the city as a whole, but also for different zones within the city, which can be selected on the dashboard. The technical problems that come along with it derive from the capability to calculate the modal shares of people walking and of people taking public means of transport.
- **Reduction of carbon emissions:** This KPI will provide an estimate of carbon emissions saved thanks to policies implemented in the scenarios

Together with the strategic KPIs, in order to have some pilot-oriented indicators, some scenario-level KPIs were identified. These ones will be already available in the first iteration phase of the pilot: bike sharing trips, bike sharing distance, bicycle flows, and park & ride usage while all the scenario-level KPI's connected to the car pooling will be studied and tested in the second iteration phase.

The scenarios that will be tested in the first iteration phase of the ROV Pilot are park & ride and bike sharing. In the first iteration phase bike sharing will not be based on a particular field test. It will be examined as one of the possible ride options within the park & ride field test. In the first iteration phase there will be three field tests: first the usage of park & ride with the help of STREETLIFE will be tested on a sample of about 50 commuters who regularly commute to ROV during the year. In a second experiment, the park & ride will not be implemented on a selected sample of people: STREETLIFE will instead be tested during the Christmas Markets in ROV and the testing field will be open. The third test will be designed to evaluate the dashboard functionality.

The functionality of the control panel that can be evaluated in the first iteration includes:

- ROV-BS/2 monitoring bike location and e-bike usage (bike sharing)
- ROV-BS/3 inspecting statistics and routes of bike usage (bike sharing)
- ROV-PR/4 the mobility manager will be able to browse and read the notifications imported from the police headquarters (park & ride)
- ROV-PR/5 the mobility manager analyses the P+R use data (park & ride)

In the first iteration phase the development of the control panel is focused on the evaluation part and so the main goal will be to test the collection, organization and visualization of data needed for the analysis of the mobility system. More complex functionality for the data analysis, the balance of the system, and the design of new or different structures will be implemented once the evaluation part is completed.

In the next months the STREETLIFE control panel will develop more complex and detailed indicators and calculations regarding these mobility aspects. This group of indicators will be essential to reach two purposes: 1) to provide a first glance of the traffic situation to the mobility manager (i.e., the person in front of the control panel); and 2) to collect and to show data and necessary information to calculate more complex indicators and implement other functionalities in the next development phases of the control panel.

3.3. TRE scenarios

TRE has not planned to use the MMECP in the first iteration phase of the project. Thus no use cases are defined yet.

4. METHODOLOGIES

This section introduces the usage of the Mobility Management and Emission Control Panel. It also shows a mockup design for the UI. It describes the methodologies that are used in the MMECP and why they are used. Thus the user of the MMECP can affect in the traffic situation, a guidance philosophy will be explained. The MMECP is responsible for providing technology for simulations. This section describes pilot specific simulation scenarios that will show how the STREETLIFE system will influence the urban mobility situation.

4.1. STREETLIFE KPI calculation and visualisation

This section gives an overview of how the MMECP can be used. The target group for the MMECP are mobility manager. Thus a main view of the MMECP is a map of an urban area the mobility manager is responsible for, see Figure 6. The MMECP system uses one installation instance for each pilot. So each MMECP system covers only the area it is installed for. The map view will display several local and project-wide KPIs; for example average availability of bike stations and parking lots as local KPIs, and current CO₂ emission as a project KPI.

All KPIs are pilot specific. The user of the MMECP is able to select or deselect the KPIs he wants to act with. The bottom area of the MMECP is foreseen to select the type of KPIs to display. The display options on the left side can be used to have a more grained filtering of the KPIs. The MMECP needs to be configured for each pilot to show the dedicated KPIs, as shown in section 3. The visualization of the KPIs depends on the related metrics used to calculate the KPIs. This is an on-going task in the project, thus a final visualization cannot be

provided in phase one. Until this is done the MMECP will be able to display local KPIs like those are presented in section 3. These KPIs are displayed as objects on the map. Each option can be selected. The secondary display options show detailed information about the selected object. If for example a bus stop is selected it will show the corresponding bus lines and their timetables. The visualization and dedicated data needed for the visualization of course needs to be configured beforehand during the installation of the MMECP. The right side of the MMECP shows the notification centre. This is used for notifications to be issued to the user, e.g. new events or occupied parking lots. The user receives information about the mobility situation in its city. If he recognised an issue he is able to react on that. Selecting an object or an area on the map he can give guidance to the selected object or area to a specific addressee. This will be explained in more detail in section 4.3.

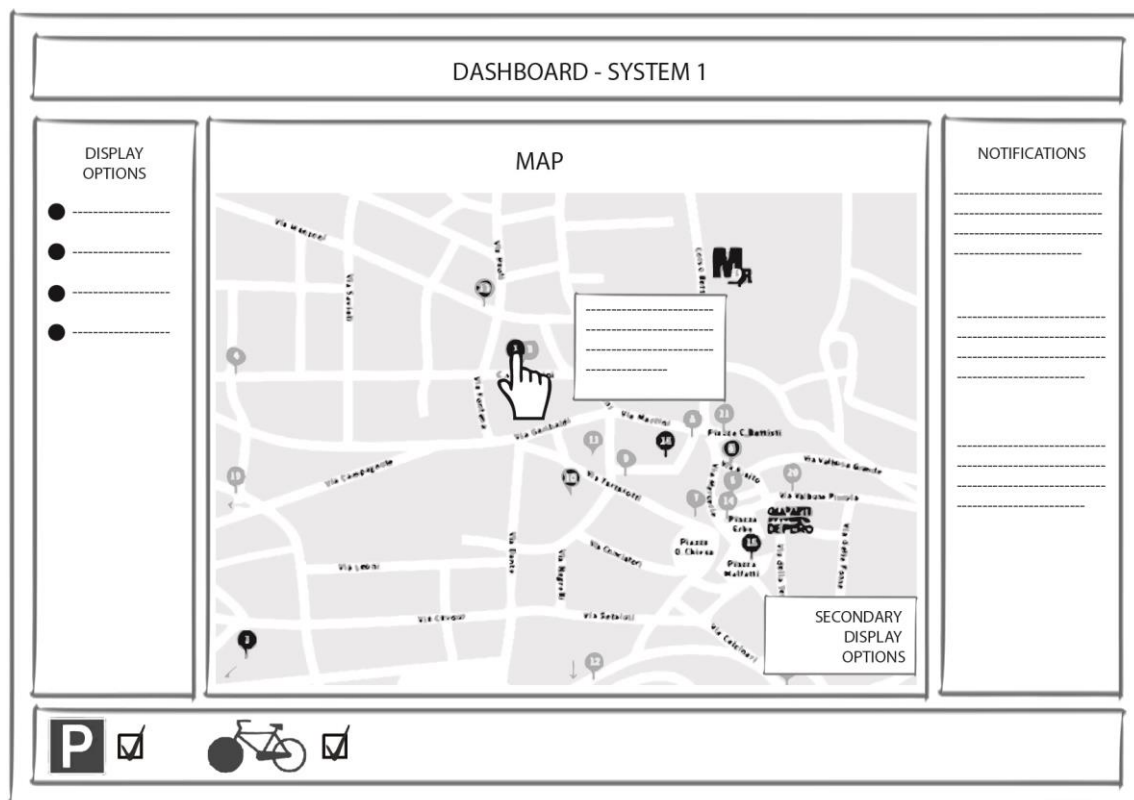


Figure 6: MMECP GUI draft

Another view on the MMECP is a statistical view. The statistical view uses historical data to show the KPIs of the urban area as dedicated charts. The user has to select an area on the map and switch to the statistical view. This will show charts of the KPIs that are configured for the pilot. An adequate default value for showing statistics about the last n days needs to be considered. The user will be able to select a certain time period for the statistics to look at. He is also able to apply certain functions and constraints to the charts. This is described in section 5.4.

4.2. Specific emission control panel KPIs

In the following, the derivation of specific emission control panel KPIs is described, considering concerns on data and their evaluation and analysis procedure. The KPIs are related to the emission control panel pilot scenarios in section 3.

For the “Guidance Bike & Car Sharing” use case, the STREETLIFE Control Panel will display KPIs and guidance information for the mobility manager as listed in section 3 on a city map or as recommendation text. There will be information visualized in the event area for all supported events as well as on city level according to the pilot scenario description above. In order to give guidance for specific events, different indicators have to be evaluated. In the following the derivation of these indicators is documented, as there are Event Mobility Demand, Modal Availability, Resource Balance and Distance to Modes. Within the first iteration of the pilot, the control panel will only realize guidance for bike sharing.

Concerning the mobility system analysis and the implementation of policies, the STREETLIFE Control Panel in the first iteration will show KPIs about the performance of the bike sharing system and the Park & Ride system. In order to have a better understanding of the dynamics involving the systems, the control panel will show data regarding the levels of utilization and occupancy rates not only in real time, but also from the previous weeks or months. In the following we describe the derivation of indicators that involve the system analysis. It will use data coming from different sources to reach the goal of having a clear picture on the performance of the system.

4.2.1. Modal split forecasting

4.2.1.1. Introduction and motivation

Giving good guidance depends on detailed and accurate information. The ability to forecast the modal split regarding to specific events is powerful information for city manager and event organizer. An event in this case means that something happens at a certain place or area to a certain time, e.g. cultural events, construction sites, demos and so on. For the first phase of the project and in this deliverable we will focus on forecasting the modal split for cultural events. This is a derived use case for BER from the deliverable D6.1 [1]. Since this process is based on generic mechanisms it is possible to apply this process to other forecasting activities beside the modal split. In the following the term “Event” means a cultural event. Because of the generic approach that is presented in this section, the forecasting mechanism is not only bound to this kind of events. In an abstract way we can think of events in case of something is happening to somewhere at a certain time.

4.2.1.2. Data

To forecast the modal split for events a lot of detailed data is needed. Not all needed data is provided to the STREETLIFE project at the current state of the project. The data for this process is currently based on surveys, catalogues and assumptions. Any fuzzy data will be aligned and corrected by surveys in the second phase of the project. For the modal split forecast we distinguish between three types of necessary data; *Type of Persons* (ToP), *Type of Event* (ToE) and the *Modal Choice* (MC).

ToP describes a cluster of people that are visiting the event to examine. The attributes of the ToP are derived from a survey (SrV 2008) that was undertaken in Berlin in 2008 by TU Dresden in cooperation with the Senate Department for Urban Development and the Environment of Berlin, see [9]. The consortium received the allowance to access and use the raw data from this survey in the STREETLIFE project. A new version, which was undertaken in 2013 will be published at the end of 2015. The consortium is looking forward to include

that new data later, when available. The SrV 2008 is also used to derive rules that are able to determine the modal choice for the ToP.

To forecast the modal split for events, the system needs to know which ToP usually visits which ToE. An assumption in this consortium is, that the modal choice depends on the individual. Thus a list of personal attributes needs to be considered. The following is a list of personal attributes that are derived from the SrV 2008.

- Age
- Gender
- Profession
- Family status
- Interests
- Financial income

In Y1 the forecasting functionality will be partly based on assumptions by the STREETLIFE consortium. Thus this is not yet a complete list of necessary attributes. In Y2 these data will be cross checked and validated with the help of a survey. To cluster the ToE we use attributes of cultural events like the following:

- Type (e.g. concert, sports)
- Genre (e.g. rock, soccer)
- Artist
- Teams/Participants
- Price

Historical data of event locations are useful to create an estimate of the amount of persons attending to the ToE. To determine the modal choice we assume other attributes concerning the event like:

- Venue
- Day of Week
- Month (or Season)
- Time
- Weather
- Distance to visitor

The following describes how the MMECP can predict the modal choice for ToP visiting a ToE. The MMECP needs to predict which ToP is visiting which ToE, this will be done in the second iteration. For the first iteration we use assumptions to work with.

4.2.1.3. Evaluation and analysis

The process to forecast the modal split that is used in the MMECP is shown in Figure 7. Before the explanation of the diagram and its components some fundamentals are described. The whole process is divided in two parts:

- Generate Knowledge Base (see also section 4.2.1.3.1) – In this first sub process rules will be generated, which enable the possibility to forecast the modal split of an event. The sub process has to be executed only if the knowledge base should be regenerated for new evaluated data out of surveys or others.
- Forecast (see also section 4.2.1.3.2) – In this second sub process the generated knowledge base is used to predict the modal split for an event based on the argumentation system Carneades. So it has to be executed for each new upcoming event and the result should be stored in the metadata of the event.

The argumentation system (AS) Carneades used in STREETLIFE constructs arguments to verify or refuse hypotheses using a rule base of argumentation schemes. It is developed by Thomas F. Gordon et Al. at Fraunhofer FOKUS and is used in European IMPACT project, see [10]. To forecast the modal split at STREETLIFE for possible means of transportation a hypothesis has to be validated like:

*To go to the event the Audience will take the **car** if the event takes place at THF starts 6 o'clock p.m. on Thursday (and other specific properties).*

To validate such a hypothesis, generated rules by the knowledge base will be used as argumentation scheme. As it is written in the beginning of this section, the prediction is partly built on assumptions, which will be corrected by evaluation and surveys in the second iteration. So the argumentation schemes have to be corrected in the second iteration as well. Actually the structure of the schemes used for the first iteration is exemplary. Carneades constructs an argumentation graph (Figure 10) based on the rules in the scheme, which verifies or refuses the hypotheses. Additionally the graph can be visualized, which may be helpful as basis for giving good guidance. Carneades was implemented in Clojure, a Lisp dialect that targets the Java Virtual Machine (JVM). Thus the necessary argumentation schemes are written in Clojure using so-called S-Expressions (Symbolic Expressions). In the Lisp family these expressions are used to represent source code but also data. With those expressions premises and conclusions are built for rules of an argumentation scheme like the following minimal example will demonstrate:

```
(make-scheme
  :id 'take_the_car
  :conclusion '(go_to_with ?E car)
  :premises [(pm '(located_at ?E o2))
             (pm '(at day ?E thursday))
             (pm '(starts_at ?E 1800))])
```

This little schema represents the hypothesis at the beginning and contains the conclusion that the audience of the event will take the car and three premises regarding the properties of the event. S-Expression of both conclusion and premises starts with the predicate followed by the subject and finally the object or rather the value. All generated rules out of the knowledge base have to be available in this structure to validate hypotheses with Carneades.

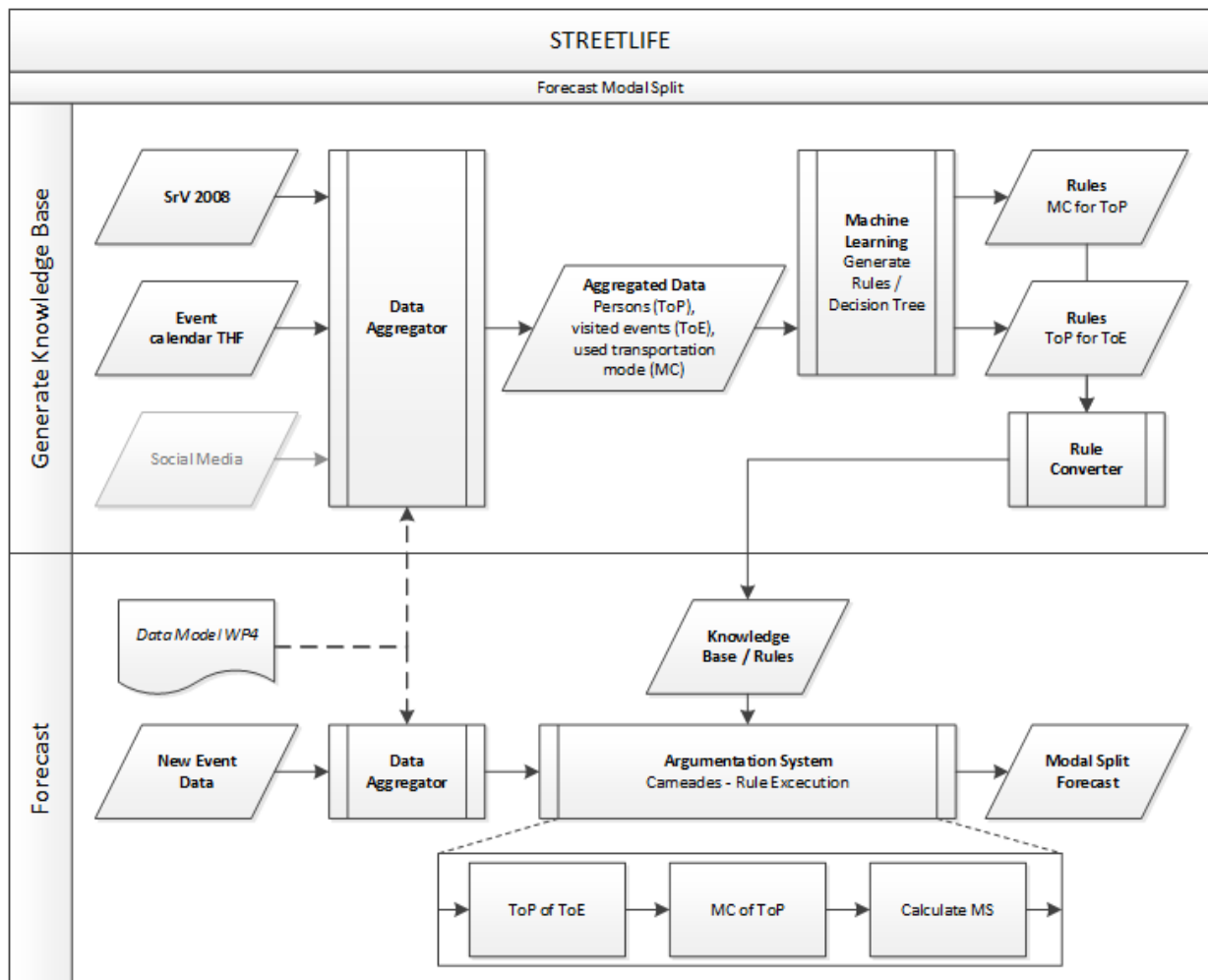


Figure 7: Forecasting process

The following is an explanation of this diagram and its components and therefore the single steps of the forecast process.

4.2.1.3.1. Generate Knowledge Base

The sub process starts by filling in the data mentioned in section 4.2.1.2. To create a rule base for the forecast with Carneades a machine learning tool shall create a decision tree regarding this data. These tree-like graphs are used in decision analysis to reach a specific goal and are built out of nodes in a hierarchical structure (see Figure 7 and 5). There exist internal nodes, which represent decisions regarding a single feature in the data like location of the event and nodes or rather leaves at the end of a tree with the conclusion. Here the target feature for the leaves is the means of transportation used by the audience to go to an event.

In STREETLIFE usable tools to generate such decision trees have to be open source at least. Here two different data analytic suites were evaluated which represents the two big groups.

