Project No. 608930

FP7-SMARTCITIES-2013

OPtimising Hybrid Energy grids

for smart citieS

WP3: Monitoring and System Analysis

Deliverable D3.1.1

System Design of parallel connection of M2M gateways to modern Smart City equipment

Apostolos Papageorgiou[1], Christer Åhlund[2], Karan Mitra[2], Saguna Saguna[2]

[1]NEC Europe Ltd., [2]LTU

Version:	1.0
Due Date:	31 Aug 2014
Submission Date:	25 Aug 2014
Lead Coordinator:	NEC

Copyright

Consisting of

Coordinator	WIP – Renewable Energies (WIP)	Germany
Participants	Hochschule Ulm (HSU)	Germany
	Stadtwerke Ulm GmbH (SWU)	Germany
	Technische Universität Wien (TUW-EEG)	Austria
	Austrian Institute of Technology (AIT)	Austria
	Deutsches Zentrum für Luft- und Raumfahrt EV (DLR)	Germany
	NEC Europe Ltd. (NEC)	UK
	Lulea Tekniska Universitet (LTU)	Sweden
	Skelleftea Kraft (SKR)	Sweden

Deliverable Description

Abstract: This deliverable D3.1.1 derives a reference M2M platform design, identifies and concretizes project-related challenges, and designs four platform extensions (self-configuration, consumer behavior analysis, data filtering, and M2M cloud systems) for tackling the identified challenges, while it also examines further ICT tasks for supporting other WPs and discusses the future usage of the developed extensions for the enhancement the solutions of other WPs, e.g., energy control and visualization.

Key Words: ICT, smart cities, hybrid energy grid, energy saving, demonstration, smart grid, energy control, monitoring, sensor communication

Document History

Version	Date	Authors	Description
0.1	2014-05-08	NEC	Draft outline
0.2	2014-05-09	NEC	Added Section 2 (Horizontal M2M systems for Smart Cities) and some draft contents for Section 3
0.3	2014-06-06	NEC, LTU	Completed 3.2 and concretized the draft of 3.1 Finalized the outline and provided guidelines for describing the M2M platform extensions (Section 4).
0.4	2014-07-08	NEC	Added self-configuration platform extension (4.1 + Appendix A) and extended 5.2 (Smart City integration)
0.5	2014-07-18	NEC	Restructured Section 4, added intro & conclusions, and consolidated the entire document
0.5a	2014-07-29	LTU	Added content to section 4 and consolidating/polishing across document
0.6	2014-08-01	NEC	Polishing for internal review
0.7	2014-08-25	NEC	Final version
1.0	2014-08-26	WIP	Final revision

Dissemination Level

Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive Summary

The OrPHEuS project elaborates a Hybrid Energy Network Control System for Smart Cities implementing novel cooperative local grid and inter-grid control strategies for the optimal interactions between multiple energy grids by enabling simultaneous optimization for individual response requirements, energy efficiencies and energy savings as well as coupled operational, economic and social impacts. Starting from existing system setups in two cities, enhanced operational scenarios are demonstrated for today's market setup, as well as for future market visions.

The Deliverable D3.1.1 (Task 3.1) provides a reference platform design, identifies and concretizes project-related challenges, and tackled them through design extensions. Along this path, the works performed in the context of D3.1.1 should help (in the future) to enhance domain-specific OrPHEuS solutions, mainly by evaluating the impact of M2M platform extensions to control strategies and by examining how the platform operation can be harmoniously integrated with OrPHEuS visualization tools and demo site data aggregation mechanisms connecting to the control strategy simulator. Thus, Deliverable D3.1.1 analyzes state-of-the-art Machine-to-Machine (M2M) architectures, derives their common denominator as the OrPHEuS reference M2M platform design, identifies M2M platform research challenges, and designs four platform extensions, namely for supporting self-configuration, energy consumer behavior analysis, data filtering (under development), and a cloud solution. Finally, it discusses the (future) usage or evaluation of these extensions in conjunction with other WPs.

Administrative Overview

Task Description

Task 3.1 is investigating "*approaches for monitoring, managing, filtering, and disseminating data with M2M (Machine-to-Machine) systems*". The first phase of Task 3.1 (described in the current Deliverable) focuses on "*system design*", "*connection of M2M gateways to modern Smart City equipment*" (incl. devices used for energy systems monitoring), and other M2M platform issues which are important for the future steps of the Task (efficient data handling mechanisms, case-specific M2M extensions etc.).

Relation to the Scientific and Technological Objectives

This task is related to STO2, addressing the targets of

a) "*Integration of existing independent energy grid ICT systems as subsystems for future Smart City Operation Centers for the energy domain*", by focusing mainly on the:
b) "*Utilization, development of extensions and customization of Machine-to-Machine(M2M) infrastructure*"

With regard to a), progress has been achieved by performing a study that derives a reference architecture of modern M2M systems that form the backbone of Smart Cities, maps the energy grid ICT systems of OrPHEuS to it, and systematically analyzes the challenges and the potential for enhancements through platform extensions.

With regard to b), progress has been achieved by designing and developing platform extensions (see previous paragraph) for efficient self-configuration, and data handling (data processing in conjunction with storage) in the M2M platform, but also for further (case- or project-specific) goals. Such extensions pave the way towards the achievement of the promised "advances beyond the state-of-the-art" for STO2, such as "*Intelligent horizontal M2M data management architecture for Smart City*" and "*Technologies for lightweight M2M data filtering solutions*".

The current Deliverable is directly linked with the following Performance Indicator:

No.	Objective/expected result	Indicator name	STO	Deliverable	MS	Expected Progress		
						Year 1	Year 2	Year 3
3	Definition of M2M system for data monitoring and management in the demosites	Number of systems for 'Data monitoring'	STO2	D3.1.1	MS1	Due: M12 Draft: M9		

Relations to activities in the Project

This Task is using knowledge about OrPHEuS ICT devices obtained from WP6 and is directly linked to Task 5.4.1 (because of the common context and the interactions between "demo site data aggregation", "visualization tools", "Smart City operation"), as well as, of course, to future Tasks of WP3 on the topics of M2M data handling and data filtering. Finally, the impact of M2M data handling techniques on control strategies (e.g., effect of data filtering / missing data) will have to be evaluated in coordination with WP5.

Terminologies

Abbreviations

3GPP	3rd Generation Partnership Project
API	Application Programming Interface
BADA	Balanced Achievement Degree Algorithm
CCA	Configuration-Comparing Algorithm
CHP	Combined Heat Power Plant
CPU	Central Processing Unit
DH	District Heating
DHS	District Heating System
DoW	Description of Work
DTA	Degrees of Target Achievement
ECCA	Extremal Configuration-Comparing Algorithm
EDS	Energy Distribution Systems
EG	Electricity Grid
ETSI	European Telecommunications Standards Institute
FFNN	Feed Forward Neural Networks
GW	Gateway
HTML	HyperText Markup Language
ICT	Information and Communications Technology
IoT	Internet of Things
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Medium Access Control
ML	Machine learning
MLR	Multiple Linear Regression
MS	Milestone
MTC	Machine-Type Communications
MVE	Most Valuable Entry
PV	Photovoltaic
QoS	Quality of Service
SCOC	Smart City Operation Centers
SDO	Standards Development Organization
SIM	Subscriber Identity Module
SINTEF	Stiftelsen for INdustriell og TEknisk Forskning
STO	Scientific & Technological Objective
SVM	Support Vector Machines
TCP	Transmission Control Protocol
WP	Work Package
WRA	Weight Ranking Algorithm

Table of Contents

1 Introduction

Task 3.1 focuses on design aspects and research advancements for Smart City Machine-to-Machine (M2M) platforms, especially for (but not limited to) the usage of such platforms as enablers of efficient control mechanisms for hybrid energy grids.

M2M platforms can support different industrial verticals which all together comprise a Smart City or a similar concept which is even bigger than that. The energy grid is one of these industrial verticals and thus its ICT/M2M system should be designed and examined as part of a bigger infrastructure that might serve other domains, as well. Some of the research challenges that appear with regard to such a platform might be related to aspects that are specific to the energy domain, while others might be related to the mentioned transition of merging the ICT subsystems into “one M2M platform for all Smart City domains.”

No matter where the challenges exactly appear, OrPHEuS needs to have a *reference platform design* upon which *project-related platform challenges are tackled through extensions*, which in the end should *help to enhance or evaluate solutions developed for the specific Use Cases of OrPHEuS*, namely related to control or visualization of Hybrid Energy Grids.

On the pursuit of the aforementioned goals, this Deliverable had to analyze state-of-the-art M2M platforms, provide reference platform designs, identify concrete platform-related challenges, develop extensions for tackling these challenges, and examine the link between technical ICT issues and other project tasks (e.g., tools visualization, control strategies).

Thus, the outline of the Deliverable is as follows:

Section 2 provides the background and the analysis of the state of the art with regard to M2M platform design in research, standards, and industry, leading to the derivation of a common denominator, which will be the reference platform design for OrPHEuS extensions. Having this reference point, Section 3 examines the respective research challenges based on a related book and explains which of them are addressed in OrPHEuS, how and why. Section 4 looks further into the details of the platform extensions that have been developed in order to tackle the identified challenges, namely extensions for self-configuration (4.1), energy consumer behavior analysis (4.2), and data filtering & network-edge computing (4.3), while Section 5 provides some information about prototypes and further supportive tasks that are being performed in conjunction with other Work Packages. Finally, Section 6 summarizes the outcomes related to this Deliverable, providing also a good overview of future steps and links to other Work Packages.

2 Horizontal M2M systems for Smart Cities

The energy network ICT infrastructure is one of the components of a Smart City ICT infrastructure. Other components might be needed in order to support different domains (so-called "verticals") of a Smart City, e.g., safety systems, traffic and transport, logistics, e-health, and assisted living. All these infrastructures have in common that they are powered by their underlying Machine-to-Machine (M2M) systems.

M2M systems are the systems in which various devices, sensors, and computing systems communicate with the physical environment and with each other without human intervention, in order to offer easy, homogeneous, and efficient remote access to machines or other connected data sources. M2M protocols and architectures are hot issues for industry, standards, and research ([6] , [10] , [13]). Most M2M systems are structured in multiple layers and each layer has a different role. For example, the bottom layer is typically responsible for gathering data from various sources (i.e., sensors, devices, home appliances, etc.) by using different M2M access technologies (e.g., Zigbee[1], KNX[2], Bluetooth[3], etc.). The data is delivered to an upper layer for further data processing, which, in turn, exposes them to user applications via uniform Application Programming Interfaces (API).

Since there are many verticals, current M2M systems are fragmented, i.e., each application or (in the best case) each domain (e.g., energy) has its own M2M system. This means that in order to launch a new M2M application, it is normally necessary to deploy new devices, gateways, and databases, which are dedicated to this application. However, because of the huge cost reduction that can be achieved by unifying the underlying M2M systems of different verticals, but also because of interoperability issues, current research and latest trends lead to a transition towards "horizontal" solutions ([6] , [1]). Figure 1 and Figure 2 illustrate the difference between the traditional approach of vertical systems and the modern paradigm of horizontal integration on a very high level.