- RapidMiner (graphical application) – older versions published under AGPL
- R (command line based application) – published under GPL

In general both support the same functions. They read the data sets from local files (*.csv, *.xls ...) or database and includes different algorithms for generation of decision trees with configurable parameters. Also it is possible to export decision trees into different types of files. For the reason that the generation of trees has to be executable by STREETLIFE backend it's necessary that the data analytic tool is accessible by other applications. For that RapidMiner supports a batch mode to run prepared processes. However, in the first iteration R will be used because the generation of the decision trees by STREETLIFE backend is unproblematic since it is command line based. For the second iteration it is conceivable to use other data analytic tools and evaluate the different decision trees regarding quality, complexity or performance of generation process.

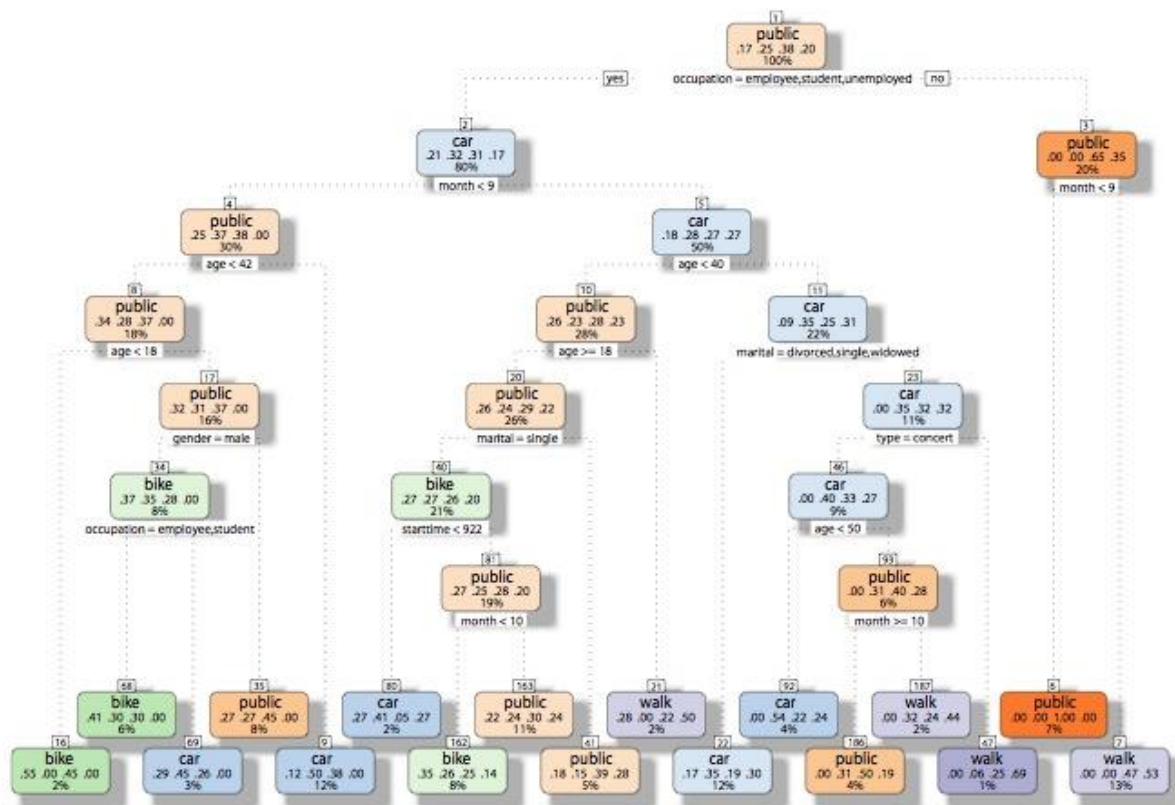


Figure 8: Decision tree generated by R

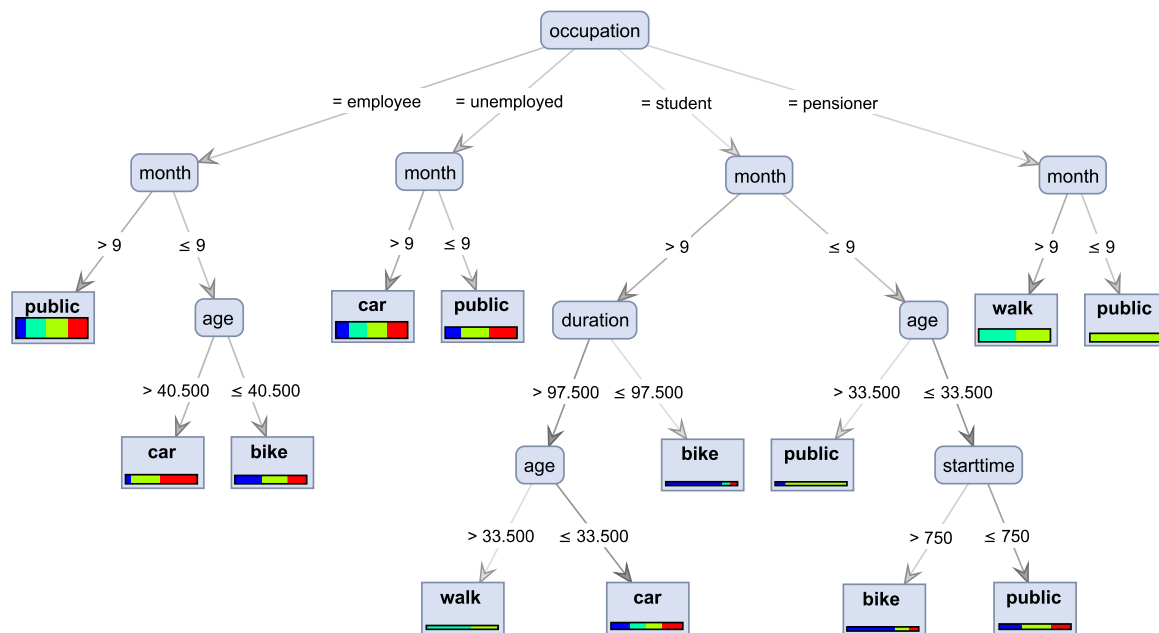


Figure 9: Decision tree generated by RapidMinder

Finally the decision trees to derive the Type of Person (ToP) for the Type of Event (ToE) and to predict the MC for ToP have to be transformed in the already shown schema structure of Carneades. Therefore the RuleConverter component was implemented which imports a decision tree and returns a set of Carneades schemes as text. To support more than trees of R a standardized exchange format of decision trees should be used. Such a standard to exchange statistical models is the Predictive Model Markup Language (PMML). The PMML models are described by own XML elements which are declared in the PMML specification, see [11]. There are several elements to represent all the information from a decision tree like his internal nodes or leaves. Since the structure of used argumentation schemes is not final the conversion has to be fully customizable. For that the way of transformation of the PMML model is specified in a separate XML schema. Table 3 shows the elements that should be used:

Table 3 PMML transformation elements

ELEMENT	DESCRIPTION
mapText	It is included in all other elements (except the head) and contains the transformation text for the corresponding elements.
head	With this element necessary metadata can be defined as text, which will be placed at the beginning of the conversion result.
body	The mapText of this element represents a complete PMML model and encloses his defined rules.
nodeWrapper	With his mapText a single rule of the decision tree is represented. It contains the internal nodes with the decisions and the leaves.
node	The specified mapText in this element represents a single internal node.
predicate	The decisions in the internal nodes were represented by predicates in PMML model. So the mapText of this element specifies the conversion of a decision step.
score	In PMML models the results in the leaves of the decision trees are represented in a score distribution and therefore this element contains the necessary mapText.

For the conversion of a decision tree into a Carneades scheme these elements listed above are used to specify the concrete transformation. After this last step of the sub process a rule base of generated decision trees regarding the aggregated data is available to forecast the modal split by Carneades.

4.2.1.3.2. Forecast

As soon as a new upcoming event is detected the necessary data has to be harvested and aggregated. Regarding those specified properties the argumentation graph has to be constructed based on the generated argumentation scheme of the first sub process. In the first STREETLIFE iteration this process will be executed by a Clojure script, which is

automatically started in the STREETLIFE backend. In the second iteration the RESTful web service API from the Carneades web server should be used on demand to construct the corresponding argumentation graphs. As described in the beginning of section 4.2.1.3 predicting probability used means of transportation a hypothesis has to be validated by Carneades. The corresponding argumentation graph can look like illustrated in Figure 10.

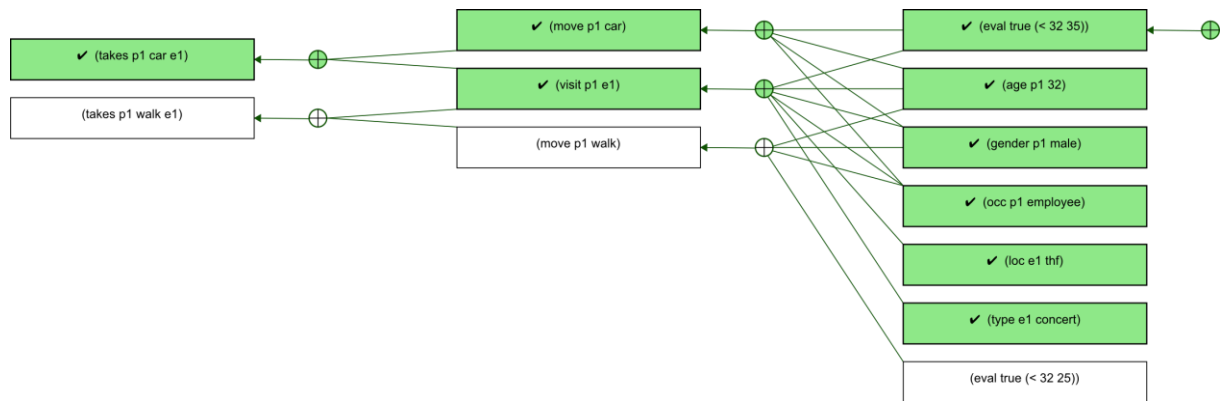


Figure 10: Argumentation graph constructed by Carneades

Based on such an argumentation graph the means of transportation for a specific event can be derived. In the current implementation of the argumentation scheme for the first iteration there are two premises which have to be fulfilled. The first one represents the derivation of ToP for ToE and actually is called *visit*. The second one is called *move* in the current implementation and predicts the MC for ToP. Both premises are bound through additional premises regarding properties of the event and corresponding audience already mentioned in section 4.2.1.2.

To calculate the overall modal split for the given event the probability from means of transportation for ToP is still missing in the argumentation graph but it is already included in decision trees. In the second STREETLIFE iteration it will be implemented in the Carneades scheme, so for example an equation similar to the following for the modal split will be produced:

$$(60\% \text{ ToA}_1 * 40\% \text{ car}) + (40\% \text{ ToA}_2 * 75\% \text{ car}) = 54\% \text{ car}$$

$$(60\% \text{ ToA}_1 * 60\% \text{ bike}) + (40\% \text{ ToA}_2 * 25\% \text{ bike}) = 46\% \text{ bike}$$

Finally with this calculation based on argumentation graphs by Carneades the modal split can be predicted for a new upcoming event.

4.2.2. Modal availability

4.2.2.1. Introduction and Motivation

The indicator for modal availability shows the actual and historical availability of transportation modes, for example of shared bikes at bike stations within the area of an event location as well as for the whole city. In case of future events a forecast of mode availability will be related to each event type, location and environmental circumstances.

The modal availability e.g. of bikes from bike sharing services for a specific area is an indicator for the availability of transport options and respective capacities to be visualised with the STREETLIFE control panel.

In summary, the control panel visualizes for the mobility manager the following information on modal availability for bike sharing in the first pilot iteration:

Bike sharing for the whole city area:

- Bike station ID
- Provider ID
- Station address
- Total number of slots per station
- Available bikes per station

For bike sharing in a specific event area the following indicator is calculated and visualized in addition to the information above:

- Availability of bikes in event area

4.2.2.2. Data

The data that is necessary in order to calculate the modal availability for bike sharing is described in Table 4. It is collected by bike sharing service providers and made available in the STREETLIFE system. For each bike sharing station real time information on the current amount of available bikes can additionally be retrieved.

Table 4: Bike sharing data

BIKE SHARING	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
BIKE SHARING STATION	Service provider	<ul style="list-style-type: none"> • Station ID • Provider ID • Station address • Total number of bike slots 		Detailed figures of the bike sharing service used in the system analysis process.
BIKE AVAILABILITY	Service provider	<ul style="list-style-type: none"> • Number of available bikes • Availability time 	<ul style="list-style-type: none"> • Number of available bikes in event area at event time • Additional results from analytics (average availability, availability profiles, event related figures) 	Figures on the number of available bikes for each bike sharing station, related to a particular time of the day as well as additional output from analytics

Additionally, information concerning specific events is required for calculating the modal availability in an event area. This data (Table 5) is collected through information given on public event portals and is made available in the STREETLIFE system.

Table 5: Event data

EVENT REGISTRATION	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
EVENT	Event portal	<ul style="list-style-type: none"> • Location • Start date • End date • Type of event • Number of visitors 	<ul style="list-style-type: none"> • Output data used for availability correlation 	Different event data are necessary for availability correlation for specific events

For advanced data analytics and availability forecast modelling also weather data have to be considered, see Table 6.

Table 6: Weather data

WEATHER DATA SOURCE		INPUT DATA	OUTPUT DATA	DESCRIPTION
WEATHER	Weather service	<ul style="list-style-type: none"> • Location of weather station • Temperature • Condition • Time 	<ul style="list-style-type: none"> • Output data used for availability correlation 	Data on weather like temperature, location of weather station etc. for analysing its influence on mode availability in forecast modelling. Weather condition consists of wind, pressure, cloud coverage/rain.

4.2.2.3. Evaluation and Analysis

In order to forecast the availability of mode options in an event area it is necessary to calculate the modal availability indicator. Furthermore it is necessary to perform historical and forecast modelling to receive the normal and event based modal availability and get informed on the difference. In the following a method is presented which calculates modal availability for bike sharing in a specific event area.

The bike sharing availability BS in an event area e , e.g. within a circle around the event location with a defined radius, e.g. 300m, at time t is calculated as the sum of all available bikes in all bike sharing stations that are placed within the event area e at time t .

$$BS(e, t) = \sum_{\substack{\text{list of all bike stations} \\ \text{within area}}} \text{availableBikes}(t) ;$$

To generate historical data and to predict modelling, data collection and evaluation described should be conducted continuously over the whole city area. With that assumption, we can then draw from several methods for advanced data analytics to forecast modal availability. Demand and area wide data collection can be analysed continuously for pattern generation and forecast.

We can use a daily rolling window for historical data analysing, as visualized in Figure . It is also possible to specifically picking out e.g. weather conditions as shown in Figure 12. Thus it shall be possible to define an average daily profile of modal availability at a specific bike or car sharing station (or others) fitting to seasonal conditions.

Average daily availability profile derived using a daily rolling window (Figure 11):

As shown in the figure below a section of different daily profiles of a certain number of running days from e.g. day $i-2$ up to day i is taken to calculate an average profile out of these single profiles. One day later the window section rolls on and the average availability profile is calculated using the single daily availability profiles from day $i-1$ up to day $i+1$.

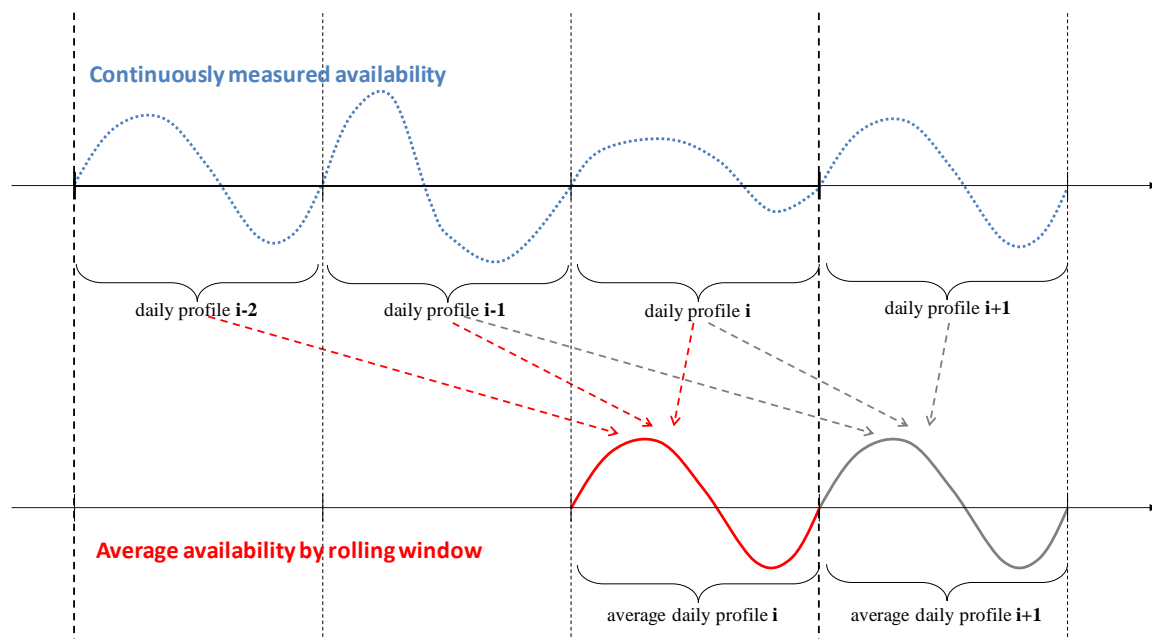


Figure 11: Average daily profile pattern by rolling window

The picture shows an average daily availability profile of modes to be chosen. It is derived by specifically picking out single profiles e.g. according to the actual weather condition:

Another option to derive average availability profiles is picking different daily profiles out of the continuously measured availability according to a special similarity condition. In the example presented below, weather is selected as similarity condition, since for bike sharing the weather situation is a crucial condition for the decision to use a bicycle or not. Thus weather conditions are also decisive for higher or lower bike availability at a bike sharing station. With knowledge of the weather conditions that prevail on different days, the sum of days with the same weather can be retrieved from historical data. Using the availability profile of these specific days an average availability profile can be calculated for different weather conditions and availability profile patterns for weather types can be applied for availability forecasting.

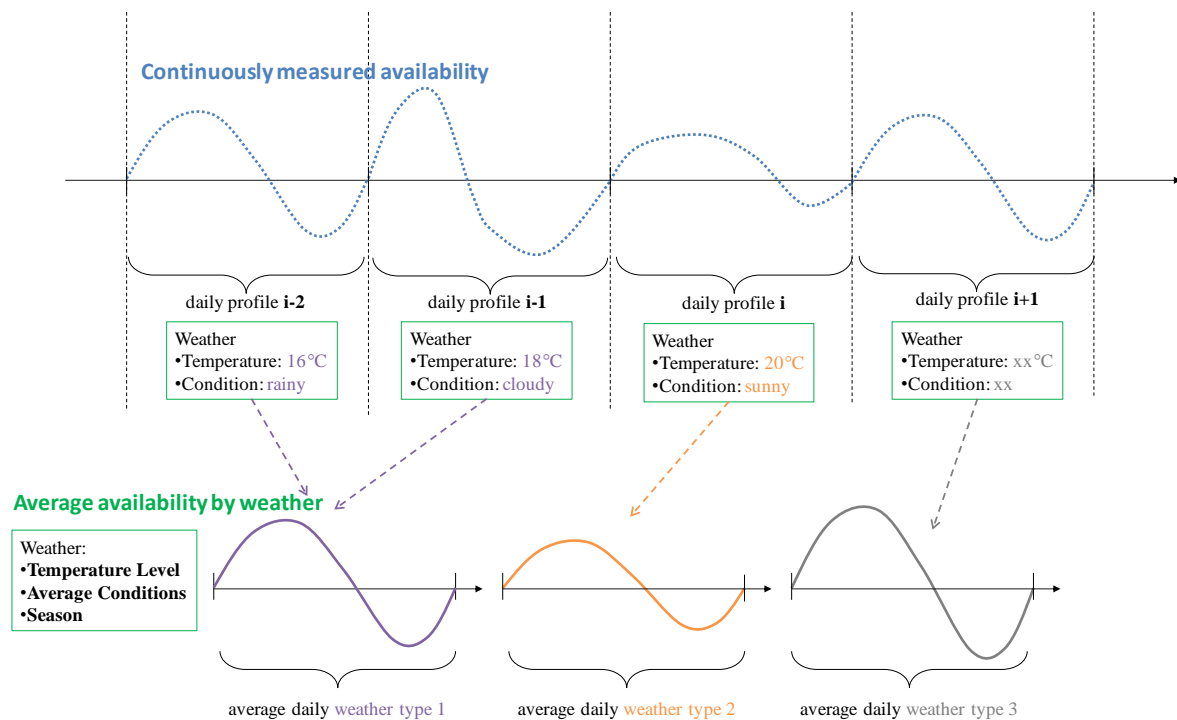


Figure 12: Daily profile pattern by weather condition

In case of events there will be a deviation between the average availability profile for e.g. a bike sharing station in the event area and the actual profile for the event time. The difference between these profiles results in a “delta” of deviation. Illustrating this on the basis of bike sharing, it means that there is a higher availability of bikes in a bike sharing station in the event area as people may come to the event by bike. On the other hand, the number of available bikes decreases in the event area at the end of an event when parts of the audience leave by shared bike (see

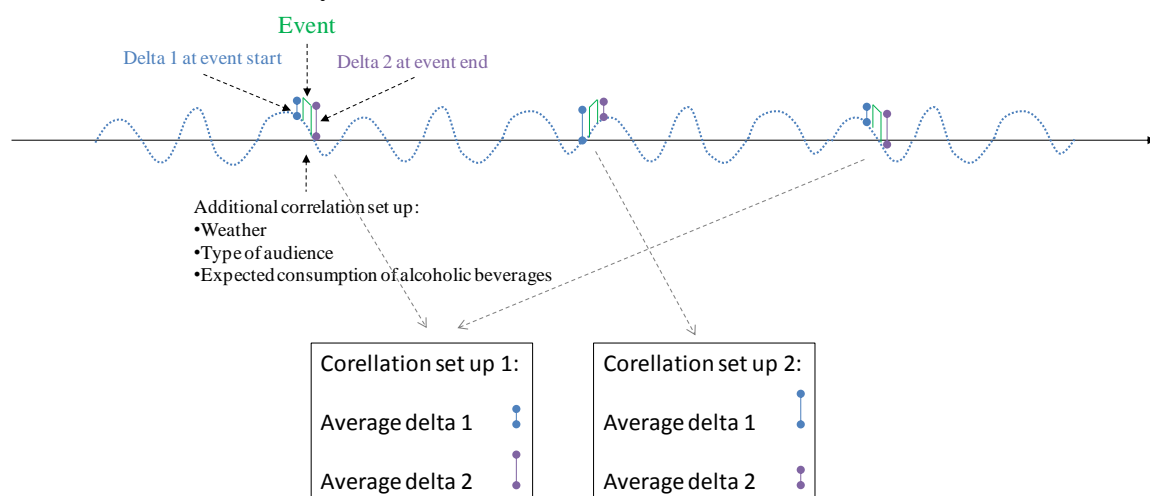


Figure 13). The calculated change represented as delta before and after an event has to be filtered and stored as well as average delta.

The deltas have to be correlated with additional information about weather, audience type and expected alcohol consume until event end time in relation to event start time. For different weather conditions or audience types the deltas differentiate in turn and lead to correlation set ups with average deltas for specific compositions of event characteristics.

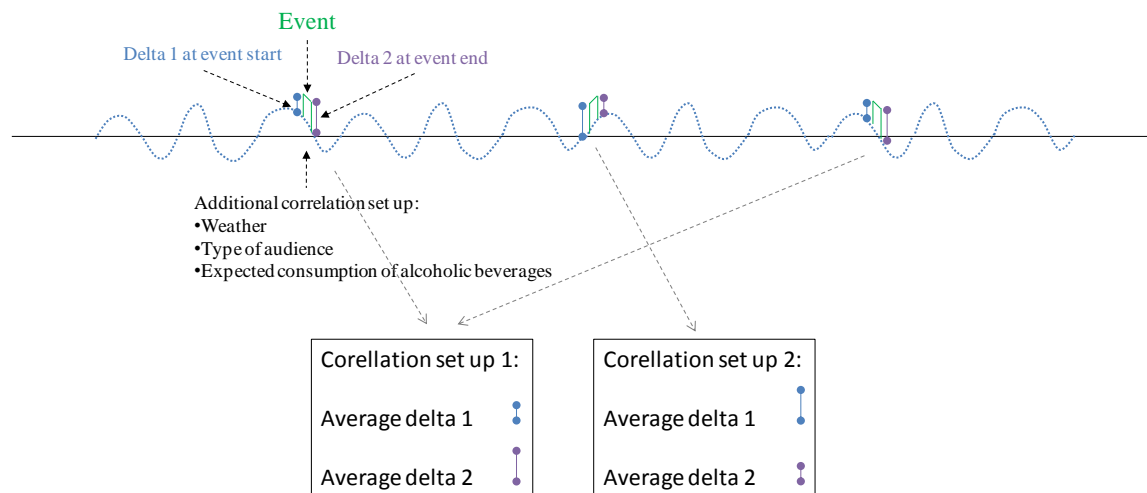


Figure 13: Change in availability profiles caused by event

4.2.3. Resource Balance

4.2.3.1. Introduction and Motivation

Resource balance means the balance between transportation demand and transportation supply. In order to evaluate the resource balance for different traffic modes, especially in an event area, appropriate mode availability and visitors demand have to be compared. It gives some indication if the demand can be satisfied or if there could be supply gaps for various traffic modes in the event area.

For visualizing the resource balance for different modes an indicator visualised as traffic light is displayed in the control panel. For all relevant traffic modes this indicator will give guidance for transportation services and mobility management. Green colour in the traffic light visualization advices that the demand can be satisfied for a certain traffic mode whereas red means that there is a supply gap.

4.2.3.2. Data

The data needed for the resource balance evaluation results from the calculation described above. In the following the data is listed that is necessary for resource balance concerning bike sharing which is in the focus of the first pilot iteration.

RESOURCE BALANCE	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
EVENT MODAL SPLIT	Calculation in section 4.2.1	<ul style="list-style-type: none"> • Event modal split • Modal split history 	Resource balance	Input data for event modal split result from evaluation methods presented in 4.2.1. Based on this input the resource balance indication is calculated.
MODAL AVAILABILITY	Calculation in section 4.2.2	Number of available bikes in event area	Resource balance	Input data for modal availability result from evaluation methods presented in 4.2.2. Based on this input the resource balance indication is calculated.

Table 7: Data overview for resource balance evaluation

4.2.3.3. Evaluation and Analysis

To calculate the traffic mode resource balance for the guidance needed in case of a specific event and type of audience, especially the modal availability and demand has to be identified. Additionally the modal split history regarding an area for a period of time can help to validate the calculation. The indicator can be calculated for different traffic modes in an event area e at time t as follows:

$$\text{ResourceBalance}(e,t)[] = \text{modalAvailability}(e,t)[] - \text{eventModalDemand}(t)[][] * \text{weightFactor}[];$$

The “modalAvailability” in the formula above was already calculated as an indicator in section 4.2.2. A way to derive the “eventModalDemand” is described in the section on “event modal split” (4.2.1). The “weightFactor” gives potential configuration to regulate the actual probability of modal use depending on further circumstances (like e.g. weather).

Calculation example for bike sharing:

The “modalAvailability” in the general method above corresponds in this example with the calculated bike sharing availability BS from section 4.2.2.3. The “weightFactor” is considered as 1 which means that there are e.g. no bad weather conditions that would decrease the bike demand at the event. The “ResourceBalance” is calculated with the following data as visualized in Figure 14.

- event start time: 19:30
- event end time: 22:00
- BS demand at event end time: 50
- BS increase at event start time: ~30

The difference between bike availability (blue line) and demand (red line) leads to a supply gap starting at event end time.

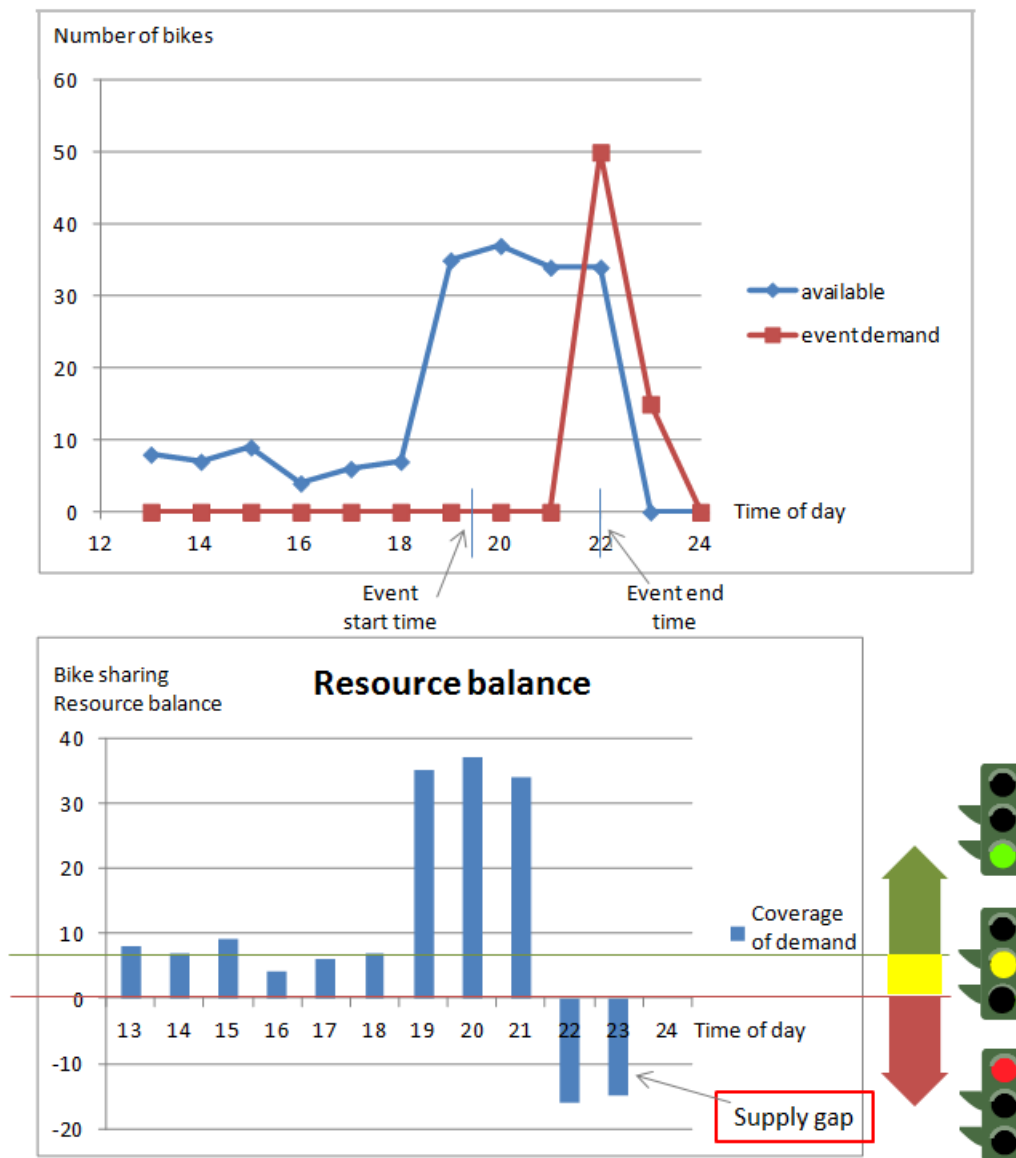


Figure 14: Example for bike sharing resource balance calculation

For history and forecast modelling the methods for advanced data analytics already presented in 4.2.2.3 can be carried forward to predict the resource balance for an event in a long term or short term forecast.

A first forecast will be done at the time of event recognition before the event starts. We use the above-described average daily pattern displayed in blue in Figure 15 and add the event caused average change (delta 1 displayed in red) in availability coming from the expected correlation setup. Illustrating this effect for long term forecast again exploiting the bike sharing example, the modal availability for bike sharing increases at the start time of the event by an average delta 1 compared to an average availability profile as people also come to an event by shared bikes. Applying the average daily availability profile until the event end time leads to an expected modal availability at that point of time (see contact point between green and light blue dashed line in Figure 15). As there is also an additionally expected demand at end time of the event that is calculated in terms of delta 2 from historical data, this has to be

applied as well in order to get a forecast for the actual demand at event end time. If the availability then results in a negative value (see grey dashed line) there is a forecasted supply gap.

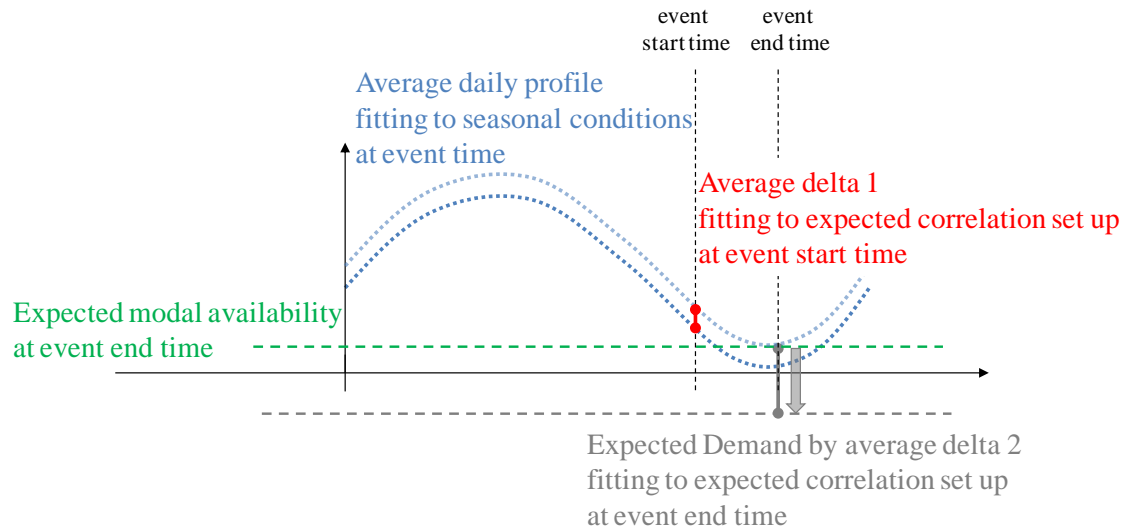


Figure 15: Long term forecast for expected modal availability

A final short term forecast for resource balance will be done with the identification of the actual availability delta (see measured delta 1 in Figure 16) at event start time. Therefore, in addition to the average daily availability profile from historical data the actually measured availability is considered. Using the average delta 2 at event end time from historical data and correlated with measured delta 1 at event start time and with additionally incorporating weather forecast, the audience type and the expected alcohol consume, the modal availability and demand at event end time can be predicted.

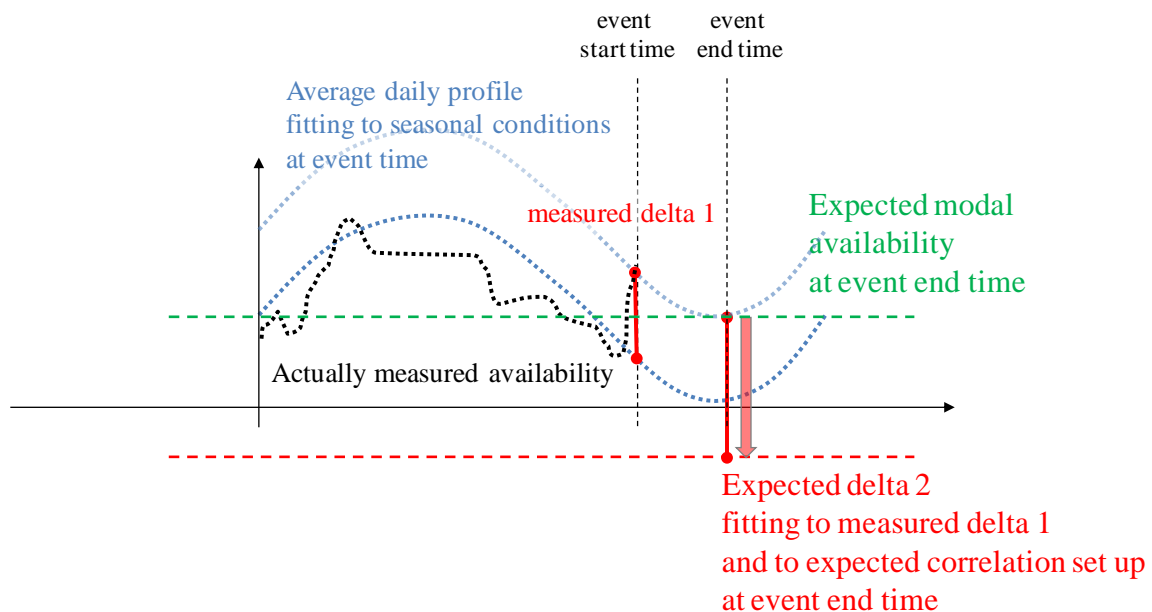


Figure 16: Short term forecast for expected modal availability

The difference between the modal availability and demand gives an answer to resource balance, in case there is a gap between the availability and the demand which leads to a supply gap for a certain traffic mode. This way the guidance for transportation services and the mobility management can be derived.