[1] http://www.zigbee.org/
[2] http://www.knx.org/
[3] http://www.bluetooth.org/

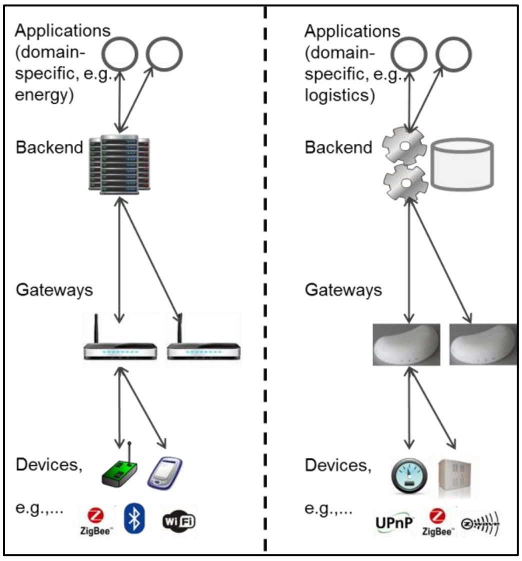


Figure 1. Illustration of M2M systems created based on the vertical paradigm

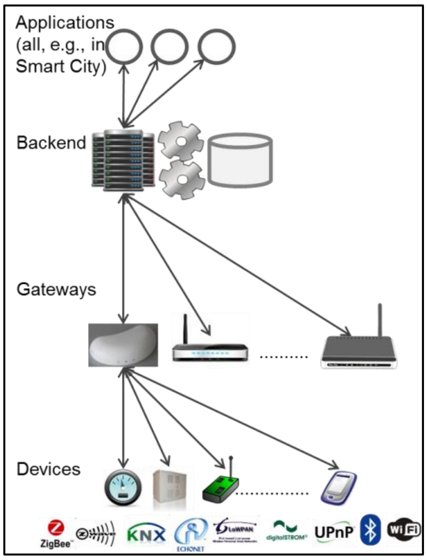


Figure 2. Illustration of M2M systems created based on the horizontal paradigm

Thus, in accordance to this transition, standardization bodies, platform development companies, and researchers are designing reference architectures for horizontal M2M systems. More or less, all these parties have implicitly agreed about the main elements and the targeted functionality of these systems. The following sections (2.1, 2.2, and 2.3) summarize the status quo and give examples of reference architectures from the mentioned disciplines (standardization, industry and research) while in section 2.4 we draw a very high-level view as a "common denominator" of these reference architectures. The purpose is to map the OrPHEuS ICT system to this architecture, in order to avoid re-inventing the wheel, and to identify and discuss platform-related research tasks on this basis, i.e., as extensions of state-of-the-art horizontal M2M systems.

2.1 Reference architectures from standardization bodies

The emergence of M2M encouraged many standardization bodies to create specifications for various aspects of M2M platforms. For example:

- ETSI M2M[4] has defined M2M reference architectures and has specified hierarchical protocols for accessing M2M resources.
- The Broadband Forum[5] has been working on specifications for the management of devices that play an important role inside an M2M system, such as gateway devices.
- 3GPP MTC (Machine-Type Communications)[6] has been working on specifications related to the use of cellular devices and cellular networks in M2M solutions.
- oneM2M[7] was launched in 2012 in an attempt to unify the fragmented landscape of M2M standardization, aiming to "develop technical specifications which address the need for a

[4] http://www.etsi.org/technologies-clusters/technologies/m2m
[5] http://www.broadband-forum.org/
[6] http://www.3gpp.org/DynaReport/22368.htm
[7] http://www.onem2m.org/

common M2M Service Layer that can be readily embedded within various hardware and software".

It goes without saying that all these Standards Development Organizations (SDOs) are using reference architectures for the entire M2M system, each one for its own purposes. However, these reference architectures look very similar. As an example, the reference architecture of ETSI M2M (many concepts of which are being adopted by oneM2M) is shown in Figure 3.

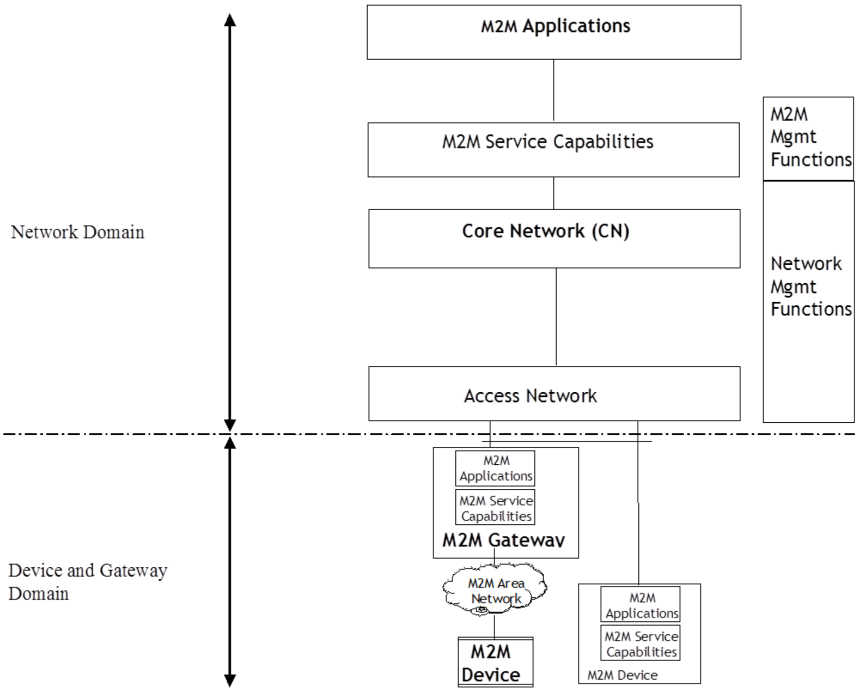


Figure 3. ETSI M2M standard reference architecture [1]

2.2 Reference architectures from the industry

Although most M2M platform providers do not reveal many technical details of their solutions, a few architectures of industrial M2M solutions have been published as well. Figure 4 is an overview of NEC's solution, *Connexive*, as described in [6] and [8] .

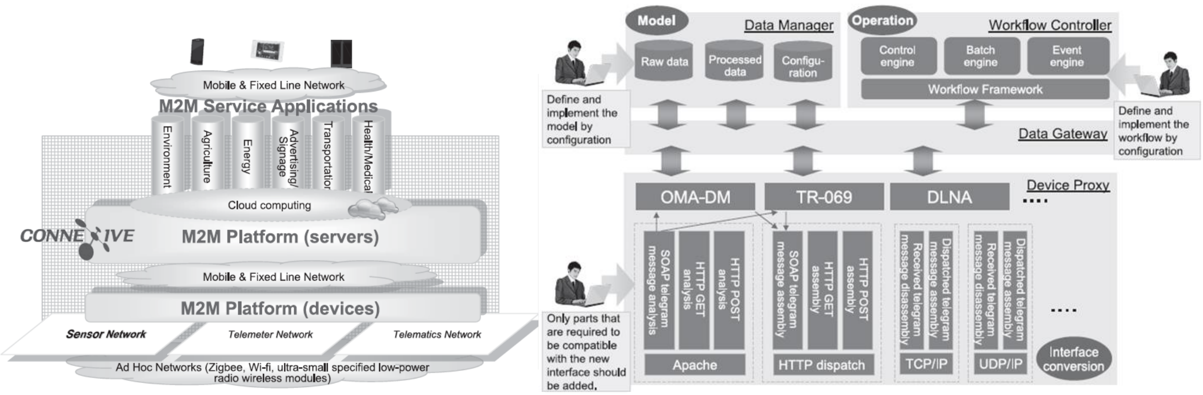


Figure 4. Overview of NEC's M2M solution "Connexive" (sources: [8] [6])

The left part is a more high-level view, including the applications and business perspectives, while the right part zooms into the platform & devices space, showing how devices of different protocols are controlled by device proxies and data gateways, which forward the data to the upper layers for further processing and exposure, as we explained earlier in the introduction of the current chapter.

2.3 Reference architectures from research

A similar example from the research discipline can be found in the OpenMTC Framework of the Fraunhofer Institute (cf. Figure 5), which aims "to provide a standard compliant middleware platform for M2M-oriented applications and services" [12] and defines the role of sensors, actuators, area networks, M2M gateways, M2M servers etc. The goal is again to support "horizontal" M2M system usage and the similarities to the previously presented reference architectures are obvious.

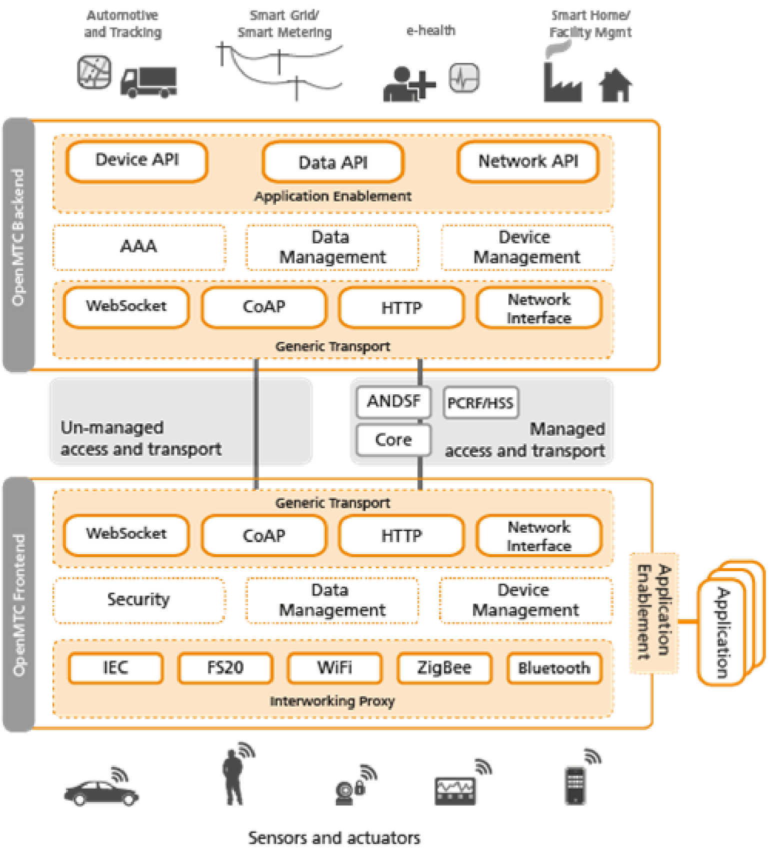


Figure 5. Architecture of the OpenMTC framework[8]

2.4 High-level overview of horizontal M2M system design

Having recognized the very big similarities between the previously shown reference architectures, it becomes obvious that we should not re-invent the wheel in order to design a high-level architecture in the context of OrPHEuS. Instead, we will present a "common denominator" of the discussed reference architectures (current section), we will map the basic OrPHEuS ICT system to it, and we will

[8] From http://www.open-mtc.org/openmtc_platform/architecture/index.html (last accessed May 2014)

design and develop our scientific (Section 4) and further technical (Section 5) contributions as extensions of this basic system overview.

Figure 6 shows the mentioned common denominator, whereby *M2M Devices* can be any kind of sensor devices, smart meters, other SIM-card equipped devices, cameras, home devices, but of course also computing devices like smartphones, if they are executing an automated task, which usually captures (or generates) data. The *M2M Frontend* refers to the set of proxies or gateways that are usually close to the M2M devices and connect them to the backbone network (for example, Home Gateways or similar devices can often act as M2M gateways), while the *M2M Backend* is a set of remote (often Cloud-based) servers, along with the corresponding databases, software modules for intelligent Big Data processing etc. The "uppermost" of the mentioned software modules offer an API to various *Applications*, which can be developed in a uniform way independently of the data and the technologies that are required from the lower layers. The way the information is captured, collected, organized, processed, and forwarded is dictated by the gateway configuration. This configuration needs to be maintained at a state that best covers the needs and serves the goals of the entire system.

It must be noted that the elements presented in Figure 6 are more or less part of all previously shown architectures. However, this should be seen neither as an attempt to unify them nor as "yet another suggested reference architecture". It is simply "what we can adopt" from the state of the art as a starting (and reference) point for our platform-related works in OrPHEuS. Figure 7 shows examples of types of M2M devices and modules, which are used in OrPHEuS demo sites and/or OrPHEuS prototypes at the different layers of this reference architecture.

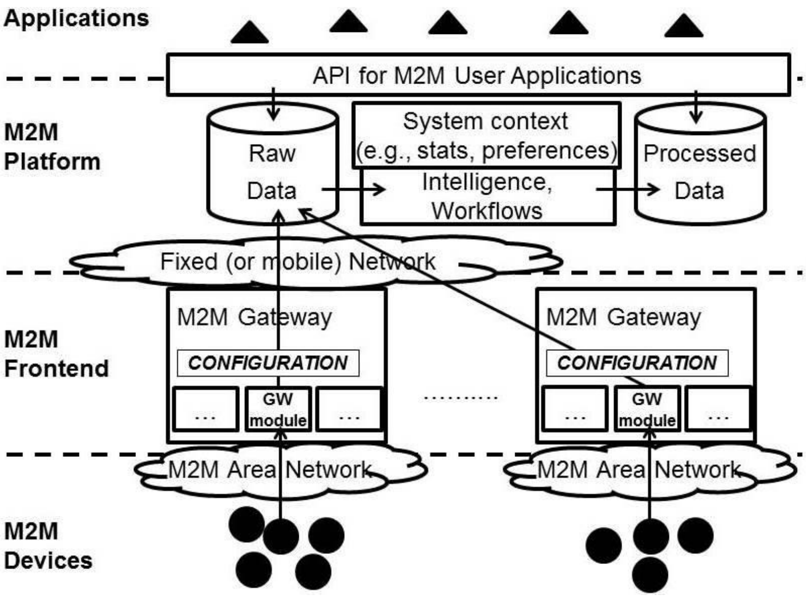


Figure 6. Generic horizontal M2M system overview for the purpose of mapping OrPHEuS ICT to it.

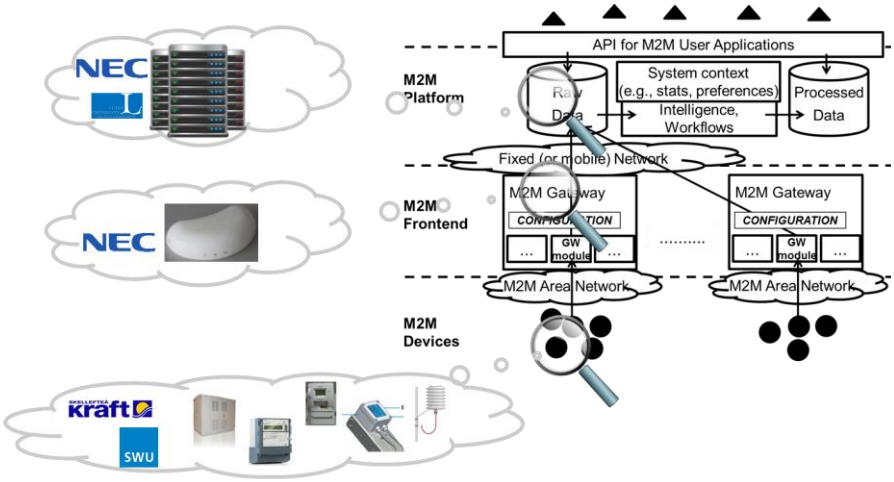


Figure 7. Examples of types of M2M devices and modules which are relevant to the Use Cases of OrPHEuS

3 Identification and discussion of platform-related research tasks

Having a more concrete view of the reference OrPHEuS ICT platform, it is now possible to analyze current research challenges related to this kind of platforms and discuss in which directions research can be conducted in order to offer added value to OrPHEuS and its "Hybrid Energy Control" and "Smart City" Use Cases.

Combining our experiences in M2M platform research and development with consortium-wide discussions and Use Case analyses, we have identified two possible classes of works in the context of our M2M platform, namely our research tasks for bringing the M2M platform beyond the state of the art and possible technical tasks for supporting the research of other tasks or work packages. Subsections 3.1 and 3.2 examine these two classes, respectively, providing decisions and a roadmap for research within Task 3.1.

3.1 Analysis of research tasks for enhancing M2M platforms

Although there are various sources in literature, we will go through the current research challenges of M2M platforms based on a relevant published study [11] , co-edited by SINTEF[9] and the European Commission, which includes elaborate suggestions on research topics for platforms such as ours.

Among others, the referenced work lists "Recommendations on Research Topics" for the Internet of Things (IoT) and Smart Cities, including aspects related to scalable sensor communication, data filtering and platform configuration, which belong to our main concerns (as also described in the project DoW). Table 1 comments on these recommendations, examining their OrPHEuS-relevance, and summarizing the respective tasks planned (or not) in the scope of the project. The book provides recommendations on a list of topics, which we list here, explaining if, why, and where they are addressed by the M2M platform extensions of Task 3.1.

Table 1. Analysis of M2M platform challenges and how they are addressed in Task 3.1

Topic	Challenges and OrPHEuS relevance	Task 3.1 roadmap
Applications *[11] , §2.12.1*	Task 3.1 targets in this respect data gathering of application information at the consumer side to be communicated, processed and stored as input to the control loop (e.g. using used machine learning), OrPHEuS is perfectly addressing the recommendation "*Set up interdisciplinary projects for smart energy, grid and mobility*", with Task 3.1 leading the IoT and Smart City focused issues.	See "4.2 M2M platform extension for consumer behavior analysis" This will firstly apply to the district heating with long delays in the control loop and later on, possibly to be included for the electricity grid as well.
Autonomic and Self-aware IoT	Recommendations such as "*Self-awareness from the design to deployment*" and "*characterization of*	See "4.1 M2M platform extension for efficient

[9] http://www.sintef.no/home/

[11] , §2.12.2	*self-x properties based on real-life Use Cases*" belong to the most important topics when broadly-scoped, inter-domain Smart City platforms are examined (such as in OrPHEuS). Self-configuration is one of the central topics our platform extensions are concerned with.	self-configuration", esp. the developed framework, description formats, and algorithms for self-configuration
Infrastructure *[11] , §2.12.3*	This recommendation involves various challenges, but specifically the recommendation for "*research on symbiosis of networking and IoT-related distributed data processing*" is considered to be very important in OrPHEuS because of the huge amounts of data that could be coming (from heterogeneous sources and various domains) into a Smart City platform and because of the multi-level hierarchy that exists in networks such as those involved in architecture like our reference architecture in Figure 6.	See "4.3 M2M platform extension for data filtering and fog computing", esp. the mechanisms suggested for exploiting fog computing in order to store and process data close to producers and consumers
Networks, Communications *[11] , §2.12.4*	Scalability of network communication in M2M based systems like the OrPHEuS infrastructure requires knowledge about network communication performances. This requires monitoring of communications paths in order to decide where to place information for further processing. E.g. wireless links like Zigbee have limited performance and battery powered sensor devices needs to keep the energy consumption as low as possible. Further, when aggregating all sensor information to be communicated in the backbone even high performance links may become congested.	See "4.3 M2M platform extension for data filtering and fog computing".
Processes *[11] , §2.12.5*	"*Modelling*", "*execution*", and "*distribution of processes*" are out of scope, as they are absent from the project work plan and Use Cases.	No concrete platform enhancements of Task 3.1 will address this recommendation
Data Information Management *[11] , §2.12.6*	The respective recommendation states that currently "*no interconnection or distribution of data is permitted, thus limiting the possibilities for parallel or concurrent processing of data and also limiting the scope and domain of data that can be collected by the system*", while "*the volume of data susceptible of being collected and automatically stored in information systems is huge*". Thus, among others, "*data sharing and optimization techniques*" are needed. Task 3.1 has described very similar issues in its description of work and is addressing data sharing and optimization with focus on data filtering (partly achieved through self-configuration) and local data storage and processing.	See "4.1 M2M platform extension for efficient self-configuration" and "4.3 M2M platform extension for data filtering and fog computing"
Security, Device-level energy issues, Interoperability, Standardization,	The mentioned issues, e.g., security and standardization are out of scope for this task and could only be partly addressed by M2M platform extensions. However, companies involved in the	No concrete platform enhancements of Task 3.1 will address these recommendations, as

Legal issues *[11] , §2.12.7-2.12.11*	consortium and in the task, e.g., NEC, are highly active in Standardization Bodies, so that impact of the OrPHEuS work is given. Further, certain legal and security/privacy issues are considered in other WPs (esp. WP6, regarding demo site data), but currently out of scope for M2M platform research.	they are partly handled in other WPs and partly technically irrelevant.

3.2 Analysis of tasks for supporting control strategies and Smart City operation

By analyzing relevant OrPHEuS Use Cases and by discussing the needs of the researchers of other Work Packages, we have compiled a list of possible additional M2M platform-related tasks that can support the research in the direction of hybrid control strategies and their usage in the general context of Smart City Operation Centers.

It is important to note that this analysis is mainly performed in order to identify possible technical obstacles (and the respective enablers). However, resolving such issues would often require demo site upgrades and ICT integration work to an extent that is out of scope of this project. Therefore, the listed tasks are optional tasks and respective technical implementations might be done only on demand, if they fit the scope of the project and the work package. Otherwise, this list is mainly a first step in the direction of providing further domain-specific "Recommendations for ICT M2M Infrastructure for the evolution towards Cooperative Hybrid Energy Grids integrated in Smart Cities Operation Center" (future Deliverable D3.1.3).

Table 2. Possible tasks for supporting research on energy control and Smart City operation and how they are addressed in Task 3.1

Possible task	How it could support OrPHEuS	Task 3.1 roadmap
Demo site data aggregation	The OrPHEuS demo sites will provide various data, which will be used by other WPs. This data will be gathered from different sources and from different formats. It is thinkable that a homogenization and a provision of uniform access could enhance the usability of the data. This task refers to the development of additional data transformers, wrappers, and APIs, as well as to the hosting of the data in uniform databases.	This task is optional and its details are currently driven by a cooperation of WP6 and WP5, esp. Task 5.4, but Task 3.1 might need to provide support (cf. "5.1 Demo site data aggregation").
Integration of OrPHEuS tools in the context of a Smart City Operation Center	The validation of platform extensions of Task 3.1, but also the testing of energy control tools, should ideally be performed in the context of a common Smart City Operation Center. Interdependencies and synergies can be better examined if such modules are integrated and used in a joint context.	This task is optional and is currently driven by a cooperation between Task 3.1 and Task 5.4 towards integrating various OrPHEuS prototypes (e.g., self-configuration solutions, energy control visualization tools, etc.) as modules of a common tool, i.e., in the direction of a Smart

		City Operation Center (cf. “5.2 Integration of OrPHEuS tools in a Smart City operation context”).
Import municipality data	There have been discussions that importing municipality data such as “planned/ongoing constructions”, “building information” etc. could help enhance energy control strategies.	Considerations about this task might lead to concrete recommendations in later steps, but no respective implementations or detailed descriptions are currently planned.
Sensors for “presence detection"	There have been discussions that installing “presence detection” sensors at customer premises in order to enhance “load forecasting” could also help enhance energy control strategies.	Considerations about this task might lead to concrete recommendation in later steps, but no respective implementations or detailed descriptions are currently planned.

4 Research tasks for enhancing the M2M platform

As explained in Table 1 the technical work has mainly focused on designing three extensions that are applicable to standard M2M platforms (see section 2), namely for a) self-configuration, b) user behavior analysis, and c) data filtering and fog computing (i.e., computing close to the network e.g., close to data sources such as sensors or substations). Sections 4.1, 4.2, and 4.3 describe details of the three designed extensions. The self-configuration mechanism of 4.1 is directly related to 4.3 because it includes the intelligence that computes -among others- “suggested data filtering settings”, while 4.2 is independent of them, but it is a platform extension that can provide useful input to control strategies of hybrid energy networks.

Note that while sections 4.1 and 4.2 describe developed solutions, i.e., implemented platform extensions, section 4.3 is at an earlier stage and holds the framework-discussion for what is planned to be the focus of M2M design during the second year of the project, namely the design of data filtering solutions and their evaluation in conjunction with control strategies (WP5), demo site data (WP6), and other interesting Use Cases.

4.1 M2M platform extension for efficient self-configuration

This section provides the self-configuration extension of the M2M platform, which has been identified to be an important research focus in the scope of OrPHEuS (refer also to row 2 of Table 1). First, an overview of the solution as extension of our reference architecture is provided, then related work is briefly explored, and finally the details of the solution and its further usage in OrPHEuS are described.

4.1.1 Overview of the extension

In order to fulfill the goal of self-configuring gateways, which can save a lot of operational costs and achieve meaningful dynamic changes of the system settings, we have designed an extension of the basic architecture of Figure 6, based on the idea of “letting the Gateways re-configure themselves” according to “suggestions” of the backend system. This was conceived because of the heterogeneity of the Gateways and the big differences in their configurable parameters. Due to this heterogeneity, the sophisticated re-configuration decisions of a backend system can only be suggestions, which can be interpreted differently by each Gateway, depending on its technical details.