4.2.4. Distance to Modes

4.2.4.1. Introduction and Motivation

The distance from the event location to the transportation mode access point gives evidence on how comfortable the usage of the transportation mode will be for the classified audience. In the control panel the distances from the event location to the different modes will be visualized:

- Distance to bike sharing station
- Distance to car sharing station
- Distance to car from a car sharing operator that is not attached to a certain station but is free to be parked
- Distance to public transport station
- Distance to parking area

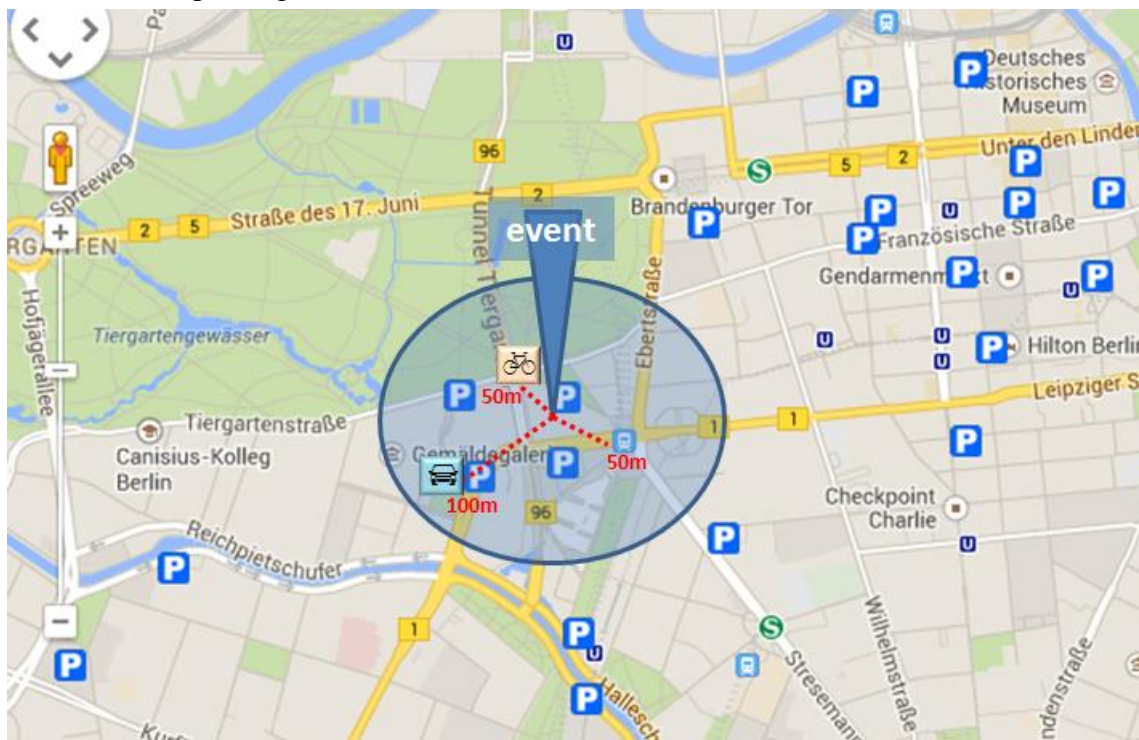


Figure 17: Visualization example for “Mode distance calculation”

4.2.4.2. Data

For calculating the mode distance to a specific event location, the location coordinates of the bike sharing stations and the car sharing stations as well as free parked car sharing vehicles, public transport stations and parking areas is needed:

MODE	DISTANCE	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
EVENT		Event portal	Location coordinates		Location data for events serve as input data for mode distance evaluation
BIKE SHARING STATION		Service provider	Location coordinates	Distance to event location	Distance between bike sharing stations and event location is evaluated
CAR SHARING STATION		Service provider	Location coordinates	Distance to event location	Distance between car sharing stations and event location is evaluated
FREE CAR		Service provider	Location coordinates	Distance to event location	Distance between free cars and event location is evaluated
PUBLIC TRANSPORT STATION		Transportation provider	Location coordinates	Distance to event location	Distance between public transport stations and event location is evaluated
PARKING AREA		Service provider	Location coordinates	Distance to event location	Distance between parking areas and event location is evaluated

Table 8: Data for Mode distance calculation

4.2.4.3. Evaluation and Analysis

The absolute distance to the event location can be evaluated for each mode by comparison of the mode location and event location. Therefore, the location coordinates can be used to calculate the distance $dist(eLoc, mLoc)$ of the event location (eLoc) to the location of mode (mLoc) via the geo-referenced functionality.

4.2.5. Park & Ride

4.2.5.1. Introduction and Motivation

The Park & Ride functionality of the control panel will enable the mobility manager to have a better understanding of the current state of the parking system and to develop policies to improve the efficiency of the system itself. The dashboard features a Park & Ride overview that enables the mobility manager to verify the functionality and use level of an outer or interchange parking, in order to relieve traffic pressure from the city centre.

In addition to the specific features concerning the data analysis on the occupancy level of those parking slots, the dashboard will display other features that enable the mobility manager to obtain a view on the state of the parking choices and to characterise the demand.

The control panel will enable the mobility manager to evaluate the tested policies for Park & Ride, assessing the amount of CO₂ saved by the planned trips with Park & Ride solutions through the app users. In particular for those parking slots pinpointed in the first iteration of the pilot in Rovereto for the Park & Ride scenario (but will be applied to every existing parking slot in Rovereto), besides parameters regarding the occupancy level of the parking slots, the dashboard will display the following features:

- number of users that plan trips including Park & Ride solutions through the app
- estimate of kilometres saved through planned Park & Ride solutions
- estimate of CO₂ saved through planned Park & Ride solutions

In order to assess the efficiency of Park & Ride system in relation to the whole urban parking system, information about the “balance” of the parking system will be displayed in the control panel thanks to significant parameters, listed below, regarding the whole urban area or only some parts.

- number of parking slots divided by type
- number of free parking slots
- number of occupied parking slots
- average occupancy coefficient
- income gained from parking slot with fee

Through the progressive enrichment of a database including that information, it is possible to estimate the occupancy rate of parking slots, on the basis of previously collected data. The hypothesis that supports this idea is that the occupancy rate of parking slots periodically changes in relation to the period of time and or particular event occurrence, as other phenomena concerning the transport sector (e.g. vehicular transit flux entity).

4.2.5.2. Data

In order to develop these evaluation instruments, an essential requirement will be to collect data on parking supply and occupancy condition of urban parking areas. This data is needed for a complete overview and description of the parking system. The result will provide the mobility manager with the necessary information for its evaluation and following management process.

The parking system is organized in:

- garages
- parking lots (free and with fee)
- parking areas with fee on the streets
- free parking slots on the streets

The system is fully described through specific features for each single item (for example garages, parking areas) of the parking system.

Table 9: Data for Park & Ride

PARK CATEGORY	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
GARAGES	Service manager access system remote control.	<ul style="list-style-type: none"> Parking supply Entries and exits in and from the parking slots (parking slot ID, date and hour of operation, user category, zone, entry or exit) Income (parking slot ID, income for a period of time, zone) 	<p>Average occupancy rate (related to parking slot ID, period of time, user category)</p> <p>Average income (related to a specific period of time)</p>	<p>Average occupancy rate value related to a single parking area for a specific hour, for a specific time band, for a specific day of the week.</p> <p>Optional: for a specific user category.</p> <p>Optional: for a specific zone.</p> <p>Average income value income related to a period of time for a specific hour of day, for a specific time band, for a specific day of the week.</p> <p>Optional:</p>

				for a specific zone
PARKING AREAS WITH FEE ON THE STREETS	Survey held by traffic aides with specific monitoring instruments. Income from parking meters.	<ul style="list-style-type: none"> • Parking supply • Hour fee • Number of free parking slots (area ID, date and hour of survey, zone) • Income (area ID, income for a reference period of time) 	<p>Average occupancy rate (for area ID, period of time).</p> <p>Average parking area income (for a specific period of time)</p>	<p>Average occupancy rate value related to a single parking area for: a specific hour, a specific time band, for a specific day of the week).</p> <p>Optional: for a particular zone.</p> <p>Average income value for a period of time: for a specific hour of day, for a specific time band, for a particular day of the week.</p> <p>Optional: For a particular zone.</p>
PARKING AREAS NOT WITH FEE	Survey held by traffic aides with specific monitoring instruments.	<ul style="list-style-type: none"> • Parking supply • Number of free parking slots (area ID, date and hour of survey, zone) 	<p>Average occupancy rate (for a specific area ID, for a specific period of time).</p>	<p>Average occupancy rate value related to a single area for: a specific hour of day, for a specific time band, for a specific day of the week).</p> <p>Optional: for a specific zone.</p>

Inside car parking lots, the data monitored is: the number of parking slots, the number of free parking slots, the occupancy rate, the hour fee and the access number. This information will be acquired directly accessing the control system of the parking lots that provides the number (for each hour) of entries and exits. With this data the dashboard is able to produce statistics about variable features, like the occupancy rate specifically related to a particular period of the day or of a specific event.

In the same manner, the data will be collected in fee-paying parking spaces in a district. In this case, the data collection process will be engaged by traffic aides who are equipped with survey instruments during their periodical inspections. Although this information is not updated constantly, the progressive database enrichment allows to pinpoint recurring features. This is done with respect to occupancy rates in order to produce a reliable estimation of occupancy rate of specific parking slots. In this case, the considered data is: number of parking slots, number of free parking slots, occupancy rate, hour fee according to the different fee zones and daily income (periodically available through the effective income of parking meters).

The remaining element needed for capturing the parking system situation as a whole are the cost-free parking area. For these parking areas the data needed can be collected like described in the case of the managed parking slots. Traffic aides provide the information that will enhance the database on the occupancy rate condition.

The totality of the information will provide the mobility manager the necessary overview and instruments to analyse the actual situation of the urban parking system and simultaneously will equip the dashboard with a continuously updated database on the occupancy situation of the urban parking areas.

For those parking areas that are considered in the experiment as Park & Ride areas, the monitoring of and guidance app users and their planned trips will play a basic role for the experimental evaluation. In particular for those trips including Park & Ride solutions the control panel will show these data: number of users that utilize Park & Ride solutions, type of vehicle used for ride side, amount of kilometres saved, see Table 10.

Table 10: Data for experimental park & ride evaluation

PARK CATEGORY	SOURCE	INPUT DATA	OUTPUT DATA
PARK & RIDE	App. route planner	<ul style="list-style-type: none"> Trip planned with Park & Ride solution (parking slot ID, date and hour of planned trip, type of transport for the ride side, ride km planned) 	<p>The number of users that plan through the app Park & Ride solution (for area ID, period of time, type of transport used for the ride side).</p> <p>Estimate of saved kilometres thanks to these trips (for area ID, period of time, type of transport used for the ride side).</p>

4.2.5.3. Evaluation and Analysis

The database including the historic occupancy rate, provided by the access control system of car parks and by traffic aides surveys, will be used to produce an average occupancy rate of a specific parking area. This average value will be calculated considering specific time bands as a particular hour of the day, morning, afternoon, evening, a day of the week, producing a reference value to use to estimate the occupancy condition of urban parking areas.

The occupancy value calculated in this way is an indirect expression of the access cost to different parking slots and can be used to determine an additional cost for the planned trips through the journey planner in order to discourage car use to reach central areas and strengthen the Park & Ride service.

Effectively, the mobility manager will be able to display on the map a set of objects standing for urban parking slots. These objects can be queried providing the following information according to their category:

GARAGES – Car Parks

- number of public parking slots
- number of available slots (data from AMR entries control system)
- average occupancy in the morning time band (10-12)
- fare per hour
- average daily income

PARKING AREAS WITH FEE ON THE STREETS

- number of public parking slots
- number available slots (data obtained by previous data collected depending on the time band)
- average occupancy in the morning time band (10-12)
- fare zone
- fare per hour
- average daily income.

CARDBLOCK CLOCK PARKING

- amount of public parking slots
- amount available parking slots (data obtained by previous data collected depending on the time band)
- average occupancy in the morning time band (10-12)

PARKING AREAS NOT WITH FEE ON THE STREETS

- amount of public parking slots

- amount available slots (data obtained by previous data collected depending on the time band)
- average occupancy in the morning time band (10-12)

In the same way, the whole urban area will be divided and analysed by homogeneous territorial parts (parking zones). Every parking zone will be represented by an object that includes the following information:

- number of public slots divided for type
- number of occupied slots
- number of available slots
- occupancy coefficient per homogeneous area
- total income for the parking system from the beginning of the year

Both for a single parking area and for homogeneous territorial zone, the mobility manager will be able to obtain a detailed analysis and data visualisation on the map, selecting a specific period of time (using data previously collected when real time data are not available):

- working/holiday day
- time band (morning, afternoon, night)
- possibility to choose the most recent data, the average data, the average data referred to the last month

Furthermore we will monitor for those parking spots pinpointed in the experiment as Park & Ride areas, the planned trips by the app users. Also we will define an amount of carbon friendly kilometres travelled. A direct relation between the number of saved kilometres and the amount of CO₂ emitted can be calculated through certified calculation methodologies as CORINAIR (European Environment Agency). This procedure allows to estimate the amount of saved CO₂ consequent to planned trips that include Park & Ride solutions. Data displayed in the control panel will be the following ones:

- number of occupied slots in the selected parking areas
- number of users that utilize the option P&R through the app
- saved kilometres by car estimate
- saved CO₂ estimate

On the basis of the information included in the object of the map as described in the above paragraphs, the mobility manager will be able to get a set of thematic maps showing the parking system condition. The end of this paragraph shows a map referring to the occupancy rate of parking areas that are reported. A chromatic gradation shows increasing occupancy rates, analogous to what is shown in Figure 18.



Figure 18: Occupancy rate map in Rovereto (Source: Urban Mobility Plan of Rovereto)

4.2.6. Bike sharing

4.2.6.1. Introduction and Motivation

In the bike sharing system the control panel is responsible to provide the mobility manager with useful instruments for the evaluation and the balance of the system. These analysis instruments enable the mobility manager firstly to improve the efficiency of the bike sharing, and secondly to elaborate an estimation of the positive effect that bike sharing can have on modal shift and on saving CO₂ emissions.

The control panel will achieve these results through the visualization of a summary report that shows features and performance of the bike sharing service, and uses these specific indicators:

- number of bike sharing stations;
- number of available bikes (real time data);
- number of users registered on the system;
- total number of bike withdrawals;
- average use time;
- estimate of km covered with bikes belonging to the bike sharing service;
- estimation of CO₂ saved as a consequence of travelling in sustainable means of transportation

The control panel equips the mobility manager with an integrated visualization of this data regarding the bike sharing system to detect critical points and opportunities that will enable the mobility manager to enhance the performance and efficiency of the aforementioned bike sharing system. At the same time the mobility manager will be able to evaluate the sustainability contribution given from the system in terms of saved CO₂ emissions.

The method used to obtain bike sharing data, in particular the monitoring of the bike identification number, will enable the control panel to identify the frequency level of trips between different stations, and that is a useful indication of the main routes followed by users to travel inside the city.

4.2.6.2. Data

Through the acquisition of bike sharing service management data, the mobility manager will be able to obtain all the information needed regarding the bike sharing service given by the municipality.

Particularly, the control panel will localize on a map all bike sharing stations; for each station it is possible to get real time information regarding: number of bikes, number of bikes taken and returned to the station (to every event that takes place in the station will be associated date, time, and ID number of the bike). The system will also notify about malfunctions and charge level of the single bikes located in the stations. The necessary data is listed in Table 11.

Table 11: Data for bike sharing

BIKE SHARING	SOURCE	INPUT DATA	OUTPUT DATA	DESCRIPTION
SERVICE FEATURES	Service Manager	<ul style="list-style-type: none"> Number of stations Number of bikes for each station Number of users registered on the system 		Detailed figures of the Bike sharing service used in the system analysis process.
OPERATION LINKED TO TAKING OR RETURNING BIKES	Service Manager TO remote control OR	<ul style="list-style-type: none"> Bike withdrawal (ID number of the station, date and time of the event, ID number of the bike). Bike return (ID number of the station, date and time of the event, ID number of the bike). 	<p>Number of available bikes (categorized by ID number of the station, time period of the day).</p> <p>Number of bikes being used (categorized by ID number of the station, time period of the day).</p> <p>Number or percentage of bikes taken and returned</p>	<p>Statistics on the number of used bikes or available bikes for each Bike sharing station, related to a particular hour of the day, or a particular moment of the day (eg. morning, night), or a particular day of the week.</p> <p>Statistics on the number of bikes taken or returned for each Bike sharing station (also as a percentage of the total number of</p>

(categorized by ID number of operations of the system), related the station, time period of the to a particular hour of the day, or a day, or as a percentage of particular moment of the day (eg. total operations of the morning, night), or a particular system). day of the week.

Intensity of OD relationships Intensity of OD relationships (categorized by ID number of calculated using the connection the station, time period of the between ID number of the bike day). The OD matrix and the ID number of the station represents the frequency level where the bike has been taken and of relationship between returned. The same data will be stations. used to calculate the km travelled in each trip and the amount of saved CO₂ emissions.

4.2.6.3. Evaluation & analysis

Each bike sharing station displayed on the map will include the following features:

- address of the bike sharing stations
- number of available bikes in each station
- number of bikes used
- number of bikes taken from each station (daily average)

Data displayed on the map will show the daily average value, but when necessary, specific time bands could be considered, as morning, afternoon and evening, or a specific day of the week, or working/holiday day, generating a reference average value related to the chosen period.

All this information allows to produce statistics, displayed on the map for each bike sharing station. These attributes involve the number of bikes taken and given back and are displayed through pie diagrams showing the bike station balance.

Furthermore, the dashboard will display the state of each bike sharing station related to the number of total implemented operations in order to enable a clearer perception of the contribution given by each station. This procedure occurs relating the number of bikes taken from and given back each station with the number of bikes taken from and given back considering the whole system.

The knowledge of each bike identification number allows the system to associate the origin and the destination of each trip. This feature, which will be introduced in the dashboard during the second experimentation, allows the system to compute the intensity of the relationship between stations. This feature will enable the mobility manager to see which stations are the ones with the highest total turnover of bikes, and which are the ones where the number of withdrawals and deliveries is smaller.

The information about every trip per bike could be used to obtain the kilometres travelled for each trip. The amount of travelled kilometres and the saved CO₂ can be set in correlation. This will enable the dashboard to define an environmental performance indicator.

Data described above will be used by the control panel to produce a summary report showing attributes and performance of the service:

- number of bike sharing stations
- number of bikes
- number of available bike on real time
- number of subscribed users
- number of the total bikes taken from stations
- average time use of a bike
- estimate of kilometres travelled with bike sharing system
- amount of saved CO₂ consequently to the choice of a sustainable mode of transport

4.3. Guidance

The Mobility Management and Emission Control Panel - MMECP is not only about monitoring and analysing the current traffic situation. The mobility manager will be able to carry out operations that give guidance to a various set of subscribers of the system using the dashboard. A subscriber is a user of the STREETLIFE system, an App, event organizer or any other system that needs to receive actual traffic information; considering the scenarios defined in section 3, subscribers include citizens in their modal choice, users of an App on a specific route or an event organizer meaning to plan the arrival and departure of visitors...

Guidance information provided by the MMECP contains detailed traffic information about a specific urban area and includes recommendations on how to act in that area with respect to the given use case. Within STREETLIFE the MMECP will be used for the three pilot cities BER, ROV and TRE, but it is not designed to solely work with this restricted group of cities. The MMECP is designed to support various cities all over Europe. Through configuration each city can apply its scenarios and use cases. Figure 19 shows the generic process of giving guidance.

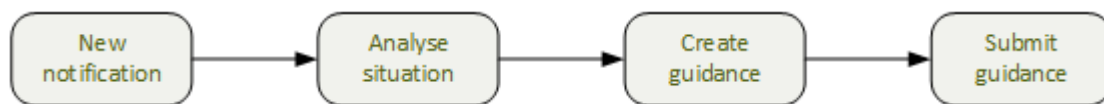


Figure 19: Generic guidance process

The first process part diagrammed “new notification” is not mandatory; the mobility manager can give guidance without receiving beforehand a notification that triggers the rest of the guidance process. Depending on the configured city and its use cases he will get specific information on a certain situation and specific possibilities to give guidance. Section 4.3.1 gives a detailed example how the MMECP will be used in ROV.

4.3.1. ROV example for giving guidance

As an example of Guidance we illustrate how the system will work in the ROV scenario, in particular scenario ROV-PR/4 from D6.1 [1, Ch. 3.3.3.4]:

- Using the apps, that are connected to the STREETLIFE system users can query journey planners to know the different possible trip option to reach their destination;
- Thanks to the analysis of the parking system, the mobility manager will be able to use the dashboard to insert in the system additional information related to extra time (delays), specific trips or access in particular urban area. Moreover the mobility manager will send a textual notification to users on special events like accidents, traffic blocks, events, market, etc.
- Besides standard information concerning the available different modes of transport or combinations thereof, users will be informed about events that occur on their planned trip and will become aware of the environmental impact (CO₂ production) that the choice of each of the possible available options involves;
- Therefore users will be able to select and confirm the most sustainable option trip in case of a planned trip, or confirm in real time their choice if they received a notification trough the app while they are having their planned trip.

The mobility manager, through the dashboard features that are described below, will be able to operate in the system and guide subscribers towards a sustainable choice.

The mobility manager is equipped with different instruments to elaborate data and check parameters. These tools estimate time and delays that, later on, can be automatically taken as input by the journey planner. Otherwise these tools convert time and delays in notification messages associated to the trip options elaborated by the journey planner. These instruments deal with procedure listed in the following paragraphs.

The mobility manager, or the dashboard operator, will be able to exclude a specific road segment until the end of the event both for a single episode (e.g. car accident) and recurrent or planned event (weekly market, working site, restricted traffic zone, etc.). Police officers will contribute data to this feature for those events that requires an immediate intervention (e.g. car accidents). This information will be at the same item acquired directly by the journey planner and displayed on the control panel. In case of planned events, the mobility manager will exclude, directly through the control panel, the interested road segments from the journey planner calculation. A calendar can be associated to this feature. In this way the mobility manager could pick the period and duration of the event and consequently inform users with a notification through the app. The journey planner will update its calculations considering variations related to roadway network.

Journey cost estimates of street networks enable to calculate the time needed for every single trip option. The mobility manager will be able to control a specifically created matrix that includes for every road segment the journey duration value without traffic (off-peak period) and with traffic assignment as provided by the simulation process (peak period). This matrix will show an extra time for each road in the off-peak and peak-hour traffic as a estimation of time lost due to traffic and congestion and the extra time will be considered by the journey planner as well. . The mobility manager, if needed, will be able to assign a cost factor to an area in order to automatically give this value to the road comprehended in the area (e.g. the historical city centre). The necessary information for this process are listed in Table 12.

Table 12: Cost factor for traffic estimation

COST FACTOR	DESCRIPTION
A	Node A of road segment
B	Node B of road segment
LENGTH	Length of the road segment [m]
SPEED_SET	Speed set in the model [km / h]
SPEED_HP	Speed during peak hour (provided by the model) [km / h]
FACTOR_HP	Peak hour correction factor set up by the mobility manager

T_HP	Travel time during peak hour [min]
FACTOR_OFF	Off-peak hour correction factor set up by the mobility manager
T_OFF	Travel time during off-peak hour [min]

This functionality will be implemented during second experimentation. Estimation of time (and costs) related to parking search in a specific urban area. The database related to parking supply will be constantly updated and enhanced with data collected during traffic aides surveys or automatically collected data from parking areas with a control access system. This process will enable to associate to the different day of the week and different time bands an average occupancy rate for each homogenous parking zone previously defined. Based on the occupancy condition in a specific moment (the default option selected is the one referred to the average value of the available data) the system will produce an "access cost" indicator connected to a parking zone. This cost indicator, calculated by the system with a pre-set parametric table (Table 13), will be provided as an input to the journey planner. The mobility manager will be able to modify the value of this indicator in order to control the parking demand in particular circumstances. When users chose a destination, the system set to the car trip the extra-cost related to the parking zone where the destination is located. Users will receive this information as an alert that approximate the extra-time spent searching a parking slot.

Table 13: Access cost indicator

OCCUPANCY RATE	PARKING SEARCH DIFFICULT LEVEL	EXTRATIME ESTIMATION
10-60%	NULL	0
60-70%	LOW	0-1
70-80%	MEDIUM	1-3
80-90%	HIGH	2-5
90-100%	VERY HIGH	3-10
>100%	IMPOSSIBLE	5-15

4.4. Simulation

Testing the STREETLIFE system and apps in the real world is not easy. For a realistic evaluation, a large number of test users (vehicle drivers, bicyclists, and pedestrians) – moving in a large-scale area, all actively using the STREETLIFE system – is needed. Consequently, such an in-the-field validation, testing and evaluation process might be too expensive. Having in mind the available resources, simulations provide an essential tool to facilitate the overall process and to create the technical foundations of this technology.

4.4.1. Berlin simulation scenario

4.4.1.1. Scenario description

4.4.1.1.1. Simulation Tooling

The simulation of the STREETLIFE BER simulation scenario is a challenge because different simulation worlds need to come together: e.g. traffic simulation (vehicle and bicycle movements), communication simulation (communication between STREETLIFE system and users to inform them about possibilities to circumnavigate dangerous road segments and intersections), and application modelling (modelling of the application to optimize the travel routes for more safety). To solve this problem, the V2X Simulation Runtime Infrastructure (VSimRTI) was chosen to couple different simulators and, thus, to enable the simulation of the various aspects, see [12]. VSimRTI has been developed by Fraunhofer FOKUS in cooperation with the Daimler Center for Automotive Information Technology Innovations (DCAITI). This tool is available to all STREETLIFE partners for their simulations.

VSimRTI is a comprehensive framework for the assessment of new solutions for Cooperative Intelligent Transportation Systems. Vehicle movements and communication technologies can be modelled in detail. VSimRTI couples different simulators to allow the simulation of the various aspects of future Intelligent Transportation Systems. The easy integration and exchange of simulators enables the substitution of the most relevant simulators for a realistic presentation of traffic, emissions, wireless communication, and the modelling of mobility applications.

VSimRTI technology has a comprehensive framework for simulator integration that facilitates the simulation of Smart Mobility scenarios. All simulator management tasks, such as synchronization, interaction and lifecycle management are handled completely by VSimRTI. Several optimization techniques, such as optimistic synchronization, enable high performance simulations. Special features, e.g. traffic lights, roadside stations, and CAM and DENM transmissions, are supported by VSimRTI. Moreover, various configuration, visualization and analysis tools assure maximum usability. In contrast to existing fixed simulator couplings, the VSimRTI simulation infrastructure allows the easy integration and exchange of simulators. Thus, the high flexibility of VSimRTI enables the coupling of the most appropriate simulators for each use case. Depending on the specific requirements of a simulation scenario, the most relevant simulators can be used. VSimRTI uses a simulator coupling concept inspired by some fundamental concepts of the *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*. Thus, it is possible to couple arbitrary simulation systems with a remote control interface. Attaching an additional simulator only requires that the ambassador interface is implemented and, then, the specified commands are executed.

4.4.1.1.2. Description of the Berlin simulation scenario

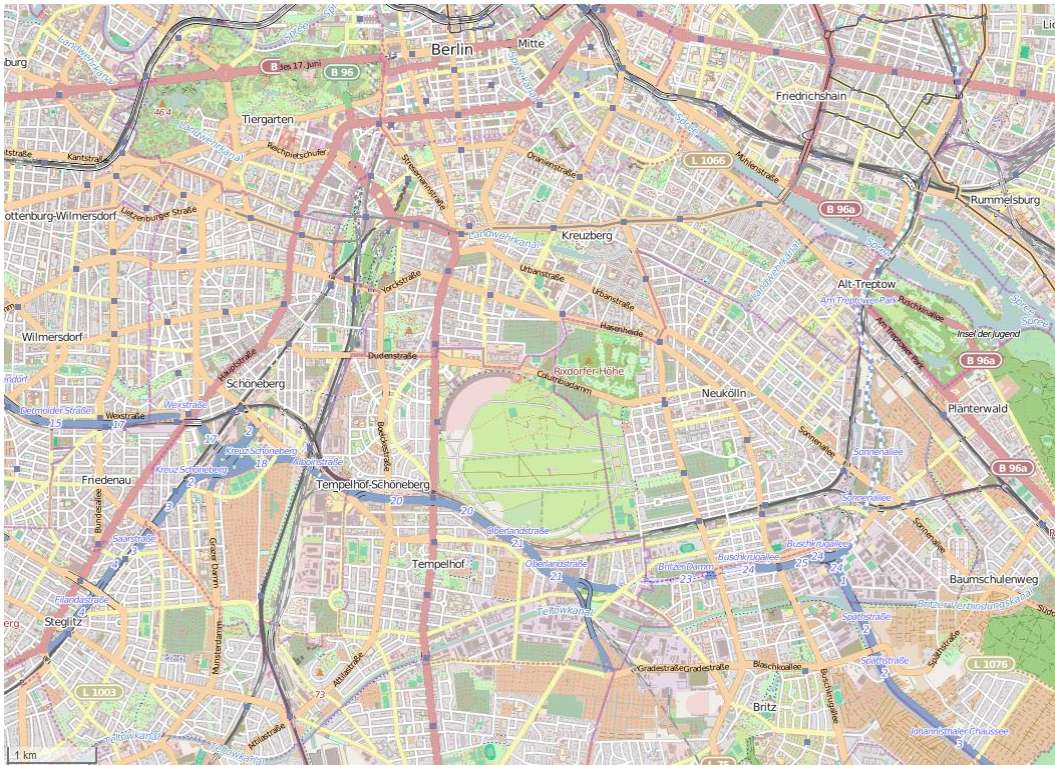


Figure 20: Area of the BER simulation scenario (Source: <http://www.openstreetmap.org/>)

The BER simulation scenario covers an area of about 75 square kilometres. The center of this area is the former Tempelhof Airport, which is now known as *Tempelhofer Freiheit* and is one of Berlin's most significant event locations, see Figure 20. With its two take-off and landing strips, the emerging Tempelhof Park offers over 300 hectares of open space for a number of events, including sport tournaments, open-air concerts and art and educational projects.

The BER simulation scenario is to model the situation after a big event at *Tempelhofer Freiheit*. A huge number of people leaves the event area and is willing to go home. Several people will use a bicycle for their trip home. Since a huge number of other bicycles, vehicles, and pedestrians are on their way in the considered area after the event, many dangerous situations can occur for bicyclists. To reduce the risks to bicyclists, the bicycle router of the VMZ ([13, Sec. 2.2]) is to be used to suggest each bicyclist safe routes that circumnavigate dangerous road segments and intersections. One aim of the BER simulation scenario is to detect which safety benefit can be achieved for the bicyclists by using the VMZ bicycle router.

4.4.1.1.3. Preparation of the BER simulation scenario

OpenStreetMap¹ (OSM) is used as source for the road network and infrastructure data of the BER simulation scenario. OSM is a collaborative project to create a free editable map of the

¹ <http://www.openstreetmap.org/>

world. Since 2004, it has grown to over 1.6 million registered users, collecting the data. The crowd-sourced data is made available under the Open Database License.

For the preparation of the simulation, OpenStreetMap data is converted to the data format of the simulation framework. Furthermore, additional pieces of data, needed for the simulation scenario, are added manually: e.g. bike sharing stations.

In addition to the road network and infrastructure data, Origin-Destination (OD) Matrices and the Modal Split are needed. Origin-Destination Matrices specify the travel demands between the different origin and destination nodes in the traffic network and the Modal Split represents the percentage/number of travellers using a particular type of transportation. With both information, a realistic movement of the travellers can be modelled by the simulation framework. For the BER simulation scenario, the Origin-Destination Matrices and the Modal Split are to be generated by the microscopic travel demand model TAPAS (Travel Activity Patterns Simulation) by DLR, see [14].

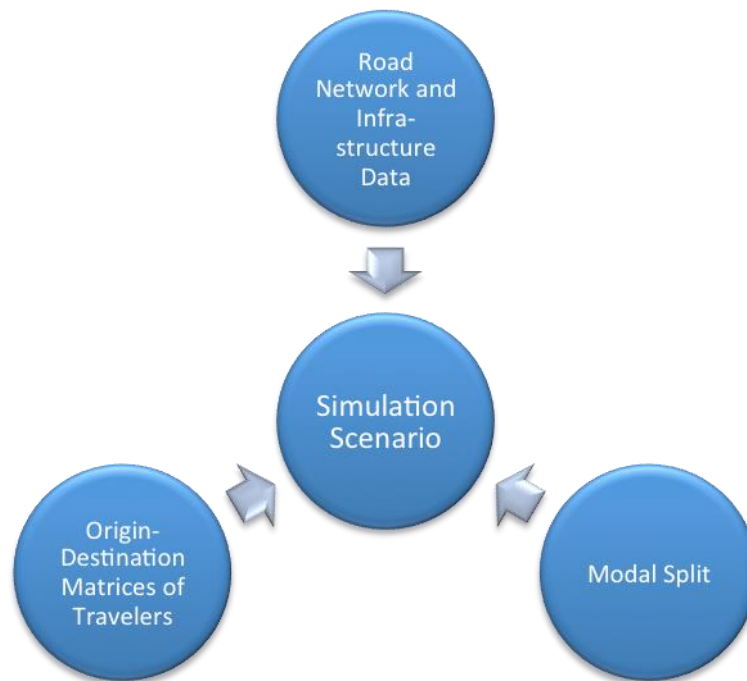


Figure 21: input data needed for the preparation of the BER simulation scenario

4.4.1.2. Scenario Key Performance Indicators

As the BER simulation scenario is not yet finalized, Table 14 shows the indicators that are to be considered as a first collection of possible key performance indicators (KPI) at the stage of the first pilot iteration. After the deployment of the first iteration of the BER pilot, this list of KPI will be adjusted, specified and amended if necessary.

Table 14: BER simulation KPIs

ID	INDICATOR	TYPE	METHOD	METRIC	DATA SOURCE
1	Modal Split for reaching and leaving the event against normal conditions	Indirect	Calculation	Percentage per mode of transport	Simulation output aggregation
2	Number of re-routed STREETLIFE users	Direct	Measurement	Count	Needs to be defined
3	Number of available bicycles	Direct	Measurement	Count	Simulation constraint
4	Number of booked bicycles	Direct	Measurement	Count	BER App
5	Average travel time of event visitors	Direct	Measurement	Seconds	Simulation output
6	Remaining capacities of public transport	Indirect	Investigation	Count	Public Transport operator

7	CO ₂ saving estimation	Indirect	Calculation and scale-up	Kg of saved CO ₂ emission Simulation per (event, capita, service aggregation penetration, etc.)	output
---	-----------------------------------	----------	--------------------------	--	--------

4.4.2. ROV simulation scenario

4.4.2.1. Scenario description

Using simulation devices in the framework of ROV Park & Ride scenario aims at assessing the impact of a new parking lot in terms of vehicular circulation improvement (reduction of congestion in central areas) or reduction of demand parking pressure in the historical centre.

The simulation procedure implies the definition a set of features of the new parking slot which the mobility manager needs to define in advance (on the basis of town planning, traffic, economic evaluation). The information that needs to be provided by the mobility manager for the simulation of a new Park & Ride are the following ones:

- location of the new Park & Ride area;
- number of slots calculated on the basis of the surface area.

Once these data is established, it can be deduced even through the analysis of similar parking slots in the city, the mobility manager will have all the information needed by the simulation model in order to calculate the impact of a new Park & Ride area.

The key aspect in this process will occur outside the control panel with specific features of Cube 6.0 software, by Citilabs, see [15]. Cube will be used in the second iteration for the Rovereto pilot simulation. It is already described in D4.1 [16].

A traffic model has been developed for the Urban Mobility Plan of the city of ROV with the aid of Cube 6.0 software (Citilabs) so as to analyse phenomena related to mobility system in the town territory.

A roadway network has been digitalized in an area that goes beyond the town boundaries and includes nearby municipalities with their principal traffic lines. This network is composed by a primary network for its entire extension (motorway, state and province highway), while for the urban area of Rovereto has been specified at finer granularity, with all the local streets. The network includes arcs and nodes that represent respectively streets and intersection. These elements are provided with geometric and traffic features (length, speed, capacity, speed flow curves, intersection category, traffic lights operation chart, etc.)

The model contains an OD matrix with journeys carried out with cars and heavy vehicles. This matrix has been built with estimate techniques based on socio-economic data and results from ad-hoc surveys. During the simulation process the matrix is subject to adjustments in order to represent with an adequate accuracy the state of the art.

The traffic model has been developed for the peak-hour in the morning to evaluate a critical situation - related to the worst traffic condition detected in an average working day.

The simulation of a new parking area is equipped with Park & Ride facilities will follow these steps in the traffic model:

Identification of OD relation interested by the new parking lots uses a procedure that leverages a Cube feature that allows to determine the number of vehicles passing on a

specific road with a particular origin and destination (in Figure 22 the road segments represented with a thicker trait are most used). The mobility manager will be able to query the model on the number of vehicles that are directed to the city centre and transiting on the road where the new Park & Ride is supposed to be located;

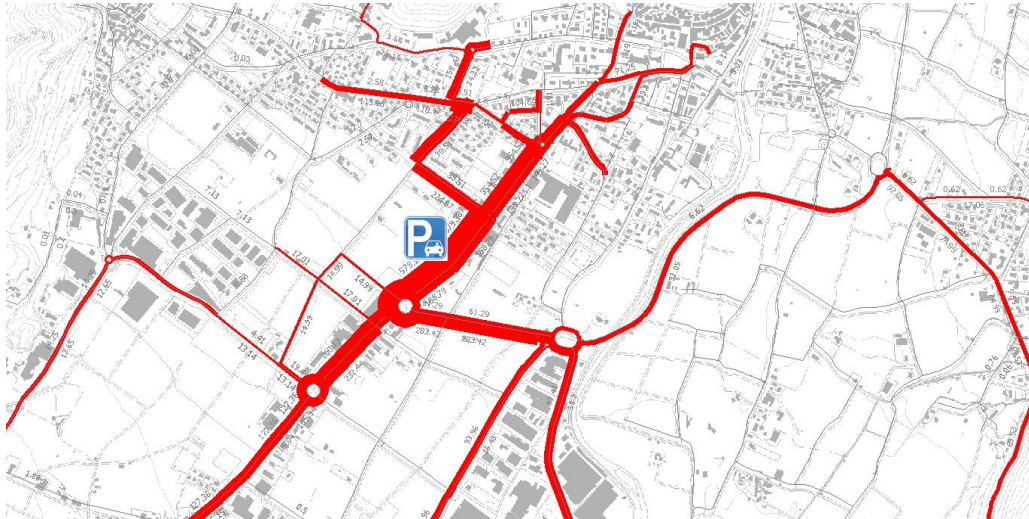


Figure 22: Traffic flow on a specific road segment (Source: Urban Mobility Plan of Rovereto)

The main steps of the Park & Ride simulation scenario will be developed according the following sequence:

- Assignment process definition of a reliable percentage of vehicles that can intercept the new Park & Ride area in relation to the total vehicles (calculated with the method described above) and facilities available in the new parking area;
- extrapolation of the part of the OD matrix that includes those trips that in the present situation are addressing the city centre and set as new destination of these trips the new Park & Ride area;
- creation and insertion of a new centred Park & Ride in order to represent the new destination zone for those trips that have been pinpointed in the previous step;
- new traffic assignment procedure run by the transportation software; a part of roadway network can be extrapolated with specific speed and time recalculated for this scenario. This procedure let the software define the amount of saved kilometres related to the previous situation and calculate the measure of saved CO₂.