Figure 8 gives an overview of the developed solution as an extension of our reference architecture, but detailed descriptions and the theory of operation are explained later (section 4.1.3), after a discussion of the scientific and technical background and related work (section 4.1.2).

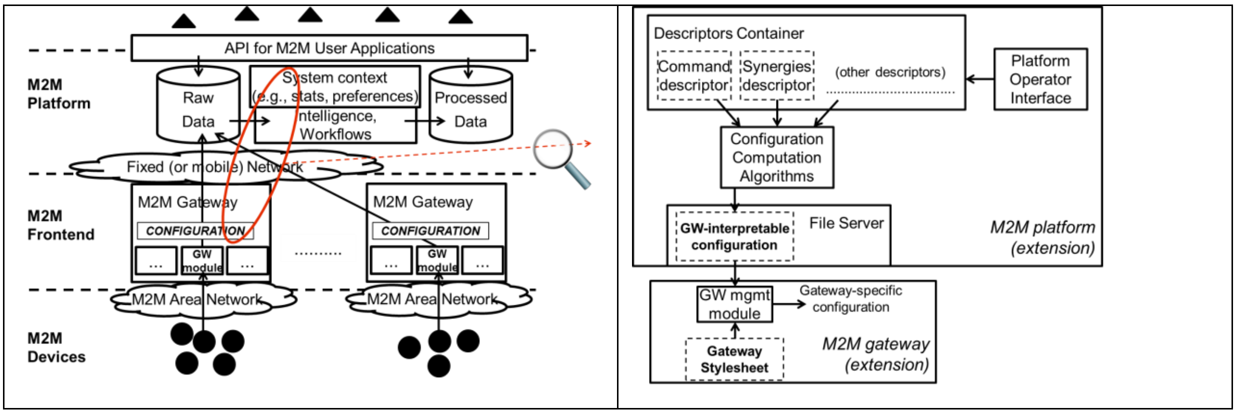


Figure 8. Extension point (left) and extension modules (right) for achieving smart self-configuration

4.1.2 Background and related work

There are quite a few works suggesting generic architectures or extensions for autonomous or self-configuring systems in different domains. Here we focus on approaches perform some kind of system or network parameter tuning, which is similar to what our approach does. Given that the issue has not been handled explicitly for M2M systems and gateway parameters before, we look into approaches from related networking areas and we explain why we approach the issue in a different way.

In the area of LTE, [3] and [7] provide flows with which an eNodeB can be self-configured by applying parameter values that are provided by central configuration servers, but no concrete algorithms for computing configuration values are examined. The work of [5] goes in that direction by providing an optimization algorithm for the tuning of the 3GPP-defined eNodeB parameters ”hysteresis'' and ”time-to-trigger'' in order to perform better handovers. However, this is possible because the central servers know exactly the value ranges of these two parameters and the impact they have on handover performance. In the Big Data domain, self-configuration approaches have only appeared for self-tuning the ”more than 190 parameters of Hadoop'' [2] . However, the techniques used there rely on Hadoop-specific ”rules of thumb'' which are often empirical. For example, a rule recommends setting the number of reduce tasks in a job to roughly 0.9 times the total number of reduce slots in a cluster. Approaches with similar focus and similar restrictions with the above have appeared for MAC protocol self-parametrization [4] , TCP active queue systems [14] , and more.

The above clearly reveals the dependence of the solutions on the domain-specific synergies between system parameters and performance metrics, as well as the necessity for constructs that can help us to deal with the heterogeneity of the configurable modules (i.e., the gateways, in our case). Such constructs can be description languages or standard protocols which respect the heterogeneity of the gateways. Therefore, as we describe in the following subsections, we have developed both an enabling system and various possible algorithms for its parameter tuning logic.

4.1.3 Details of the developed solution

As already implied in Figure 8, the general idea is to get operator goals and system knowledge as input and provide configuration suggestions that are transformed into configuration settings (final output) with the help of style sheets that hide technology- and gateway-specific details. Thus, one of the most important tasks is to specify a well-defined, machine-readable format for all the inputs and outputs of the system. Such formats are often called "description languages" (depending on their scope and complexity). Although the descriptions that follow will use only snapshots of these formats in order to enhance readability, complete versions of all the descriptors (Command Descriptor, Synergies Descriptor, Interpretable Configuration, and Gateway Stylesheet) are provided in Appendix A.

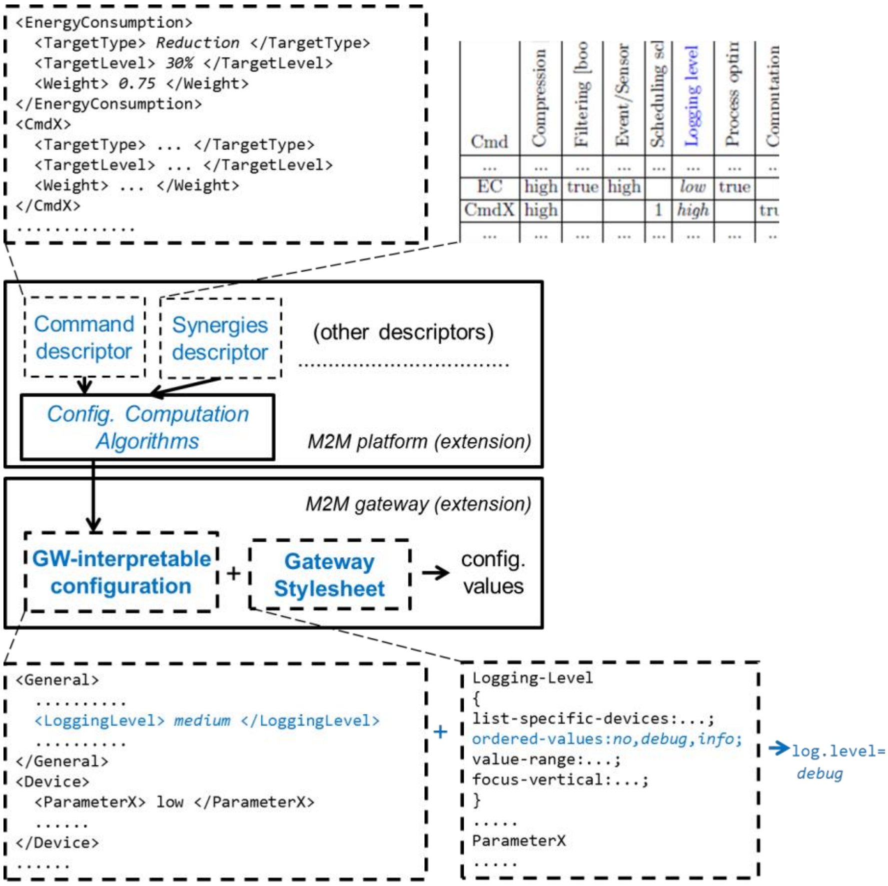


Figure 9. Description of operation of our self-configuration extension with examples

Putting it all together, Figure 9 describes how the system extension for auto-configuration works, namely: Machine-readable descriptions of (a) the operator commands (i.e., the things that she/he intends to achieve via reconfiguration) and (b) the system parameter synergies (i.e., the impact that certain reconfiguration actions have on various aspects), are provided as input. As shown in Figure 9, the *Command Descriptor* contains -among others- weights and target levels for various goals and can be auto-generated from a GUI through which the operator can provide the corresponding preferences. Contrary, the *Synergies Descriptor* is expected to be quite static and will normally be pre-set, though it can also be edited any time. It contains values that indicate how each parameter impacts each possible operator target. The version of the Synergies Descriptor we are currently working with contains 19 commands and 26 parameters, specifying synergies that have been determined through a related educated study (see Appendix A.).

The two mentioned inputs are then processed by a *Configuration Computation Algorithm*. There are practically infinite possibilities for the logic of this algorithm. In this section, we will describe five algorithmic solutions that we have developed. However, the output must be in a specific, well-defined format. More concretely, the output of the algorithm is a *GW-interpretable Configuration*, i.e., a directive of the backend to the gateways in the form of suggested (non-final) parameter values. As shown in Figure 9, the respective machine-readable descriptor contains one tag for each configurable gateway parameter, indicating what the current suggestion for this parameter is. This suggestion must be one of the values that appear in the synergies descriptor for the respective parameter. Finally, each gateway uses its internal *Gateway Stylesheet* in order to "translate" the suggestions into concrete configuration parameter values. Similarly to the way HTML stylesheets can contain one entry for each HTML tag (in order to specify the formatting of this tag in the Web content), a Gateway Stylesheet can contain an entry for each tag of a GW-interpretable Configuration. The example of Figure 9 helps to understand this better: If the suggestion is `<LoggingLevel>medium</LoggingLevel>` and the stylesheet indicates that this gateway supports for this parameter the (ordered) values `no`, `debug`, and `info`, then `debug` should be selected as the final value.

CONFIGURATION COMPUTATION ALGORITHMS

As already mentioned, there are uncountable possibilities for the self-configuration algorithm. Three basic algorithms and two extensions of them, i.e., five algorithms in total, have been currently developed. All of them take the command descriptor and the synergies descriptor as input and provide suggested configurations, i.e. a GW-interpretable configuration, as output. All algorithms have a first optional step in common:

Step 0: Dependencies of gateway parameters on external factors lead to a recalculation of the knowledge base, i.e., the synergies descriptor. External factors refer to any extra information that might dictate adaptation of some values of the synergies descriptor table (top right in Figure 9). For example, extra information might be the compressibility of the data used in the system, actual platform utilization degrees, or feedback from energy consumption measurements.

The rest of the algorithmic steps is summarized in the following, while examples based on the command and synergies descriptor shown in Figure 9 are provided in brackets (note that pseudocode for the algorithms is provided in Appendix A.).

CCA / ECCA algorithms: The Configuration-Comparing Algorithm (CCA), as well as its variation ECCA (Extremal Configuration-Comparing Algorithm), calculates gateway-interpretable configurations based on the idea of separately optimizing each gateway parameter. The following elementary steps are performed *for each gateway parameter*:

- ***Step 1:*** All high-level commands from the operator targets are associated with the corresponding knowledge base entries. (e.g., capture rate: `low` (Costs, Energy Efficiency), `high` (Big Data quality)).
- ***Step 2:*** The classified commands are examined:
    - ***Step 2a:*** Based on the weights of the commands, the most valuable entry (MVE) is calculated. Most valuable means the entry with the highest accumulated weight (e.g. for capture rate: w_{low}(= 0.4 + 0.3) < w_{high}(= 0.8), thus MVE = `high`).
    - ***Step 2b:*** The parameter characteristics are considered. For qualitative values or ECCA, the MVE is directly suggested (e.g. capture rate: `high`). In case of quantitative values and CCA, a target value negotiation is realized: The combination of Target Type and Target Value yields in a specific desired interval of the MVE, a so called scale. A scale is calculated for each command. The scales are weighted in order to map the importance of the commands. The resulting aligned target value represents the suggestion for this gateway parameter (e.g.

capture rate: MVE = `high`, but due to a Target Level of 50% for “Big Data quality” the capture rate is set to `medium`).

WRA algorithm: The Weight Ranking Algorithm (WRA) calculates its output based on the idea of satisfying the commands based on their descending order of weights. The following two elementary steps are performed:

- ***Step 1:*** All commands from the operator targets are sorted in a descending order of their weights (e.g. Big Data quality (0.8) > Energy Efficiency (0.4) > Costs (0.3)).
- ***Step 2:*** The interpretable configuration is filled in a weight ranking-based manner: The commands are considered in the prior calculated order for the purpose of copying their entries as long as these entries are not yet set (e.g. first copy the entries of Big Data quality, then copy the entries of Energy Efficiency -if not already set- and so on).

Since the algorithm strongly depends on weights and follows a ``first things first'' principle, WRA leads to valuable results only if the commands have big differences in importance (which is reflected by their weights).