Once completed the assignment process in the traffic simulation model with the Cube software, the mobility manager will be able to continue the simulation producing the following results:

- Estimate of CO₂ saved in relation to the new Park & Ride. Amongst the output of the assignment process that the Cube software automatically provides, there is an indicator of total kilometres travelled on the network. The amount of saved CO₂ thanks to the new Park & Ride, will be calculated considering the difference between the kilometres travelled in the new scenario and those ones travelled in the present situation. Alternatively and in a simpler way, the number of kilometres not travelled on the network can be defined as the product of number of trips intercepted by the new Park & Ride area and the average

distance of this area from the city centre (the original destination of those journeys). For both solutions, firstly an amount of saved CO₂ related to the peak hour will be calculated. Later this quantitative, thanks to a specific coefficient to adopt for the number of vehicles that use Park & Ride during the peak hour, will be converted in a daily or annual value;

- Recalculation of occupancy rate in central areas that are less occupied thanks to the new Park & Ride. This operation is necessary to evaluate a possible reorganisation of the parking system (e.g. reduction of parking slots in the city centre) and estimate the decrease of time spent for finding an available parking slot. The number of estimated vehicles intercepted during the peak hour by the new Park & Ride area will be subtracted from the number of cars that determine the occupancy rate in the city centre. The occupancy rate for other time bands is determined using parametric calculations. The new thematic map with the estimated occupancy rate related to each homogenous parking zone, for a specific time band, will be displayed in the control panel.

4.4.2.2. Scenario Key Performance Indicators

Based on the ROV scenario description the following KPI can be specified, see Table 15.

Table 15: ROV scenario key performance indicators

ID	INDICATOR	TYPE	METHOD	METRIC	DATA SOURCE	COMMENT
1	CO ₂ saving estimation	Indirect	Calculation and scale-up	kg of saved emissions per ? (event, capita, service penetration, etc.)	Simulation output aggregation	Most challenging!
2	Km travelled on network	Direct	Measurement	Km	Simulation output	
3	Number of vehicles using P&R	Direct	Measurement and calculation	Number	Simulation output and estimation	
4	Occupancy rate in central area	Direct	Measurement	Percentage	Simulation output	
5	Decrease of time spent for finding a park slot	Indirect	Calculation	Percentage	Simulation output and estimation	

4.4.3. TRE simulation scenario

TRE has not planned to use the MMECP simulation in the first phase of the project. Thus no simulation scenarios are defined yet. This is planned as task for the second phase of the STREETLIFE project.

5. DESIGN OF MMECP

This section describes the design for the MMECP software. The MMECP will be used for the STREETLIFE pilots, within their own instantiation of the STREETLIFE system, and in accord with the STREETLIFE blueprint architecture (see Deliverable D2.2.1 [2]). The next sub-sections are describing the design by means of a UML component diagram and some sequence diagrams, detailing certain significant interaction scenarios for the MMECP. This section also contains a technical description of the API MMECP relies upon to retrieve statistical data from the STREETLIFE data management subsystem for its analyses.

5.1. MMECP component diagram

The functional breakdown of the MMECP is shown in Figure 23. The UML component diagram is basically divided into two parts. The “Outside of MMECP” and the components that make up the MMECP.

The MMECP has no storage component; this is delegated to the databases that are provided by the pilots in accord with the STREETLIFE blueprint architecture. Thus in the “Outside of MMECP” a legacy *data storage* component is described. The term legacy in this diagram describes, from the perspective of the MMECP, an already existing component that will be used. Thus the “Outside of MMECP” has two black boxes, the data storage with its API and a subscriber component. In Berlin the City Intelligence Platform (CIP) is used as data storage, whereas in ROV a FIWARE-based data storage subsystem is used. They both must provide an API that is common, described in D3.2.1.

The following is describing the subscription mechanism inside the MMECP, which is a synchronous mechanism. Other subscribers like third party users will use this mechanism as well. However the STREETLIFE system will implement an asynchronous subscription mechanism as well, but this is not part of this deliverable, it is described in D2.2.1, see [2].

The MMECP implements a synchronous notification mechanism with subscribers. This mechanism is used by the MMECP itself and can also be used by the subscriber of the MMECP. An external subscriber of the MMECP is for example the ROV routing app, which is described in section 4.3.1. Each subscriber will have its subscription channel. This can be configured in the MMECP. Whenever a subscriber receives a notification this is called guidance. The other way around, a subscriber can push notifications to the MMECP. In this case the notification is pushed to a channel, the MMECP is a subscriber from and is called notification. This will be used to inform the mobility manager about issues in a certain area.

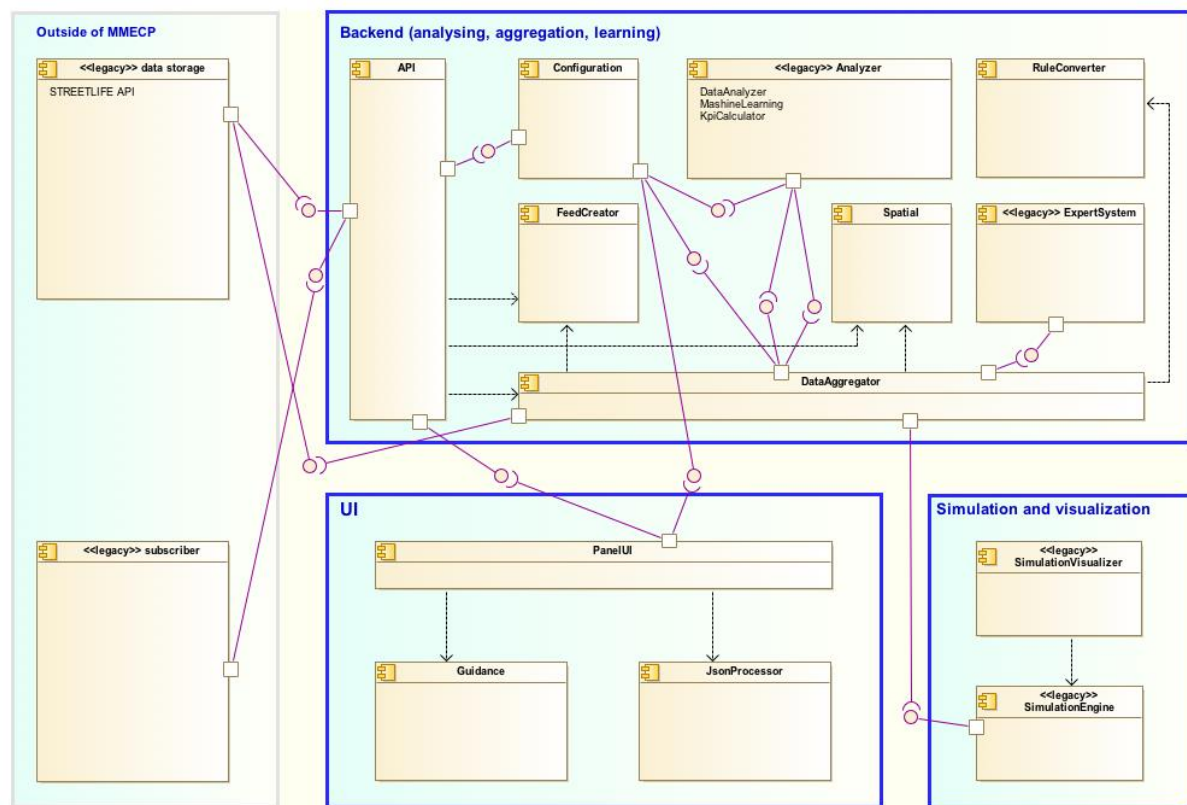


Figure 23: MMECP Architecture

The communication between the components is realised via HTTP RESTful services. Each communication from “The outside of MMECP” to the MMECP is handled by the *API* component in the MMECP *backend*. The *backend* describes a bundle of components that are responsible for the business logic of the MMECP and for the analytics tasks. The *API* is responsible to delegate each request to its responsible component. In this initial version of the MMECP these are the *Spatial* component for geo referencing tasks, the *FeedCreator* for handling notifications and guidance and the *DataAggregator* for the collection and aggregation tasks. A more detailed description of the *Spatial* component is provided in the chapters 5.4 and 5.5.

The *FeedCreator* transform the ATOM feeds into formats that can be handled by the MMECP and creates ATOM feeds out of the guidance the mobility manager wants to give to the subscriber. For the subscription and pushing of notifications or guidance, ATOM (Atom Syndication Format) feeds are used. ATOM is a standardised format for cross platform information exchanges, see [17]. It uses XML (Extensible Markup Language) for serialization. As already said a subscriber can be for example an app or an event organizer. Thus the guidance needs to be machine or human readable in case of the scenario. Both can be done with ATOM. The configuration of the MMECP decides whether the content is machine or human readable. Another advantage of using ATOM is the fact that there are many existing tools that can handle ATOM feeds, e.g. Microsoft Outlook or Mozilla Thunderbird. Even developers can use a broad number of libraries for almost all programming languages to process ATOM feeds in their applications. Further explanation can be found in the sub-sections below.

The *DataAggregator* is used to retrieve data from the *data storage*. This is done via HTTP RESTful services as well. After the data is received the *DataAggregator* delegates analytics tasks to the *Analyzer*. The *Analyzer* is responsible for the KPI calculation based on the metrics that are stored in the *Configuration*. For the second STREETLIFE iteration it is planned to use machine learning together with the data analysis to enhance the prediction. The *Analyzer* also creates the decision tree for the prediction. This decision tree will be handed over to the *RuleConverter*. The *RuleConverter* converts the decision tree into format that can be handled by the *ExpertSystem*. Based on the decision tree and the data that comes into the *Analyzer*, the *ExpertSystem* can create the forecast.

Besides the backend the MMECP consists of a frontend, called the UI and simulation part. The UI has three components, the *PanelUI*, the *Guidance*, and the *JsonProcessor*. The *PanelUI* is responsible for the visualization of the map- and statistical view. The communication between the backend and frontend is loose coupled via HTTP RESTful services through the backend *API*. For serialization JSON is used. The frontend doesn't need to aggregate or analyse the data. The JSON, which is retrieved by the *PanelUI* contains all necessary information that is needed for the visualization of the map- and statistical view. Thus it is JSON serialized it needs to be transformed to HTML. The *JsonProcessor* does this. *Guidance* is a component to create the guidance based up on the actions the user does on the map view. A more detailed description on how to create guidance and display data on the map- or statistical view is done the sub-sections below.

The simulation components are dedicated services that are used by the mobility manager separately. They are not further described in this initial version of the document. For the first iteration, only the scenarios for the simulation were discussed. The implementation of the scenarios and integration within the workflow of MMECP will be done in the second iteration.

5.2. Get historic data sequence diagram

Historic data is used for analytics, statistics and forecasts. Figure 24 shows a sequence diagram that explains how MMECP receives and processes historical data.

In an initial step the *DataAggregator* asks the data storage for possible data, methods and constraints to apply on the data. This is done through the generic data access API that is explained in Section 5.6 The next step is to call this data with the *getData()* method of the same API. The *DataAggregator* receives the data as a multidimensional array of scalars. The *Analyzer* will analyse the data and ask the *Configuration* for the metrics to calculate the pilot or use case specific KPIs. Thus the *DataAggregator* questions the *ExpertSystem* for prediction if the data is describing a cultural event. The now enriched data needs to be converted to proper JSON serialized data that can be processed and rendered by the UI. Thus a specific JSON schema needs to be developed.

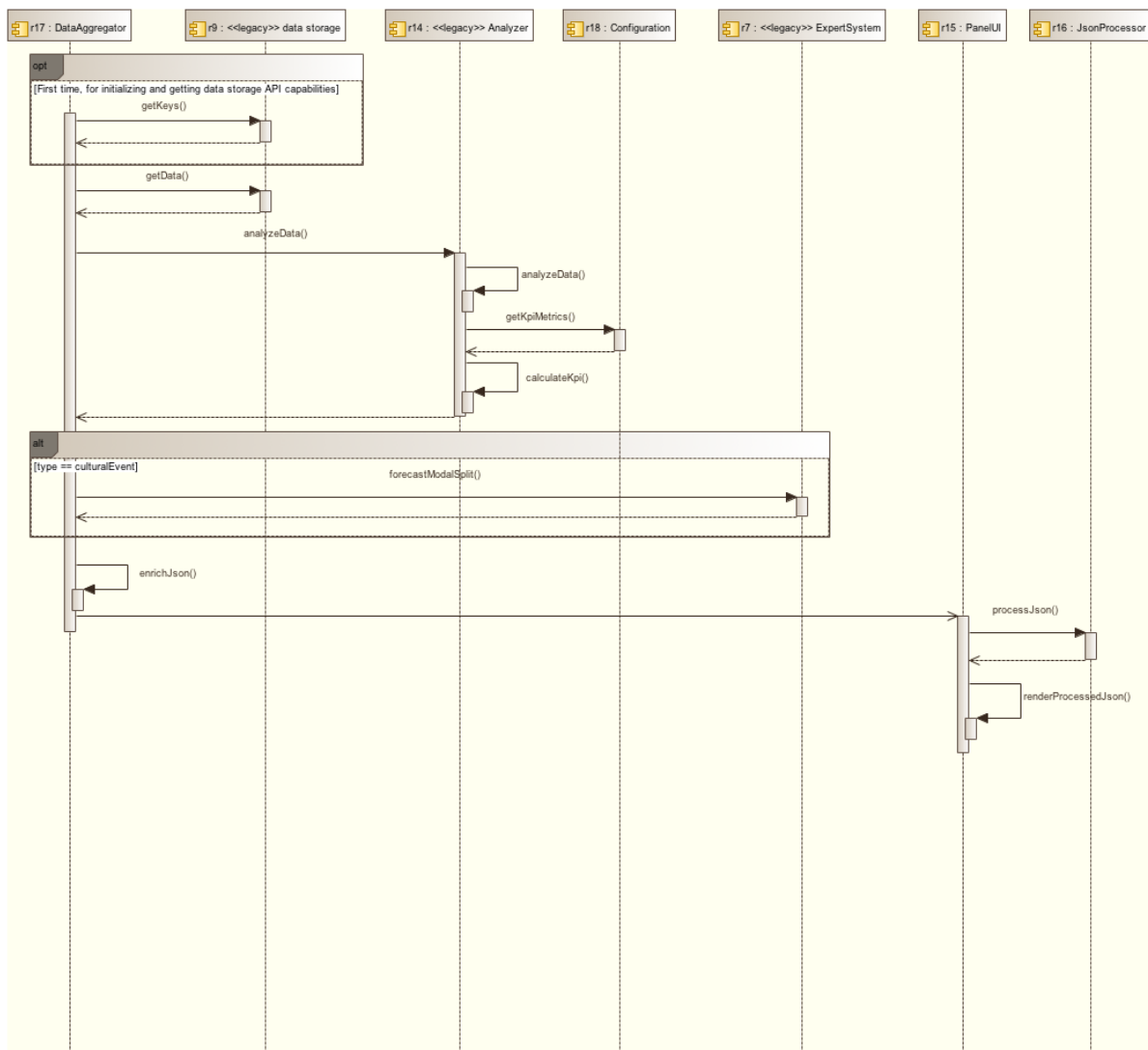


Figure 24: Get historic data sequence diagram

5.3. Get Feed Entry sequence diagram

Feeds can be pulled from the subscribers or the MMECP system itself. The MMECP configures a channel for each subscriber. This enables the MMECP to provide recipient specific guidance or notification if the subscriber is the MMECP system. Thus the actual data is not stored in the MMECP, the MMECP acts as a proxy. The subscriber asks the MMECP API to get the feed. The DataAggregator proxies the request to the DataStorage API. If the subscriber is the MMECP itself the response is converted to a Java entity that can be handled by the MMECP, otherwise the response goes back to the subscriber. This process is shown in Figure 25.

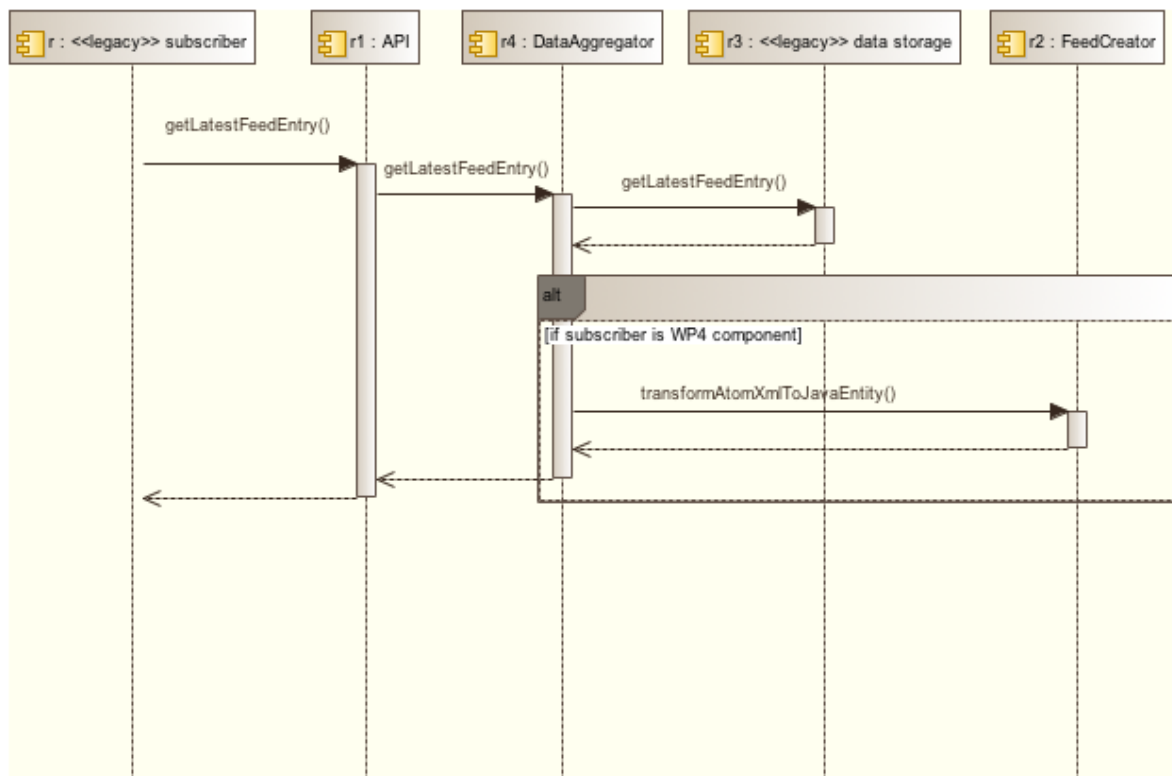


Figure 25: Get Feed Entry sequence diagram

5.4. Show statistics sequence diagram

To show the statistics view in the MMECP certain steps may be necessary. The user can select an area on the map view.. This location is used by the PanelUI to ask the API for spatial information about it. The location is then enriched with information e.g. NUTS code, street segments, names, etc. by the Spatial component and get back to the PanelUI. Then the next step is to retrieve the scalars that are needed to calculate the KPIs for the system. Thus the PanelUI requests the metrics for KPI calculation from the Configuration. Now its setup to request the necessary scalars via the API and DataAggregator from the data storage. Last step left is to process the JSON response and render the HTML page of the statistic view. The user is able to apply more time constraints on the selected data. He can do this via the PanelUI. If the user does so, a new request asking for the scalars is triggered. Default time constraints are configurable. Figure 26 shows this process in a sequence diagram.

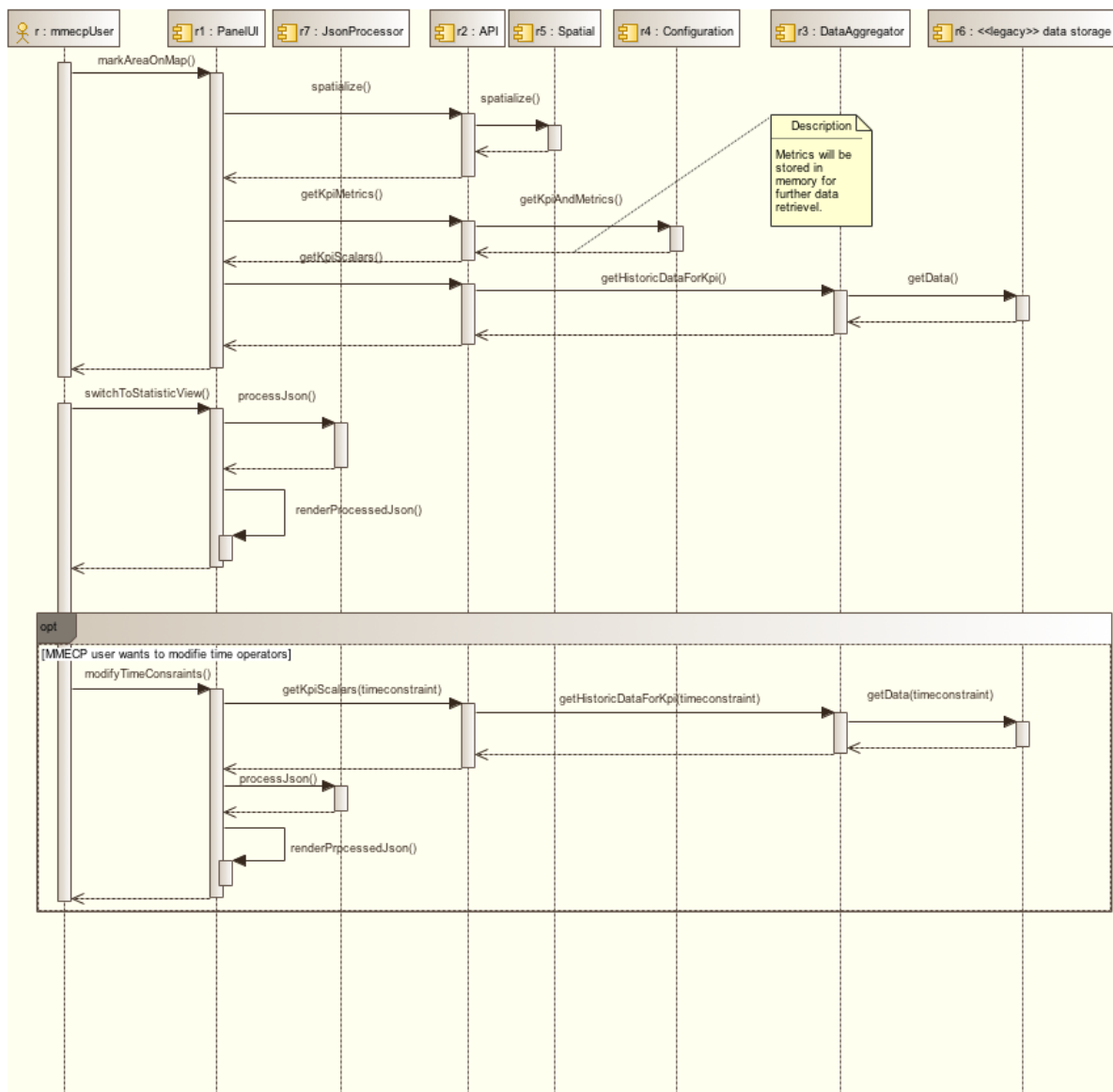


Figure 26: Show statistics sequence diagram

5.5. Create and post Guidance sequence diagram

The creation and posting of guidance typically occurs after the mobility manager has observed a certain situation on the MMECP UI. The user has to select the area on a map, which covers the content the guidance is about. As already said in section 5.4, the selection of a polygon is for realization in the second phase. Thus the user selects a coordinate on the map. The location needs to be enriched with spatial information. The user decides to whom he wants to give the guidance. This is done by posting the guidance to a certain channel. These channels receive the guidance in different formats. Actually it is always ATOM, but the content tag is different. A channel that is subscribed by event organizers needs human-readable content; conversely, a channel that is subscribed by an application needs machine-readable content. Next the PanelUI asks the API to provide a list of registered channels. The mobility manager then chooses the channel he wants to give the guidance. By asking the Configuration the PanelUI receives all necessary data to provide web-form to the mobility

manager. This web-form is partly prefilled accordingly to the selections the mobility manager has already done, e.g. spatial data. This form represents the guidance that the user can give after he submits the form. Submitting the form results in a HTTP PUSH request to the MMECP API. The API converts the Java entity to an ATOM feed and this feed is then pushed to the data storage. After this, the subscribers of the selected channel will notice this new guidance whenever they have a next look at the channel.

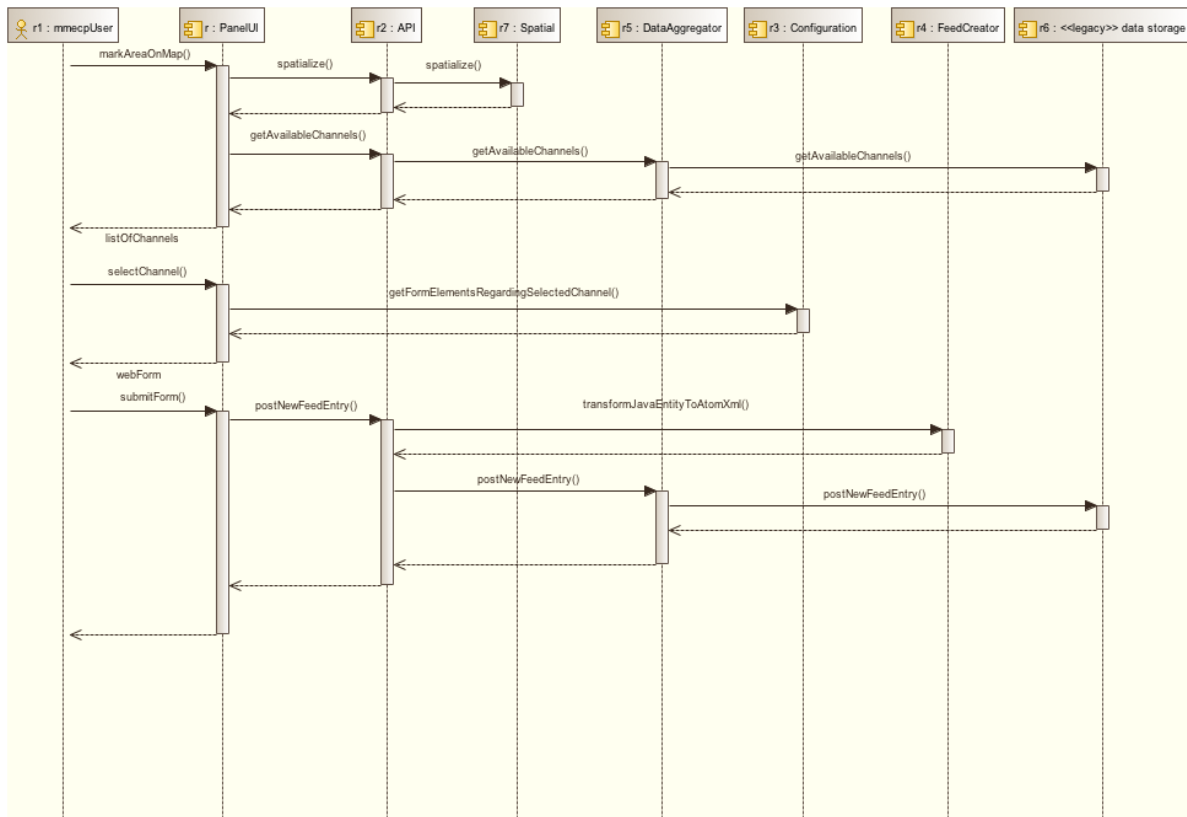


Figure 27: Create and post Guidance sequence diagram

5.6. A generic interface specification for standardized retrieval and evaluation of spatio-temporal data

The interface defined in the following sections provides a generic way to select, structure, aggregate, and retrieve spatially and/or temporally localized measurement data in a standardized manner. It is generic in the sense that it is neither specific to a particular type of database, nor is it specific to a particular programming language. The implementation in a particular programming language should be straightforward. A key feature of the interface design is that all the data can be arbitrarily distributed in space and time. It does not require that the data is arranged regularly in any way. In particular, the measurements of the quantities are not required to be synchronous or equally spaced. This said, for best practical use it is nonetheless often beneficial if the measurements were performed frequently and on a regular basis.

The interface proposed here requires only two functions. They can be implemented, for example, as service invocations:

1. an initialization function,

```
(list_of_functions, list_of_attribute_identifiers, list_of_condition_keywords) =  
get_keys(),
```

2. the actual retrieval function,

```
array_of_scalars = get_data(list_of_functions, list_of_attribute_identifiers,  
                             list_of_conditions0, list_of_conditions1, list_of_conditions2).
```

The types of objects required are arrays of floating point numbers and (ordered) lists. Technically, the minimum requirement for this framework is that a list contains a single element, which could even be just an integer number serving as an identifier. Correspondingly, the array can be zero-dimensional, containing just a single floating point number. However, for a more efficient use of the interface, we henceforth consider multi-element lists and four-dimensional arrays. As list elements we use text strings, which have the benefit that their meaning can be largely self-explanatory.

The initialization function **get_keys**() receives no parameters and returns three lists specifying the capabilities of the server with respect to the retrieval function **get_data**(). They effectively define the vocabulary understood by the data server. In trusted computing environments one might consider adding a *clientID* input parameter to **get_keys**() in order to provide different clients with different sets of capabilities. In untrusted environments one should rather resort to more reliable authentication.

list_of_functions:

The *list_of_functions* returned by **get_keys**() contains a list of all supported scalar aggregation functions (at least one) that the server can apply to each group of selected data. In particular, statistical functions can be applied. For example,

- “mean” – the arithmetic mean,

- “SD” – the (corrected) sample standard deviation,
- “n” – the number of selected/aggregated elements,
and for robust statistics and box plots:
- “median” – the median,
- “Q1”, “Q3” – the first and third quartile,
- “min”, “max” – the minimum and maximum.

Vice versa, the *list_of_functions* provided as a parameter to **get_data()** can be a (non-empty) list of any of these elements. A corresponding number of results will be returned accordingly.

list_of_attribute_identifiers:

The *list_of_attribute_identifiers* returned by **get_keys()** contains the list of identifiers of all accessible attributes in the database comprising a multitude of measurements of scalar quantities in space and/or time. Each individual measurement can be associated with a timestamp, a spatial location (e.g., latitude and longitude), both, or even neither of the two quantities. As an example, an attribute identifier might look like this:

“BikeSharingCompanies(FlexiBike).Stations(Berlin_Zoo).NumberOfAvailableBikes“

This identifier would be associated with measurements of the number of available bikes in a bike sharing station of a particular bike sharing company.

The *list_of_attribute_identifiers* provided as a parameter to **get_data()** can be a (non-empty) list of a subset of these elements. A corresponding number of results (in the corresponding order) will be returned accordingly.

list_of_condition_keywords:

The *list_of_condition_keywords* returned by **get_keys()** contains a list of all supported keywords that can be used to define filters for the selection of measurements to be aggregated and retrieved. The first type of these filters is one that selects measurements from particular time intervals. Such filters can be, e.g.,

- “time_of_day” – to specify a time interval within a day
- “day_of_week” – to specify a range of week days, e.g. Mon-Fri
- “day_of_month” – to specify a range days within a month, e.g. 1.-15.
- “week_of_year” – to specify a range of calendar weeks
- “month_of_year” – to specify a range of months within a year
- “year” – to specify a range of years
- “last_n_days”

Special keywords are

- “continuous_binning” – to specify a series of time intervals
- “all” – to impose no restrictions

The second type of filter restricts the selection of measurements spatially, e.g.:

- “within_distance_of” – to select measurements only within a certain distance from a given spatial location
- “within_area_of” – to select measurements only within a certain area

The third type of filter is a restriction with respect to the values of the measurements:

- “value_within” – to select measurements whose value is within a certain range
- “value_is” – to select measurements with a particular value

Finally, there is a more complicated fourth type of filter that conditions the selection of each measurement on the measurements of a different attribute:

- “corresponding_attribute”
- “corresponding_temporal_attribute”
- “corresponding_spatial_attribute”
- “corresponding_spatiotemporal_attribute”

The next section explains how to use these keywords.

list_of_conditions0, list_of_conditions1, list_of_conditions2:

Each of the three last parameters of **get_data()** contains a list of *conditions*. For each attribute provided to **get_data()** in the *list_of_attribute_identifiers*, each measurement associated with that attribute is aggregated in a cell of a two-dimensional table if it fulfils particular conditions given in the three lists of *conditions*. First of all, *all* conditions given in the *list_of_conditions0* must be met (the “global” conditions). Second, if the *m*th condition in *list_of_conditions1* and the *n*th condition in *list_of_conditions2* are fulfilled, then the measurement is assigned to the *m*th row and *n*th column of the table (the “local” conditions). Measurements can be assigned to more than one cell of the table if they fulfil more than one pair of conditions in *list_of_conditions1* and *list_of_conditions2*. After all measurements have been checked and assigned, each set of measurements (given in each cell) is processed by all aggregation functions specified in *list_of_functions*. In effect, we obtain a scalar result for each function in *list_of_functions*, for each cell of the table and each attribute in *list_of_attribute_identifiers*, making up a four-dimensional *array_of_scalars* returned by **get_data()**.

If the set of measurements in a cell of the aggregation table is empty, most aggregation functions are undefined and the *array_of_scalars* should reasonably contain a NaN at the corresponding position (“not a number”, IEEE 754 floating-point standard used for missing values). One exception is the counting function “n”, which returns a zero in this case. Another special case is the corrected sample standard deviation, “SD”, which is undefined also for a single measurement. In this case, one could either consistently return a NaN or revert to the uncorrected sample standard deviation, which is zero. The latter choice is somehow justified as the corrected sample standard deviation for two identical measurements is zero as well.

conditions:

Each individual *condition* can be either true or false with respect to a single scalar measurement. It consists of a keyword from the *list_of_condition_keywords* supplemented with a certain number of parameters. We will use the following syntax hereinafter:

condition_keyword(parameter1, parameter2, ..., parameterN)

(a) temporal constraints

Conditions related to the time of the measurements will mostly have one or two parameters, specifying a point in time or a time range, e.g.

- “time_of_day(10:00, 12:00)” – to select only data measured between 10 and 12, more precisely, for all data measured at times t fulfilling the condition $10:00 \leq t < 12:00$. Note that $t=12:00$ is excluded here to prevent the same data appearing twice in selections like “time_of_day(10:00, 12:00)” and “time_of_day(12:00, 14:00)”.
- “time_of_day(10:00)” - to select data from a single point in time, here 10:00.

Similarly, “day_of_week(Mon, Fri)” would restrict data to be chosen between Monday and Friday and “day_of_week(Mon)” would restrict data to be chosen from Monday only. In all cases except “time_of_day”, the second argument is meant to be inclusive, e.g. “year(2012, 2014)” selects data from years 2012, 2013, and 2014.

Alternatively, one might consider defining all time ranges uniformly in terms of two integer numbers: time in the format HH:MM could be defined as a number, HHMM (or even HH:MM:SS as HHMMSS). The days of the week could be enumerated from 1 (Monday) to 7 (Sunday).

Another alternative, instead of requiring a given agreement on the time specification format between client and server, one could extend the **get_keys()** function to return the time specification format expected by the server.

Using keywords with a variable number of parameters might be inconvenient in some cases. One can simply use more keywords instead, e.g., “time_of_day” and “time_of_day_range” or, less intelligible, define points in time as, e.g., “time_of_day(10:00, 10:00)”.

The following conditions are special:

- “continuous_binning(*start_time*, *time_interval*, *end_time*)” – this specifies a whole list of consecutive time intervals of length *time_interval* (in seconds), starting at a given point in time, *start_time*, and ending at *end_time*. Such a *time* can be specified as appropriate, for example as Unix time or ISO 8601 timestamp. Recommended is the use of “continuous_binning” as sole element in either *list_of_conditions1* or *list_of_conditions2*. It is useless in *list_of_conditions0* (where it should be evaluated consistently as an unfulfillable condition).
- Aside from statistical assessments, continuous binning can be used with “mean” or “median” to equidistantly (sub-)sample the measurement data and thus to obtain a regularized representation of the data.
- “all” – always fulfilled, to impose no restrictions (has no parameters).

(b) spatial constraints

Conditions related to the location of the measurements can be defined, e.g., as follows

- “within_distance_of(*location*, 10, 20)” – to select data measured within 10 and 20 meters distance of a certain *location*. A location can be specified as appropriate, for example as Cartesian coordinates or in terms of latitude, longitude and elevation. Alternatively, instead of requiring a given agreement on the location specification between client and server, one could extend the **get_keys()** function to return the location format expected by the server. The server could even provide just a list of names (text strings) defining particular locations of interest, each name being accepted as a location specification in “within_distance_of”.
- “within_area_of(*area*)” – to select data measured within a defined area. An area can be specified as appropriate, for example as a polygon on a two-dimensional surface. Again, instead of requiring a given agreement on the area specification between client and server, one could consider to extend the **get_keys()** function to return a list of names (text strings) defining particular areas of interest, each name being accepted as an area specification in “within_area_of”.