BADA / BigBadaBoom algorithms: The Balanced Achievement Degree Algorithm (BADA) is based on the idea of a fair, dynamic leveling of the target achievement degrees for all commands. BigBadaBoom is an extension of BADA which considers the command weights for calculating the suggested values, as well. Whereas ECCA, CCA and WRA attempt to reach a maximum total fulfillment, the BADA algorithms are attempting to reach a fairly distributed fulfillment. BADA performs the following fundamental steps:

- ***Step 1:*** The number of “desirably fulfilled parameters” is calculated based on the Target Level and the applicable gateway parameters for each command (e.g., if the Target Level is less than 100%, then not all parameters must be satisfied for a command).
- ***Step 2:*** The interpretable configuration is initialized with possibly existing distinct entries, e.g., if all given commands require a `high` value for a given parameter then this value is directly copied into the suggested configuration.
- ***Step 3:*** BADA/BigBadaBoom iterations are executed for all remaining (unset) entries of the interpretable configuration. In, the following, these iterations are described for BigBadaBoom (BADA works identically, with the difference that it uses simple degrees of target achievement (*DTA*), instead of weighted degrees of target achievement (*wDTA*):
  - ***Step 3a:*** Weighted degrees of target achievement (*wDTA*) are calculated for all commands. The weights slip into the calculation of (*wDTA*) in a non-linear way depending on the gap to a degree of 100% target achievement, i.e. on “how far from achieving an absolute fulfillment the command currently is”. Thus, the closer *wDTA* is to 100%, the smaller is the effect of the weight of a command, so that the command with the lowest *wDTA* is chosen for setting a suggested entry in the next step.
  - ***Step 3b:*** Among the gateway parameter values that satisfy the command selected in 3a, the one which maximizes the aggregated benefit for the other commands is chosen. This benefit depends on whether the specific entry correlates with the remaining entries taking their *wDTA*'s into account, as well. This is because it is better to select a gateway parameter that helps to satisfy most other commands, too, while it is less damaging to set a non-matching entry to a command with an already high *wDTA*.

- ***Step 3c:*** The entry of the ``poorest'' command (in terms of current *wDTA*) that is most beneficial for the other commands (cf. 3b) is written into the suggested GW-interpretable configuration. The algorithm exits when there are no more entries to fill into the latter.

One of the main reasons why these algorithms have been developed is the fact that the logic with which the operators want to fulfill their commands can vary from operator to operator and cannot be known a priori. This means that there is no standard fulfillment metric, which the configuration computation algorithms could try to maximize (potentially by setting up an optimization problem). For each of the presented algorithms, there could be an equivalent objective function, the maximization of which would give exactly the same results as the algorithm itself, only that the optimization-based solution would be much more time-consuming and complex, thus error-prone. For example, the calculation of an optimal configuration based on our knowledge base (26 parameters with an average of 4.6 entries each) involves more than 19 quadrillion ($19*10^{15}$) combinations of parameter entries, which could lead to prohibiting times even if the objective function was linear. However, if an operator comes up with a concrete and different fulfillment metric that she/he wants to maximize, the setting up of an optimization problem instead of using any of these algorithms might be a feasible solution (if the nature of the objective function and the size of the problem allow it).

Instead of trying to enforce a universally applicable fulfillment metric, the set of developed algorithms provides different kinds of solutions to the problem, which can be examined and evaluated with regard to their behavior against different fulfillment metrics. We have performed such an evaluation in order to help operators to choose one of them or decide what kind of logic they prefer for the computation of the suggested configurations. However, this evaluation work is only complementary to the design of our self-configuration extension, it is omitted from this deliverable (interested readers can refer to our respective publication [9]), and it SHOULD NOT be confused with the evaluation of OrPHEuS data filtering solutions, which will be developed in the second year of the project and evaluated in Deliverable D3.1.2.

4.1.4 Further usage in OrPHEuS

Two of the most important M2M platform-related tasks of OrPHEuS are:

- Smart data filtering of OrPHEuS data (i.e., hybrid energy grid data)
- Examination of hybrid energy grid control systems as part of Smart City Operation Centers (and their integration into the latter)

With regard to data filtering, the presented extension for self-configuration regulates –among others- parameters such as *polling intervals*, *compression*, and *filtering thresholds*. The fine-tuning of these parameters will be the core enabler of the data filtering logic that will be developed in the next steps of the project. The suggestions provided by our self-configuration solution and their local interpretations on the gateways will play an important role for the data filtering.

With regard to Smart City Operation Centers (SCOC), the operator interfaces of our self-configuration solution are already being designed as part of an SCOC. More concretely, they will be integrated together with other visualization tools (cf. project task 5.4) into an SCOC framework and their usage inside it will be examined. Section 5.2 provides more information about this.

4.2 M2M platform extension for consumer behavior analysis

This section provides information about the gathering of consumer usage information that can and will be used to plan, predict, and optimize the energy production for different energy sources. This extension is mainly related to the challenges related to application enablement (cf. Table 1, row 1) First, a description of the validity of the work is provided, overview of the solution as extension of our reference architecture is described, then related work is briefly explored, and finally the details of the solution and its further usage in OrPHEuS are described.

To address the challenge of energy efficiency, there is a need to focus on demand side monitoring of energy consumption in user homes and buildings. Further, the use of renewable energy sources presents challenges such as uncertain and intermittent energy production, which requires forecasting and monitoring user energy demands.

Recent times have witnessed the evolution of smart-homes/buildings for different applications such as energy and comfort management systems as well as healthcare. Future smart-homes aim to assist in achieving energy efficiency targets in a cost-effective way. A smart-home/building can be defined as a house or building that is designed to assist and monitor people in their daily activities. It can be leveraged to collect information about energy (electricity, heating, cooling and gas) and water consumption of users. This information can be used to efficiently manage and reduce energy consumption in line with the EU goals.

Smart-homes/buildings may integrate energy management systems that can control energy consumption non-intrusively while maintaining the lifestyle of end-users. This requires non-intrusive load monitoring techniques. Smart-homes/buildings are a sensor rich environment that can enable collection of sensory information for example, temperature, humidity, occupancy, user activity and behavior, appliance type and usage information. This contextual information can be used to build energy efficient systems thereby, providing real-time analytics by leveraging the recent advances in cloud computing. Data analytics relating to user energy consumption can give important cues to build novel energy saving mechanisms at both the production and consumption sides. This will enable dynamic load balancing in a hybrid energy scenario. Different user behaviors can be better suited for different types of energy sources, by understanding and recognizing user activities the user's energy needs can be better matched. This dynamic delivery of energy at different points of time will lead to better utilization of resources for energy production along with efficient consumption of energy at the consumption side.

At the Skellefteå test site, within the district heating (DH) grid, we perform heat load forecasting at the consumer side. A number of internal and external factors affect heat load in a DH grid. Internal factors refer to those at the substation such as supply temperature and flow rate. External factors can be classified into three types namely, behavioral, seasonal and meteorological factors. All these parameters influence the heat load at the consumption side of the DH grid. In [15] , different parameters at the substation along with meteorological data are collected and machine learning techniques such as Support Vector Machines (SVM), Multiple Linear Regression (MLR), Feed Forward

Neural Networks (FFNN) and Regression Trees are applied to predict the heat demand for multi-family buildings in Skellefteå.

Early results shows a good prediction using a forecasting model, in the DH system, for horizon values of 1, 3, 6, 12, 18 and 24-h using a limited amount of training data that makes the method applicable in real time situations [15] .

Additionally, activity recognition at the consumer side will be performed to improve the prediction of energy demand further. This involves the creation of end user energy profiles based on the activity signatures of users living in smart homes/buildings. Activity signatures represent the most likely way in which the user performs an activity [16] . Results in [16] show that a user's complex activities can be recognized using sensors placed in a user's home and office with an overall accuracy of 95.73%. These activity signatures can be further used to build a user's energy profile for different types of energy such as electricity, gas and heating. This involves data collection from sensor enabled homes and buildings where the user performs their daily activities. The data is collected and processed locally and on the cloud using advanced data analytics to create for every user, home and building an energy profile based on their behaviour. The work done in [16] is extended further to achieve the goal of building user energy profiles based on a user's activity information. Further, real-time feedback to the production side or service provider can enable a timely response for the dynamically changing energy demands of consumers. This feedback loop can enable efficient management of resources for energy production.

A significant challenge for energy providers is to deal with peak hour energy consumption. Energy monitoring at the consumption-end by inferring energy usage pattern for different types of energy (electricity, gas, water and heating) can be used to influence consumers to alter their usage patterns during peak hours. Machine learning (ML) techniques can be applied to enable service providers in gaining a better understanding of the energy usage patterns of consumers. Such a system can also be used to recommend alternative behaviour patterns or provide incentives to change behaviour to reduce peak hour energy consumption. Thus, information relating to user energy profiles/usage patterns will play a crucial role in meeting the user's dynamic energy demands and efficient management of energy production resources.

4.2.1 Overview of the extension

In this section, we present the architecture of our target system, which presents the big-picture of this paper. Figure 10 shows the architecture and its key components. In this work, the tasks so far considered are 1) data collection, 2) data aggregation and preprocessing, and 3) the application of ML for prediction. A more detailed description of the target system is presented in [17] . The next step is to extend this system to incorporate external factors such as user behavior or occupancy within smart homes/buildings for efficient energy production and consumption.

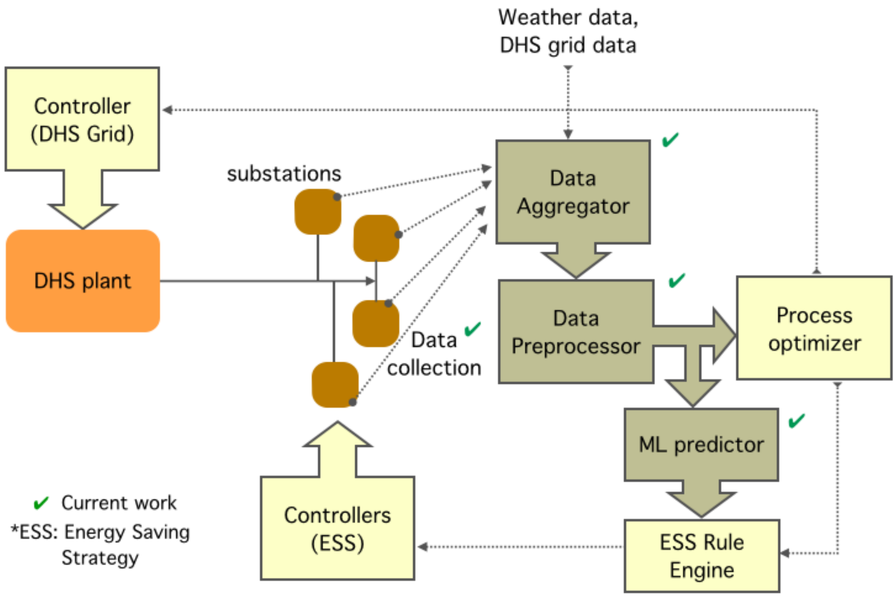


Figure 10. High-level system architecture of the extension for consumer behavior analysis

4.2.2 Background and related work

Production side: Related work that focuses on production side deals with analysis or forecast of **Q**net in DHS (i.e.,top-down approach). Research in [18] presented a novel assessment method which describes daily variations of heat load in DHS. This assessment could be used for design and planning of storage for the DHS network. Further related works ([19] , [20] , [21] , and [22]) have presented load forecasting methods in DHS with limitation to the production environment. A new heat load prediction method is proposed in the work in [20] which uses a recurrent neural network to deal with the dynamic variation of heat load and its characteristics. The approach shows decent prediction accuracy for non-stationary heat load. Further, the studies in [21] proposed a heat load prediction method which is robust enough to handle cases of outliers and missing data. The method uses a simplified robust filter and a three-layered neural network. Research in [22] used a grey-box approach to identify the model that links the heat consumption in a large geographical area to its climate and the calendar information. The process involved a theoretical based identification of an overall building model structure followed by data based modeling. The work in [19] applied wavelet analysis in combination with neural network, and its evaluation shows the approach is suitable for short-term thermal load forecast.

Consumer side: Related work that focuses on consumer side deals with forecast or analysis of **QS**i in DHS (i.e., bottom up approach). Recent works ([23] , [24]) employ the bottom-up approach with focus on the consumption environment and considers single family buildings. The work used computationally effective recursive least squares scheme with meteorological variables as input. The model presented in [23] provides forecast up to 42-hrs forecast horizon.