(c) constraints regarding the measurement value

Conditions related to the measured value itself can be defined, e.g., as follows

- “value_within(-0.5, 1)” – to select measurement data with values between -0.5 and 1.
- “value_is(-0.5)” – to select measurement data with a value of -0.5.

Note that these constraints apply to all attributes requested in **get_data()**. Therefore, if multiple attributes are requested in conjunction with value constraints, the attributes should probably be of the same type.

(d) constraints with respect to another attribute

Sometimes it is necessary to select data depending on the state of a different attribute. For example, the number of bicycles in use will probably depend on the weather. To facilitate evaluations in this respect, the following conditions are suggested:

- “corresponding_attribute(*identifier*, *mode*, *min*, *max*)” – the basic condition. All measurements of the attribute specified by the *identifier* are processed according to a given *mode* taking one of the following values:
 - “mean”, the mean of all measurements is considered,
 - “median”, the median of all measurements is considered,
 - “exists”, to verify if at least one measurement fulfils the condition,
 - “all”, to verify if all measurements fulfil the condition,

- “closest_in_time”, only the measurement with the smallest absolute time difference to the measurement of the primary attribute is considered,
- “closest_in_space”, only the measurement with the smallest spatial distance to the measurement of the primary attribute is considered. The condition is fulfilled if the considered value is between *min* and *max*, i.e. $min \leq \text{value} \leq max$. If there is no measurement, the condition is not fulfilled.
- “corresponding_temporal_attribute(*identifier, mode, min, max, t_min, t_max*)” – extends the basic condition with a time constraint. Each measurement of the corresponding attribute is considered only if the difference in time between the measurement of the primary attribute and the measurement of the corresponding attribute lies within a certain range [*t_min, t_max*]. Each parameter, *t_min* and *t_max*, denotes a time difference in seconds, which can also be a negative value.
- “corresponding_spatial_attribute(*identifier, mode, min, max, d_min, d_max*)” – extends the basic condition with a spatial constraint. Each measurement of the corresponding attribute is considered only if the spatial distance between the measurement of the primary attribute and the measurement of the corresponding attribute lies within a certain range [*d_min, d_max*]. Each parameter, *d_min* and *d_max*, denotes a distance in meters.
- “corresponding_spatiotemporal_attribute(*identifier, mode, min, max, d_min, d_max, t_min, t_max*)” – extends the basic condition with a spatial and a temporal constraint. Each measurement of the corresponding attribute is considered only if (a) the spatial distance between the measurement of the primary attribute and the measurement of the corresponding attribute lies within a certain range [*d_min, d_max*], and (b) the difference in measurement time lies within a certain range [*t_min, t_max*]. The parameters *d_min* and *d_max* denote a distance in meters and the parameters *t_min* and *t_max* denote a time difference in seconds.

5.6.1. A simple example using lists of conditions

The *list_of_conditions1* and *list_of_conditions2* are of the same type, but will usually contain different lists of conditions. As an example, *list_of_conditions1* could possibly contain

[“day_of_week(Mon)”, “day_of_week(Tue)”, “day_of_week(Wed)”, “day_of_week(Thu)”, “day_of_week(Fri)”]

and *list_of_conditions2* might contain

[“time_of_day(10:00,12:00)”, “time_of_day(12:00,14:00)”, “time_of_day(14:00,16:00)”, “time_of_day(16:00,18:00)”],

resulting in a table with 5×4 entries of aggregated data (for each attribute specified in *list_of_attribute_identifiers* and each aggregation function specified in *list_of_functions*):

Table 16: Example result of a query

ATTRIBUTE	10:00 – 12:00	12:00 – 14:00	14:00 – 16:00	16:00 – 18:00
Mon	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...
Tue	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...
Wed	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...
Thu	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...
Fri	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...	mean(matching data), median(matching data), ...

The *list_of_conditions0* contains a list of global constraints that apply to the overall table in addition to the constraints previously defined for the columns and rows of the table. As an example, this could be [“year(2001,2010)”, “month_of_year(1)”] to get some statistics averaged over all Januaries from 2001 till 2010.

array_of_scalars:

The *array_of_scalars* returned by **get_data()** is a four-dimensional array of floating point numbers. The array is of the size $k \times l \times m \times n$ corresponding to

- k aggregation functions specified in *list_of_functions*,
- l scalar attributes specified in *list_of_attribute_identifiers*,
- m conditions specified in *list_of_conditions1*,
- n conditions specified in *list_of_conditions2*.

Each element in the array contains the result of the corresponding aggregation function applied to those measurements of the respective attribute that match the corresponding conditions:

$$single_element = array_of_scalars[g][h][i][j],$$

in which g refers to the g th function in *list_of_functions*, h refers to the h th attribute in *list_of_attribute_identifiers*, i refers to the i th condition in *list_of_conditions1*, and j refers to the j th condition in *list_of_conditions2*.

For example, with a single call to **get_data()** one could get the mean and standard deviation (*list_of_functions*) of the number of available bicycles, individually for a number of bike sharing stations (*list_of_attribute_identifiers*), separately for each hour of the day (*list_of_conditions1*) and each day of the week (*list_of_conditions2*). These statistics can be limited, for example, to a certain time span, like “all Januaries from 2001 till 2010” (*list_of_conditions0*).

Interpretation of the results

It is important to note that if, for example, the mean is calculated over all measurements that fulfil all the given time constraints for a particular table entry, the result is indeed just the *mean of the measurements* in that time span, which is generally not identical to the average value of the attribute over the course of this time span (which is often the desired quantity). The two quantities are only identical if the measurements were equally spaced in time (with a fixed measurement interval d) and the first measurement coincides with the start of the time span and last measurement was made at time $t-d$ before the end of the time span, t , and changes to the attribute’s value appear only at the times of measurements.

Since all these conditions can usually not be fulfilled, one should be aware that the *mean of the measurements* is generally just an *approximation* to the average value of the attribute in the time span. Having many equally spaced measurements in the respective time span usually helps to get a good approximation.

6. EMISSION CONTROL PANEL TECHNOLOGIES AND FRAMEWORKS

This section lists all technologies and frameworks that are used within the MMECP. Some of the technologies were described already in D4.1 [16] and are thus only listed here without a further description.

6.1.1. *Intelligent Test Evaluation Framework*

The Intelligent Test Evaluation Framework (ITEF) will be used in the second iteration for the visualization of the BER pilot simulation results. It is already described in D4.1.

6.1.2. *VsimRTI*

VsimRTI will be used in the second iteration for the simulation of the BER pilot simulation. It is already described in D4.1.

It is developed by Fraunhofer FOKUS.

6.1.3. *Cube*

Cube will be used in the second iteration for the ROV pilot simulation. It is already described in D4.1.

It is not licensed under an open source license, all rights reserved to CitiLabs². CAIRE owns a licence of it and will use Cube for the simulation in Rovereto.

6.1.4. *Carneades*

Carneades is an argumentation system, which is developed by Fraunhofer FOKUS, see [18]. It is used as a main component for forecasting tasks in the MMECP.

It is licensed under Mozilla Public License 2.0³.

6.1.5. *Java*

For the implementation of the MMECP Java is used. Java a cross platform programming language and provides many useful frameworks to develop a complex application like the MMECP. The following sub-sections list the Java frameworks or software that are used by the MMECP.

6.1.5.1. Google Web Toolkit

Google Web Toolkit (GWT) is a development framework from Google. Its purpose is to abstract JavaScript from the implementation. This enables the developer to write her

² <http://www.citilabs.com/products/cube>

³ <http://mozilla.org/MPL/2.0/>

application in Java, GWT compiles this to optimized JavaScript that is able to run on all browsers. In MMECP GWT is used for the implementation of the PanelUI.

GWT is licensed under Apache License Version 2.0.

6.1.5.2. JBoss RESTeasy

JBoss RESTeasy is a Java framework that supports developers with libraries to implement RESTful clients and RESTful web services, see [19]. It is based on the standard Java API JAX-RS and a fully certified implementation of it. In the MMECP the JBoss RESTeasy framework is used to build the MMECP API as a RESTful web service and for the DataAggregator as a RESTful client.

It is licensed under Apache License, Version 2.0⁴.

6.1.5.3. Apache Tomcat

Apache Tomcat is a servlet container, see [20]. It is an implementation of the Java Servlet and JavaServer Pages technology. It is used as a runtime environment for the MMECP.

It is licensed under Apache License, Version 2.0.

6.1.6. *OpenLayers*

OpenLayers is a framework that provides a common API to use several web-based maps in one implementation, see [21]. It is used for the MMECP map view. Thus STREETLIFE cannot expect that every city agree on the usage of e.g. Google Maps or OpenStreetMap, OpenLayers is a nice framework to be adaptable. So far Google Maps and OpenStreetMap are used in the MMECP, others are possible as well.

It is licensed under BSD 2-Clause License⁵.

6.1.7. *ATOM*

ATOM is an umbrella term that describes two standards. The Atom Syndication Format (ASF) and the Atom Publishing Protocol (AtomPub or APP). ASF is a standard for cross platform information exchange via HTTP, see [17]. XML is used for serialization. AtomPub is a protocol to publish ASF, see [22]. It follows the RESTful paradigm. The MMECP uses ASF as a format to provide notifications and guidance. AtomPub is used as protocol to give the notifications and guidance.

6.1.8. *RapidMiner*

RapidMiner is software tool for machine learning and data mining, see [23]. It is already described in D4.1

⁴ <http://www.apache.org/licenses/LICENSE-2.0.html>

⁵ <https://tldrlegal.com/license/bsd-2-clause-license-%28freebsd%29>

The community edition of RapidMiner is licensed under GNU Affero General Public License (AGPL).

6.1.9. R

R is a free software programming language and software environment for statistical computing, data analysis, and statistical graphics, see [24]. It is available under the GNU General Public License. In STREETLIFE, the R software environment can be used to analyse and visualise mobility data and to compute key performance indicators (KPIs).

It is licensed under GNU General Public License Version 2.0⁶.

6.1.10. Google Maps

Google Maps is a web-based service that provides a detailed map for almost all cities on earth, see [25]. It provides services to search for specific locations and to display the results on a map. The data for this service were collected by Google and all rights to it and the services are belonging to Google. In the MMECP it will be used in the map view.

6.1.11. OpenStreetMap

OpenStreetMap (OSM) is similar like Google Maps. The main different is that OSM is open source licensed under the Open Database License 1.0⁷ (ODbL). OSM provides a web-based service that can be used like Google Maps. The data for OSM is not collected from a company, it is collaborative work of the community, see [26]. Thus many urban areas have more details in e.g. street networks than in Google Maps. On the other hand, some areas are not covered as well as in Google Maps. In the MMECP OSM will be used in the map view, because ROV and TRE pilots use OSM instead of Google Maps.

6.1.12. City Intelligence Platform

The City Intelligence Platform (CIP) is already described in D4.1. From a technological perspective it could have been used as a basis for the MMECP. But the CIP is not open source licensed. Therefore it will be specifically opened for STREETLIFE API to access the database with all mobility data and services and to use pre processing data analysis and computational modules for MMECP in BER.

6.1.13. FIWARE

FIWARE is an open initiative that includes various tools and components, see [27]. It is further described in D3.2., see [28, Ch. 3.2].

⁶ <http://www.r-project.org/>

⁷ <http://opendatacommons.org/licenses/odbl/>

7. CONCLUSION

Deliverable D4.2.1 describes the current status of the MMECP. Section 2 gave an overview of state of the art on city cockpits, which range from interactive transportation information systems to traffic management and adaptive control software. Like in the STREETLIFE approach, these tools are addressing different kind of users, such as citizen and mobility managers. None of the evaluated tools, except Imtech Traffic & Infrastructure, seems to have forecasting or simulation features. It is an on-going task to further look for similar technologies, evaluate them, learn from them and – if advantageous and possible – even adopt them.

Section 3 describes in a more detailed way the pilot scenarios which should be implemented for the MMECP, drawing from Deliverable D6.1 [1]. The fundamental functionality of the MMECP are discussed in this section. For year one the main goal is to solve the issues especially regarding the visualization and organization by calculated KPIs. Therefore many analytical approaches and technologies are needed, which are specified in following chapters.

Section 4 presents the first draft of layout from the MMECP, which will be used by mobility managers operating on the pilots. Most information and KPIs must be calculated out of historical data or also predicted data. These calculations are the biggest task of the MMECP. In general there are several steps necessary to calculate different KPIs successfully and some of them are even dependent of each other. The forecast of the modal split should be evaluated by surveys, which will collect data about the audience of concrete events and their used means of transportation. Also the forecast process has to be optimized, expanded with probabilities and his execution started via an API of the argumentation system.

Section 5 describes the implementation of all the functionalities from MMECP besides the calculation of KPIs. Additionally the complete design of MMECP is shown and all of its components presented. The MMECP architecture is conceptually split in three different parts: Backend, UI and Simulation. The main communication between these parts will take place about an API, which is implemented in the backend, which represents a common interface for all components of MMECP. All the dynamic content, which is visualized in the UI, and the requests of mobility managers over the MMECP are provided and processed with the API as well.. The goal of this architecture is to provide a generic platform to facilitate the general and customizable usage of the MMECP across the STREETLIFE pilots and beyond.

Section 6 collects all used technologies and frameworks at the first year of MMECP.

APPENDIX A: LITERATURE

- [1] G. Valetto, D. Frigeri, R. Kelpin, T. Pezzato, F. Thiemer, M. Vuorio, and M. Wollina, “D6.1 Specification of city pilots for the first STREETLIFE operation and evaluation.” Feb-2014.
- [2] M. Bahr, S. Cuno, L. Farid, F. Thiemer, G. Valetto, V. Brandstetter, B. Obst, S. Schaffer, M. Vuorio, and B. Dittwald, “D2.2.1 – STREETLIFE Blueprint Architecture, Security Architecture, and site-specific architectures (initial).” Oct-2014.
- [3] W. Kaczorowski, *Die smarte Stadt: den digitalen Wandel intelligent gestalten ; Handlungsfelder, Herausforderungen, Strategien*, vol. 1. Boorberg, 2014.
- [4] “Homepage - Imtech Traffic & Infra,” 22-Aug-2014. [Online]. Available: <http://imtech.com/EN/traffic-infra>. [Accessed: 22-Aug-2014].
- [5] *Ottawa Transport Management - Interactive Traffic Map*.
- [6] “Interactive Traffic Map,” 22-Aug-2014. [Online]. Available: <http://traffic.ottawa.ca/map/>. [Accessed: 22-Aug-2014].
- [7] “Econolite :: Centracs,” 22-Aug-2014. [Online]. Available: <http://egiweb.concrete5.arvixe.com/index.php/products/systems/centracs/>. [Accessed: 22-Aug-2014].
- [8] “Tempelhofer Freiheit,” 05-Oct-2014. [Online]. Available: <http://www.thf-berlin.de/en/>. [Accessed: 05-Oct-2014].
- [9] Dr.-Ing. Frank Ließke, R. Dr.-Ing Wittwer, S. Dipl.-Ing. Hubrich, and G.-A. Prof. Dr.-Ing. Ahrens, “Endbericht zur Verkehrserhebung ‘Mobilität in Städten - SrV 2008’ in Berlin,” Aug. 2009.
- [10] “Policy Impact Project,” 22-Sep-2014. [Online]. Available: <http://www.policy-impact.eu/>. [Accessed: 22-Sep-2014].
- [11] “PMML 4.2 - General Structure,” 22-Sep-2014. [Online]. Available: <http://www.dmg.org/v4-2-1/GeneralStructure.html>. [Accessed: 22-Sep-2014].
- [12] “Simulation of Vehicle-2-X Communication - DCAITI - TU Berlin,” 28-Sep-2014. [Online]. Available: <https://www.dcaiti.tu-berlin.de/research/simulation/>. [Accessed: 28-Sep-2014].
- [13] A. Nurminen, G. Valetto, S. Schaffer, and A. Marconi, “D5.2.1 – End-user applications techniques and tools.” Oct-2014.
- [14] “DLR - Institute of Transport Research - TAPAS: Model for future demand in passenger transport,” 05-Oct-2014. [Online]. Available: http://www.dlr.de/vf/en/desktopdefault.aspx/tabid-2974/1445_read-29381/. [Accessed: 05-Oct-2014].
- [15] “Software for transportation planning, modeling, traffic engineering, GIS and urban planning | Citilabs,” 05-Oct-2014. [Online]. Available: <http://www.citilabs.com/>. [Accessed: 05-Oct-2014].
- [16] B. Dittwald, B. Schünemann, B. Obst, V. Brandstetter, D. Frigeri, P. Gilka, and J. Kohlmorgen, “D4.1– Requirements analysis, mechanism and technology selection, and control panel specification.” Feb-2014.
- [17] “Atom Enabled,” 23-Sep-2014. [Online]. Available: <http://atomenabled.org/>. [Accessed: 23-Sep-2014].
- [18] “Carneades · tools for argument (re)construction, evaluation, mapping and interchange,” 28-Sep-2014. [Online]. Available: <http://carneades.github.io/carneades/>. [Accessed: 28-Sep-2014].
- [19] “REStEasy - JBoss Community,” 24-Sep-2014. [Online]. Available: <http://reステasy.jboss.org/>. [Accessed: 24-Sep-2014].

- [20] “Apache Tomcat - Welcome!,” 24-Sep-2014. [Online]. Available: <http://tomcat.apache.org/>. [Accessed: 24-Sep-2014].
- [21] “OpenLayers 3 - Welcome,” 24-Sep-2014. [Online]. Available: <http://openlayers.org/>. [Accessed: 24-Sep-2014].
- [22] “Atom Publishing Protocol,” 24-Sep-2014. [Online]. Available: <http://atompub.org/>. [Accessed: 24-Sep-2014].
- [23] *rapidminer*. rapidminer, inc.
- [24] “The R Project for Statistical Computing,” 24-Sep-2014. [Online]. Available: <http://www.r-project.org/>. [Accessed: 24-Sep-2014].
- [25] “Google Maps,” 24-Sep-2014. [Online]. Available: <https://www.google.de/maps/@51.1758057,10.4541194,6z>. [Accessed: 24-Sep-2014].
- [26] “OpenStreetMap,” 24-Sep-2014. [Online]. Available: <http://www.openstreetmap.org/#map=5/51.500/-0.100>. [Accessed: 24-Sep-2014].
- [27] “FIWARE » Open APIS for Open Minds,” 28-Sep-2014. [Online]. Available: <http://www.fi-ware.org/>. [Accessed: 28-Sep-2014].
- [28] M. Mitrevska, G. Valetto, M. Vuorio, and A. Nurminen, “D3.2.1 – Mobility Data Integration Models and Techniques (initial).” Oct-2014.