Our work differs from related work as we also focuses on using ML algorithms for forecasting heat load in multi-family apartment buildings. We also investigate the impact of a combination of internal and external parameters at the consumer/substation side of a DHS. Further, we aim to use information pertaining to user behavior, as an external influencing factor, initially within district heating and later for other types of energy resources to efficiently manage both production and consumption of hybrid energy supply.

4.2.3 Further usage in OrPHEuS

So far the work explained this section is only applied to the DH system as a proof of its applicability. The architectural approach and model are expected to be extended to other types of energy sources as well targeted within the Orpheus project depending on future requirements.

4.3 M2M platform extension for data filtering and fog computing

This section provides initial discussions and draft design elements for the M2M platform extension for data filtering, the design and evaluation of which will be the main focus of the next phase of Task 3.2. Therefore, no platform modules or solution details are described here and the section is limited to providing a background discussion and the most important points of the future usage of this extension in the OrPHEuS project.

4.3.1 Background

The Smart City area includes numerous challenges in machine-to-machine (M2M) and machine type communication (MTC) in terms of the scalability of deployed devices and sensors. It is expected that there will be a multitude of applications and services deployed on top of these devices leading to the requirement for horizontal architectures to be added to the existing M2M platforms. Further, the information captured by sensors, devices and/or machines must be communicated in an efficient way considering that:

- The network infrastructure (wired/wireless) may be stressed due to the huge amount of data communicated.
- Sensor devices have limited capacity with regard to battery lifetime, etc.

The above requires that data processing and storage of information be done in nodes that are located closest to the sensors and where there is sufficient computation and storage capacity according to the application requirements posed by both producers and consumers. According to the idea of fog computing, for M2M this usually means that the processing and filtering of data on the M2M Gateway devices, before they are forwarded and stored into the Smart City backend system. However, hierarchical systems with even more layers than those of Figure 6 might exist, while it might also be required to preprocess the captured data, e.g., by naming them such that they can be searched for without knowledge of IP-addressing schemes or other types of location awareness. When locating the data, routing to the right location for data retrieval can be based on these pre-processed names.

Specifically for energy-related data, for the optimization of heterogeneous energy sources and for energy efficiency in general, activity recognition may also be of interest. This means that we have two datasets that should be considered, i.e., both the dataset that provides information directly from the energy sources as well as the dataset that provides information about the consumers themselves. While the design and evaluation of the M2M platform extension (modules and algorithms) for this purpose is the subject of the next phase of Task 3.1, the remaining part of this section provides already some further explanations using examples from the ICT world of hybrid energy systems.

Examples of handling data close to their sources in energy systems:

In a producer-consumer relation, data should be stored at locations where it can be accessed in a resource efficient manner (e.g. considering communication links, storage and processing devices). Device/sensors/substation/cloud locations need to be discovered and decided in real-time based on the producer, consumer, type of information and processing requirements. The data needs to be discoverable by using information centric approaches to avoid the need of topology awareness. This provides the application domain with a simplified architecture. Energy production in a Combined Heat Power Plant (CHP) and Photovoltaic (PV) system can be facilitated using private clouds, which can be interconnected. Further, cloudlets can be hosted in substations (cf. Figure 11) that processes and stores the information related to the subset of the substations leading to faster data processing and storage for efficient energy production. It may happen that the cloudlet is severely stressed due to increased processing and storage requirements. In such cases the cloudlets may simply offload the processing and storage capacity to the private clouds. Further, performance indicators of communication links need to be considered for offloading decisions.

A private cloud in this context maintains the intelligence that decides on where processing and storage of information should take place when using cloudlets. The private cloud has much higher capacity and is rather scalable to handle the processing and storage requirements of the substations if the need arises. For example, the private cloud may store and process information on behalf of several cloudlets and then disseminate the results to the respective cloudlets. In regards to this, it is necessary that the private clouds are intelligent and have information about all the cloudlets in a DHS. This knowledge is based on the capabilities of substations and metering systems and provides a private cloud with capabilities of defining processing and storage in these systems. Furthermore, in this architecture, mobility of software (application mobility) enables private clouds to instruct cloudlets of the storage, processing and communication needs by installing software components at run-time. This enables an adaptable system that adapts to the current conditions. One example is that with a huge amount of data produced at substations/meters that processing should be managed in cloudlets close to the producers to offload the communication infrastructure and private clouds. The main cloud can be both distributed and replicated with cloudlets at the consumer side.

4.3.2 A Cloud-based M2M architecture for energy distribution systems

A smart city energy distribution systems (EDS) and smart electricity grid (EG) may comprise several Information and Communications Technology (ICT) components including wireless networks and cloud computing systems. The components of an EDS/EG, for example, the power plant, PV enabled homes, buildings and the sub-stations will interact with each other for efficient energy production and consumption. In order to achieve this goal, the data from M2M devices deployed in homes, buildings, substations as well as power plants need to be collected and processed in a timely manner. Further, the data may need to be stored for longer duration for business intelligence and analytics. Figure 11 presents possible reference architecture for M2M and cloud based EDS in a smart city. In this architecture, each major entity, for example, Skellefteå Kraft and SWU/HS ULM may have their own private clouds for secure data processing and storage. Further, for increased data processing and storage capacity (as and when required), the private clouds can also be connected to the public clouds. In an EDS, the data may originate from both the power plant or from the district heating substations at regular intervals that may need to be processed and stored in a timely manner.

To reduce the data transfer between the substations and private clouds, we consider the use of ***cloudlets*** that provide data compute and storage functionality (in places where a high concentration of information exists) closest to the end node (substations/homes), assuming end node cannot process and store the information by itself. The cloudlets will also work as the M2M gateways and can in turn be used to control several parameters related to the end node (M2M device). The use of cloudlets along with private clouds may significantly reduce the bandwidth cellular network and will assist in efficiency. The extension of M2M gateways as cloudlets will facilitate the processing, storage and forwarding of the data originating from the M2M devices (e.g., smart meters). However, one of the biggest challenges here is scalability of cloudlets, private cloud resources, awareness of available communication performance and cloud scheduling. For instance, as mentioned previously, the nearest gateway or cloudlet may be stressed due to the data generated by the M2M devices. In such cases, the private cloud will have to decide where this information will be stored, processed and disseminated. The cloud computing resources of the private cloud will have to be scaled and de-scaled based on the requirements posed by the data generated by the M2M devices and cloudlets. To achieve this goal, we are working towards developing novel scheduling algorithms that decide the cloud resources for content placement and processing by considering several Quality of Service (QoS) parameters such as end-to-end latency and bandwidth and CPU usage.

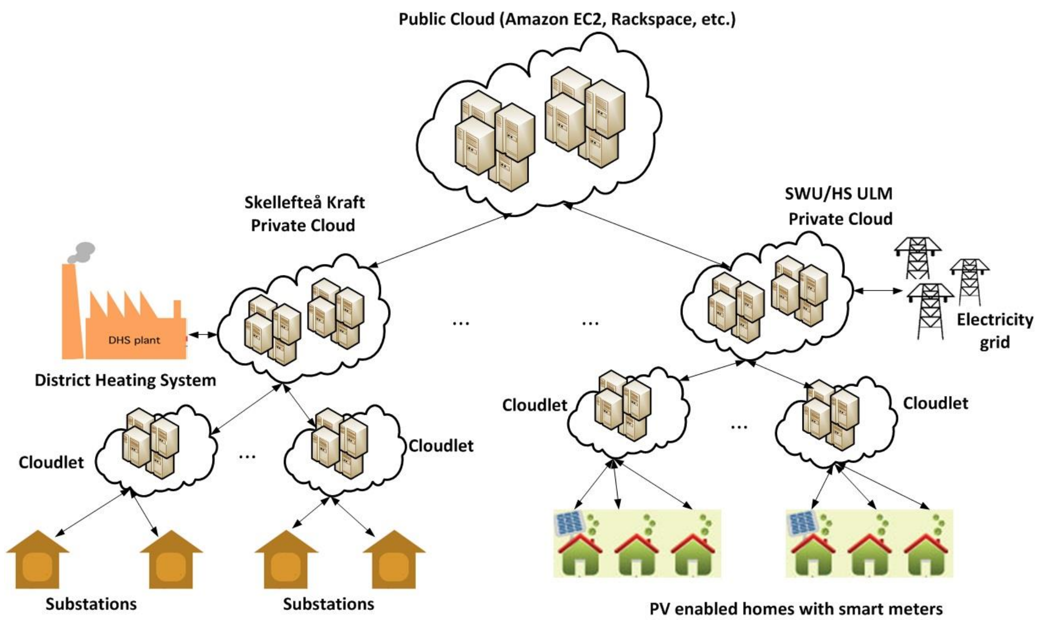


Figure 11. An example cloud-based architecture for smart district heating and electricity grids

4.3.3 Further usage in OrPHEuS

In order to understand the further usage of data filtering and other fog computing-related solutions in OrPHEuS, a list of main future tasks and a list of explicit links to other WPs are provided in the following.

Main future tasks:

- Analytics of existing and possibly future processing and communication demands in hybrid energy systems.
- A M2M Cloud based solution for smart cities targeting energy distribution.
- A prototype implementation as proof of concept.

Main links to other WPs:

- WP3 (see sections 5.1 and 5.2 in D5.2)
- WP4 (possible ICT infrastructure considerations needed for the modeling)
- WP6 (possibly supporting in demonstration via a prototype implementation)

5 Optional tasks for supporting research on control strategies and Smart City operation

In addition to the main task of developing general M2M platform extensions, this task has been concerned with smaller case- or project-specific technical aspects related to the M2M data. A list of such possible tasks was provided in Table 2 of Section 3.2. This Section is briefly explaining the situation of the main tasks of Table 2 that have been identified as relevant for the scope of the project.

5.1 Demo site data aggregation

The data aggregation at demo sites is based on information gathered in existing substation and customer devices as well as possibly new sensors being installed based on requirements in the project. The M2M platform described will support in processing and communicating the gathered information. New gateways may be installed in order to gather complementary information required for achieving the goals of OrPHEuS. Further, the M2M can, based on this, be used for analyzing the scalability requirements of large-scale systems applying the proposed solution

5.2 Integration of OrPHEuS tools in a Smart City operation context

As we already briefly mentioned in section 4.1.4, one of the main goals of the M2M platform-related tasks of OrPHEuS is to examine hybrid energy grid control systems as parts of (bigger and broader) Smart City Operation Centers (SCOC). In a first step towards this goal, we have included in Task 3.1 the design of a prototype, in which operator interfaces for the various possible control actions will be brought together, considering how their “sum” can result in a valuable module of an SCOC.

For the moment, the prototype simply binds a Web interface to the fully-functional implementation of the self-configuration extension described in section 4.1. Therefore, it can demonstrate functions such as:

- Detailed input of (ICT-related) operator goals (“targets”) for specified “areas of interest”
- Selection of self-configuration logic
- Calculation of suggested re-configurations based on the selected logic
- Visualization of the possible fulfillment of the operator goals
- Elastic Cloud based solutions

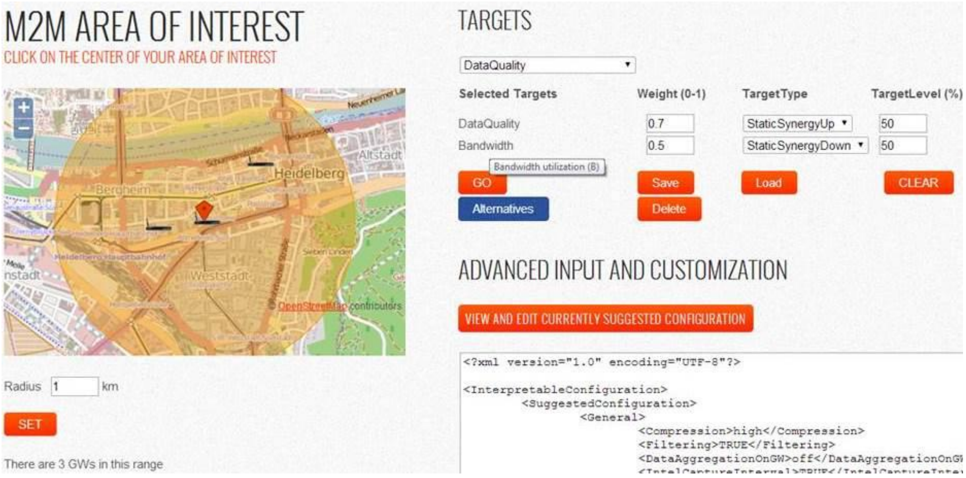


Figure 12. Snapshot of the prototype for integrating OrPHEuS control tasks in an SCOC context

Figure 12 shows a snapshot of the current prototype. However, the goal is to examine how well these platform-related functions integrate with further control functions (e.g., with the visualization tools planned in Task 5.4 or other energy control strategy tools developed in the context of OrPHEuS) under the umbrella of an SCOC. Therefore, no further details are provided at this point.

6 Conclusions

All in all, this Task has:

- Analyzed state-of-the-art Machine-to-Machine (M2M) architectures, deriving their common denominator as the OrPHEuS reference M2M platform design
- Identified M2M platform research challenges based on a related survey of the European Commission and examined which ones should be addressed in the context of OrPHEuS and how
- Designed and developed four platform extensions, namely for supporting self-configuration, energy consumer behavior analysis, data filtering, and cloud based solutions for M2M systems in Smart Cities (under development; main goal of second project year)
- Analyzed the (future) usage and scalability and evaluation of these extensions in conjunction with the work of other WPs, namely their potential impact on control strategies (WP5), their harmonious integration with energy control visualization tools and demo site data aggregation (WP5, WP6), their usage in Smart City Operation Centers, and more.

The achievements include the support of the data-related discussions and the understanding of potential ICT-enhancements within the project, while major parts of the developed solutions have been published in the following high-profile publications:

- Papageorgiou, A., Zahn, M., Kovacs, E. (2014) "Auto-configuration System and Algorithms for Big Data-enabled Internet-of-Things Platforms", in 3rd IEEE International Congress on Big Data (BigData '14). IEEE, 2014, pp. 490-497.
- Papageorgiou, A., Zahn, M., and Kovacs, E. "Efficient Auto-configuration of Energy-related Parameters in Cloud-based IoT Platforms," 3rd IEEE International Conference on Cloud Networking, 2014, under submission.
- Idowu, S., Saguna, S. Åhlund, C., Schelen, O., "Forecasting Heat Load for Smart District Heating Systems: A Machine Learning Approach," to appear in proceedings of IEEE SmartGridComm 2014.
- Idowu, S., Åhlund, C.,and Schelen, O., "Machine Learning in District Heating System Energy Optimization," in IEEE PERCOM '14. IEEE, 2014, pp. 224-227.
- Mitra, K., Åhlund, C., "Mobile Cloud based Computing considering Storage Capability, Computation Performance, and Communication Capacity, under submission.
- K. Mitra, Saguna, C. Åhlund, and D. Granlud, "Mobility Management in Mobile Cloud Computing Systems. (To be submitted to IEEE WCNC 2015)
- K. Mitra, C.Åhlund, Saguna and S. Idowu, "QoS-aware Scheduling of Cloud Resources in Energy Distribution Systems". (Work-in-progress)

7 Bibliography

[1] Boswarthik, D., Elloumi, O. and Hersent, O. (2012) "M2M Communications: A Systems Approach", John Wiley & Sons Ltd, UK.

[2] Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, L.B., and Babu, S., "Starfish: A Self-tuning System for Big Data Analytics," in 5th Biennial Conference on Innovative Data Systems Research (CIDR '11), 2011, pp. 261–272.

[3] Hu, H., Zhang, J., Zheng, X., Yang, X., and Wu, P., "Self-configuration and Self-optimization for LTE Networks," IEEE Communications Magazine, vol. 48, no. 2, pp. 94–100, 2010.

[4] Hurni, P. and Braun, T., "MaxMAC: A Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks," in 7th European Conference on Wireless Sensor Networks. Springer, 2010, pp. 289–305.

[5] Jansen, T., Balan, I., Turk, J., Moerman, I., and Kürner, T., "Handover Parameter Optimization in LTE Self-Organizing Networks," in IEEE Vehicular Technology Conference (VTC '10). IEEE, 2010, pp. 1–5.

[6] Matsuda, N., Sato, H., Koseki, S. and Nagai, N. (2011) "Development of the M2M service platform", NEC Technical Journal, Special Issue on the Network of Things, Vol. 6, No. 4, pp.19–23.

[7] NEC Corporation, "Self-Organizing Networks: NEC's Proposals For Next-Generation Radio Network Management," February 2009. [Online]. Available: http://www.nec.com/global/solutions/nsp/mwc2009/images/SON_whitePaper_V19_clean.pdf

[8] Okuya, S. (2011) "NEC's approach to M2M business", NEC Technical Journal, Special Issue on the Network of Things, Vol. 6, No. 4, pp.9–13.

[9] Papageorgiou, A., Zahn, M., Kovacs, E. (2014) "Auto-configuration System and Algorithms for Big Data-enabled Internet-of-Things Platforms", in 3rd IEEE International Congress on Big Data (BigData '14). IEEE, 2014, pp. 490-497.

[10] Tan, S.K., Sooriyabandara, M. and Fan, Z. (2011) "M2M communications in the smart grid: applications, standards, enabling technologies, and research challenges', International Journal of Digital Multimedia Broadcasting, Vol. 2011, pp.1–8.

[11] Vermesan, O. and Friess, P. (2013) "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", River Publishers Series in Communications.

[12] Wahle, S., Magedanz, T. and Schulze, F. (2012) "The OpenMTC framework – M2M solutions for smart cities and the internet of things", in International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM '12), pp.1–3, IEEE.

[13] Wu, G., Talwar, S., Johnsson, K., Himayat, N. and Johnson, K.D. (2011) "M2M: from mobile to embedded internet", IEEE Communications Magazine, Vol. 49, No. 4, pp.36–43.

[14] Zhang, H., Hollot, C.V., Towsley, D., and Misra, V., "A Self-Tuning Structure for Adaptation in TCP/AQM Networks," in IEEE GLOBECOM '03. IEEE, 2003, pp. 3641–3646.

[15] Idowu, S., Saguna, S. Åhlund, C., Schelen, O., "Forecasting Heat Load for Smart District Heating Systems: A Machine Learning Approach," to appear in proceedings of IEEE SmartGridComm 2014.

[16] Saguna S., Zaslavsky A., and Chakraborty D., Complex activity recognition using context-driven activity theory and activity signatures. *ACM Trans. Comput.-Hum. Interact.*20, 6, Article 32 (December 2013), 34 pages.

[17] S. Idowu, C. Åhlund, and S. Olov, “Machine Learning in District Heating System Energy Optimization,” in IEEE PERCOM ’14. IEEE, 2014, pp. 224-227.

[18] H. Gadd and S. Werner, “Daily heat load variations in Swedish district heating systems,” Appl. Energy, vol. 106, pp. 47–55, Jun. 2013.

[19] M. Wang and Q. Tian, “Application of wavelet neural network on thermal load forecasting,” Int. J. Wirel. Mob. Comput., vol. 6, no. 6, p.608, 2013.

[20] K. Kato, M. Sakawa, and S. Ushiro, “Heat load prediction through recurrent neural network in district heating and cooling systems,” 2008 IEEE Int. Conf. Syst. Man Cybern., pp. 1401–1406, Oct. 2008.

[21] M. Sakawa and S. Ushiro, “Cooling load prediction in a district heating and cooling system through simplified robust filter and multi-layered neural network,” Syst., Man and Cybern., pp. 995–1000, 1999.

[22] H. A. Nielsen and H. Madsen, “Modelling the heat consumption in district heating systems using a grey-box approach,” Energy Build., vol. 38, no. 1, pp. 63–71, Jan. 2006.

[23] P. Bacher, H. Madsen, H. A. Nielsen, and B. Perers, “Short-term heat load forecasting for single family houses,” Energy Build., vol. 65, pp. 101–112, Oct. 2013.

[24] N. Fumo, “A review on the basics of building energy estimation,” Renew. Sustain. Energy Rev., vol. 31, pp. 53–60, Mar. 2014.

8 Disclaimer

The OrPHEuS project is co-funded by the European Commission under the 7th Framework Programme "Smart Cities" 2013.

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission.

The European Commission is not responsible for any use that may be made of the information contained therein.

9 Contacts

Project coordinator

Ingrid Weiss & Silvia Caneva @ WIP – Renewable Energies

Sylvensteinstrasse 2, Munich, Germany

Email: Ingrid.weiss@wip-munich.de / Telephone: 0049 (0) 720 12 742

Email: silvia.caneva@wip-munich.de / Telephone: 0049 (0) 720 12 733

Task 3.1 Leader

Apostolos Papageorgiou @ NEC Europe Ltd.

Kurfürstenanlage 36, Heidelberg, Germany

Email: apostolos.papageorgiou@neclab.eu / Telephone: +49 (0)6221 4342 233

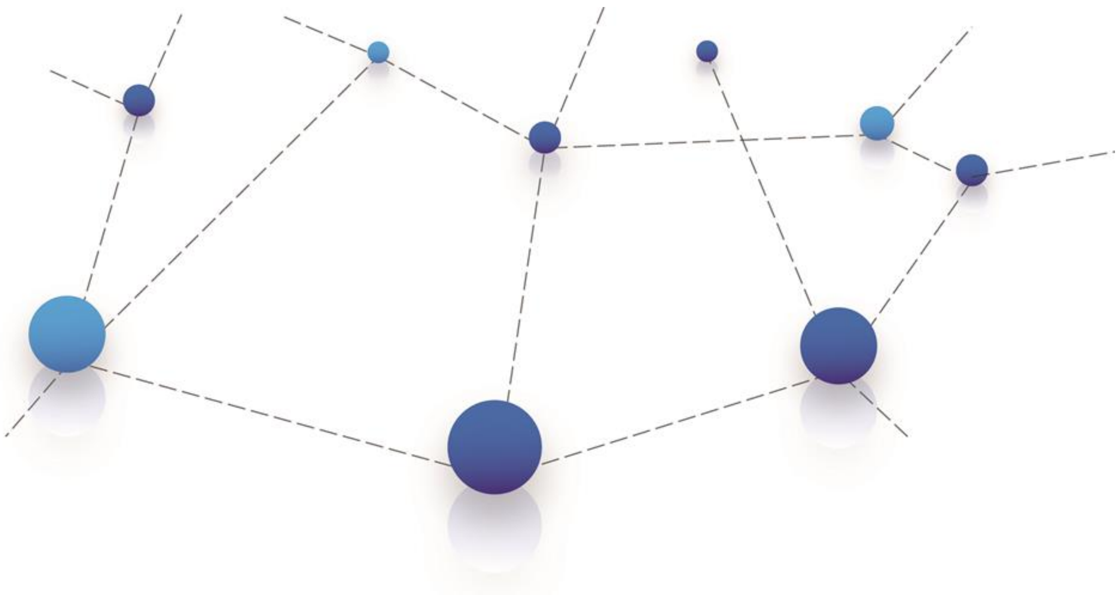
WP3 Leader

Holger Ruf @ Ulm University of Applied Science

Institute for Energy and Drive Technologies

Eberhard Finckh Str. 11, D-89075 Ulm, Germany

Email: ruf@hs-ulm.de / Telephone: +49 (0) 731 50 283



Appendix A. Self-configuration solution details

This Appendix specifies in more detail the machine-readable formats, the pseudocodes of the algorithms, and the evaluation details for section 4.1.

A.1. Self-configuration descriptors

```
<Commands>
  <Costs>
    <TargetType>...</TargetType>
    <TargetLevel>...</TargetLevel>
    <Weight>...</Weight>
  </Costs>
  <BackendEnergyConsumption>
    <TargetType>...</TargetType>
    <TargetLevel>...</TargetLevel>
    <Weight>...</Weight>
  </BackendEnergyConsumption>
  <!--
    .............
    All operator‘s commands
    .............
  -->
  <AreaOfInterest>
    <CenterX>...</CenterX>
    <CenterY>...</CenterY>
    <Radius>...</Radius>
  </AreaOfInterest>
  <VerticalPriorities>
    <TargetType>...</TargetType>
    <TargetLevel>...</TargetLevel>
    <Weight>...</Weight>
    <Domains>
      <Safety>...</Safety>
      <Energy>...</Energy>
      <Automation>...</Automation>
      <Logistics>...</Logistics>
      <Home>...</Home>
      <Infrastructure>...</Infrastructure>
      <Automotive>...</Automotive>
    </Domains>
  </VerticalPriorities>
</Commands>
```

Figure 13. Template of the Command Descriptor

Backend commands		Possible TargetType's [specific criteria]	Compression (M2M system) [off, high]	Filtering (GW) [boolean]	Event/Sensor data aggregation (G [off, low, mid, high]	Intelligent capture rate (D) [boolean]	Scheduling schemes for delay-tol [0..4]	Logging level (M2M system; enca [off, low, mid, high]	Monitoring (M2M system) [boolean]	Process optimization (M2M syste [boolean]	Device computation offloading (D [boolean]	Backend/Gateway selection mechan [0..8]	Capture rate (D) [low, mid, high]	Control of n wireless interf [(vector of boolean)^n]	Sleep phase (D) (rate[s,m,l],	Senso
Time behavior	Latency	StaticSynergyDown	off	FALSE	off		0		FALSE			1		(1,0,0,0,0)	(s,s,P1)	FALSE
Economic factor	Costs	StaticSynergyDown or Adaption [budget]		TRUE				off	FALSE	TRUE	TRUE	5	low	(0,1,0,0,0)		
Security	Security	MinimumSynergy							TRUE	FALSE	FALSE	2				
M2M-specific	Domain priority	SynergySelection [list of (domain, (priority level, optional latency tolerance))]	off		off		0		FALSE			1	high	(1,0,0,0,0)	(s,s,P1)	FALSE
	Scheduling / Synchronizing of data/events	SynergyCreation [schedule]			off							6				FALSE
	M2M Data quality/accuracy	StaticSynergyUp		TRUE	off	TRUE	0		TRUE				high	(0,0,0,0,1)		FALSE
	M2M Device/System diagnostics	SynergyCreation [patterns and rules]			off	TRUE	0	high	TRUE				high	(1,0,0,0,0)	(s,s,P1)	FALSE
	M2M System maintainability	StaticSynergyUp]					0	high	TRUE	TRUE		6				
	M2M Backend energy consumption	StaticSynergyDown or Adaption [energy budget]	high	TRUE	high			off		TRUE		1	low	(1,0,0,0,0)		TRUE
	M2M Device energy consumption	StaticSynergyDown or Adaption [energy budget]	high				1	off			TRUE	1	low	(0,0,1,0,0)	(l,l,P3)	TRUE
	M2M Backend temperature	StaticSynergyDown or Adaption [temperature]	off	TRUE	high			off		TRUE		1	low	(1,0,0,0,0)		TRUE
	M2M Device temperature	StaticSynergyDown or Adaption [temperature]	off				1	off			TRUE	1	low	(0,0,1,0,0)	(l,l,P3)	TRUE
Resource utilization	Balanced loads (B)	Alignment [1 ± x]	off	TRUE	off		2	off		TRUE		3	low			
	Bandwidth utilization (B)	StaticSynergyDown	high	TRUE			2	off		TRUE			low			
	Network utilization (M2M system)	StaticSynergyDown	high	TRUE			3	off		TRUE	TRUE	8	low			
	CPU/Memory utilization (B)	StaticSynergyDown	off	TRUE				off		TRUE		1	low			
	Storage utilization / DB size (B)	StaticSynergyDown		TRUE				off					low			

Figure 14. Currently used synergies descriptor (part)

```
<InterpretableConfiguration>
  <SuggestedConfiguration>
    <General>
      <Compression> ... </Compression>
      <Filtering> ... </Filtering>
      <DataAggregationOnGW> ... </DataAggregationOnGW>
      <IntelCaptureInterval> ... </IntelCaptureInterval>
      <SchedulingScheme> ... </SchedulingScheme>
      <LoggingLevel> ... </LoggingLevel>
      <Monitoring> ... </Monitoring>
      <ProcessOptimization> ... </ProcessOptimization>
      <ComputationOffloading> ... </ComputationOffloading>
      <GWSelectionMechanism> ... </GWSelectionMechanism>
    </General>
    <Device>
      <CaptureInterval> ... </CaptureInterval>
      <WirelessInterfaceState> ... </WirelessInterfaceState>
      <SleepPhase> ... </SleepPhase>
      <DataAggregationOnDev> ... </DataAggregationOnDev>
    </Device>
    <Security>
      <EncryptionOnGW> ... </EncryptionOnGW>
      <EncryptionOnDev> ... </EncryptionOnDev>
      <WirelessSec> ... </WirelessSec>
      <SecVPNChannel> ... </SecVPNChannel>
      <IntegrityProtection> ... </IntegrityProtection>
    </Security>
    <Network>
      <PrefProtocol> ... </PrefProtocol>
      <PrefTransport> ... </PrefTransport>
      <EndpointAssignment> ... </EndpointAssignment>
      <DevFaultMgmt> ... </DevFaultMgmt>
    </Network>
    <Data>
      <Buffering> ... </Buffering>
      <PollingIntervals> ... </PollingIntervals>
      <Queuing> ... </Queuing>
    </Data>
  </SuggestedConfiguration>
  <ForwardedInput>
    <Geo>
      <CenterX>...</CenterX>
      <CenterY>...</CenterY>
      <Radius>...</Radius>
    </Geo>
    <Priorities>
      <Vertical>
        <Name> ... </Name>
        <Weight> ... </Weight>
      </Vertical>
      <Vertical>
        ...
      </Vertical>
    </Priorities>
  </ForwardedInput>
</InterpretableConfiguration>
```

Figure 15. Template of the GW-interpretable Configuration

```
Compression
{
list-specific-devices:...;
supported-ordered-values:...;
value-range:...;
upper-limit:...;
lower-limit:...;
focus-vertical:...;
/* Parameter- or vendor-specific fields */
}
Filtering
{
list-specific-devices:...;
supported-ordered-values:...;
value-range:...;
upper-limit:...;
lower-limit:...;
focus-vertical:...;
/* Parameter- or vendor-specific fields */
}
/* ......... */
/* Any other parameter of the interpretable configuration */
InterpretableParameterX
{
list-specific-devices:...;
supported-ordered-values:...;
value-range:...;
upper-limit:...;
lower-limit:...;
focus-vertical:...;
/* Parameter- or vendor-specific fields */
}
General
{
/* Support of M2M-related technologies */
access-technologies:...;
m2m-protocols:...;
/* Handled data categories */
smallDataNum:...%;
largeDataNum:...%;
smallDataText:...%;
largeDataText:...%;
video:...%;
audio:...%;
/* Served M2M vertical industries*/
safety:...%;
energy:...%;
automation:...%;
logistics:...%;
home:...%;
infrastructure:...%;
automotive:...%;
other:...%;
/* Geo */
coordinates:...;
/* Vendor-specific fields */
<!-- ... -->
}
```

Figure 16. Template of the Gateway Stylesheet

A.2. Pseudocodes of configuration computation algorithms

Algorithm 1. CCA pseudocode

```
_________________________________________________

DEFINITIONS
CE: List of classified entries,
        each list element consists of a CP entry and assigned commands
ES: List of entry scores,
        each list element consists of a CP entry and an assigned score of weights
_________________________________________________

// Input:       CP      (table of commands and parameters)
// Input:       OT      (set of operator targets)
// Output:      IC      (interpretable configuration)
_________________________________________________

for all parameters p of CP
        // Step 0: Consideration of external factors
        if p depends on external factors
                recalculateCP(p)
        end if
        // Step 1: Classification of commands
        Initialize CE
        for all commands c of OT
                if CP[c][p] != null
                        CE.add(CP[c][p],c)
                end if
        end for
        // Step 2: Examination of the classified commands
        if CE.isEmpty()
                IC.getParameter(p).setNoSuggestion()
        end if
        else
                // Step 2a: Calculation of the most valuable entry
                Initialize ES
                for all entries e of CE
                        ES.add(entry,CE.get(e).getSumOfWeights())
                end for
                mostValueableEntry = ES.getEntryWithHighestSumOfWeights()
                // Step 2b: Consideration of parameter characteristics
                if p consists of qualitative values
                        IC.getParameter(p).setSuggestion(mostValueableEntry)
                end if
                else
                        weights = 0, weightedScales = 0
                        for all commands c of mostValueableEntry
                                scale = calculateScale(c.getTargetType,c.getTargetValue)
                                weightedScale = AVG(scale) * c.getWeight()
                                weights += c.getWeight()
                                weightedScales += weightedScale
                        end for
                        alignedTargetValue = weightedScales / weights
                        IC.getParameter(p).setSuggestion(alignedTargetValue)
                end else
        end else
end for

return IC
```

Algorithm 2. BADA / BigBadaBoom pseudocode

```
_________________________________________________

DEFINITIONS
DP:             Map of commands,
                        each command is associated with a double of desired fulfilled parameters
wDTA:           Map of commands,
                        each command is associated with a double of the weighted degree of
                        target achievement
BlackList:      Set of commands,
                        each command of this set is completed and does not need to be handled
                        any more
_________________________________________________
```

```
// Input:        CP       (table of commands and parameters)
// Input:        OT       (set of operator targets)
// Output:       IC       (interpretable configuration)
_______________________________________________________

// Step 1: Calculate the desired number of parameters
for all commands c of OT
        DP.put(c, (50%+c.getTargetLevel/2)*c.getNumberOfSuggestedEntries)
end for
// Step 2: Set distinct suggestions
for all parameters p of CP
        if entrySet of p is distinct
                IC.getParameter(p).setSuggestion(p.getDistinctEntry())
        end if
end for
// Step 3: BADA
minWeight=getMinWeight(OT)
e=0.000001
while IC is not complete
        // Step 3a: Calculate wDTA's and filter the poorest cmd
        for all commands c of OT
                if BlackList does not contain c
                        pro=c.getNumberOfProEntries(IC)
                        contra=c.getNumberOfContraEntries(IC)
                        all=c.getNumberOfSuggestedEntries()
                        if pro+contra==all
                                BlackList.add(c)
                        end if
                        else
                                DTAi=(pro-contra)/all
                                wDTAi=0
                                if BADA
                                        wDTAi=DTAi
                                end if
                                if BigBadaBoom
```

$$wDTAi=\begin{cases}\left(\frac{minWeight}{1-c.getWeight()+minWeight}\right)^{1-DTAi} * DTAi & DTAi>1\\ \left(\frac{minWeight}{c.getWeight()}\right)^{1-DTAi} * DTAi & 0\leq DTAi\leq 1\\ \left(\frac{minWeight}{c.getWeight()}\right)^{DTAi} * DTAi & otherwise\end{cases}$$

```
                                end if
                                wDTA.put(c, wDTAi)
                        end else
                end if
        end for
        poorestCmd=wDTA.getCmdWithLowestWDTA()
        lowestWDTA=wDTA.getLowestWDTA()
        highestWDTA=wDTA.getHighestWDTA()
        // Step 3b:      Calculate the entry benefits
        highestBenefit=Double.MinValue
        highestBenefitParam=null
        for all parameters p of CP
                if IC.getParameter(p).getSuggestion==null && CP[poorestCmd][p]!=null
                        benefit=0
                        for all commands c except poorestCmd of OT
                                if CP[c][p]!=null
```

$$b=2-\left(\frac{|wDTA.get(c)-lowestWDTA|}{|highestWDTA-lowestWDTA|+e}\right)$$

```
                                        benefit+=(CP[c][p]==CP[poorestCmd][p]) ? b : -b
                                end if
                        end for
                        if benefit>highestBenefit
                                highestBenefit=benefit
                                highestBenefitParam=p
                        end if
                end if
        end for
        // Step 3c: Choose highest benefit parameter
        IC.getParameter(highestBenefitParam).setSuggestion(CP[poorestCmd][highestBenefitParam])

end while

return IC
```