European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Large Scale Integrating Project

EXALTED

Expanding LTE for Devices

FP7 Contract Number: 258512

WP7 – Integration & Proof of Concepts

D7.2
Integration of selected algorithms into platforms & interfaces finalization

Contractual Date of Delivery to the CEC:	31 August 2012
Actual Date of Delivery to the CEC:	31 August 2012
Responsible Beneficiary:	ALUD
Contributing Beneficiaries:	GTO, ALUD, CEA, TST, UNIS, CTTC, TUD, VID, SWIR
Estimated Person Months:	76
Security:	Public
Nature	Report
Version:	1.0

Document Information

Document ID:	EXALTED_WP7_D7.2.doc
Version Date:	31 August 2012
Total Number of Pages:	103
Abstract	This report provides a detailed specification of three testbeds developed in EXALTED. Components and interfaces are described, and the implemented algorithms, protocols and features are presented. Moreover, an assessment, how the testbeds contribute to the achievement of the EXALTED objectives is given.
Keywords	EXALTED, LTE-M, capillary networks. E2E connectivity, IPv6, security, device management

Authors

Name	Organisation	Email
Stephan Saur (Editor)	Alcatel-Lucent Deutschland AG (ALUD)	Stephan.Saur@alcatel-lucent.com
Hongfei Du	University of Surrey (UNIS)	hongfei.du@surrey.ac.uk
Bruno Corlay	Sierra Wireless (SWIR)	bcorlay@sierrawireless.com
Nhon Chu	Sierra Wireless (SWIR)	nchu@sierrawireless.com
Sofiane Imadali	Commissariat à l'énérgie atomique et aux energies alternatives (CEA)	sofiane.imadali@cea.fr
Eleftheria Vellidou	Vidavo SA (VID)	projects@vidavo.gr
Javier Valiño	TST Sistemas S.A. (TST)	jvalino@tst.sistemas.es
Juan Rico	TST Sistemas S.A. (TST)	jrico@tst.sistemas.es
Walter Nitzold	Technical University of Dresden (TUD)	walter.nitzold@ifn.et.tu-dresden.de
Jerome d'Annoville	Gemalto (GTO)	jerome.d-annoville@gemalto.com
Christian Ibars	Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)	christian.ibars@cttc.cat
Jesús Alonso-Zárate	Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)	jesus.alonso@cttc.es

Approvals

	Name	Organisation	Date	Visa
Internal Reviewer 1	Srdjan Krco	EYU	31/08/2012	OK
Internal Reviewer 2	Walter Nitzold	TUD	31/08/2012	OK
Technical Manager	Pirabakaran Navaratnam	UNIS	31/08/2012	OK
Project Manager	Djelal Raouf	SWIR	31/08/2012	OK

Executive Summary

Proof of concept with testbed experiments and demonstrations is an effective method for the validation of technical solutions, in particular if different types of devices are included and the technical requirements of a specific application are considered as the benchmark. The report at hand aims at a complete specification of three different testbeds. In a first step, the testbeds are justified by showing the added value and explaining the respective objectives of their realization. Afterwards, the components of the testbeds are introduced and the interfaces between these components are specified. Then, all algorithms, protocols and features, which can be demonstrated with the testbeds, are presented. Finally the relationship with the technical requirements is clarified. This final item is already done in anticipation of the final report of this work package, which will disclose the detailed numerical evaluation of the testbed experiments.

Testbed 1 is dedicated for the LTE-M air interface. It consists of lab equipment that emulates the PHY layer uplink with two candidate LTE-M algorithms, namely General Frequency Division Multiplexing (GFDM) and Code Division Multiple Access (CDMA). A physical LTE-M radio signal can be sent over the air. The main purpose of the testbed is to facilitate the evaluation of these algorithms. Moreover, the testbed provides the proof that LTE-M can coexist with LTE. It can be shown that superimposed LTE and LTE-M signals can be separated from each other and decoded with only minimal mutual impact. As the same receiver is utilized for both communication systems also the intended reusability of hardware is demonstrated.

Testbed 2 addresses different flavours of end-to-end connectivity between non LTE-M devices and a M2M server. It is composed of four autonomous subtestbeds. The cellular LTE-M radio access network is not part of the Testbed 2. Instead its functionality is emulated by a workaround.

- Subtestbed 2.1 reflects a combined Intelligent Transportation System (ITS) and eHealth scenario based on an end-to-end IPv6 framework. The objective is to demonstrate connectivity when mobility, reliability and delay are critical aspects. The subtestbed is composed of several components in the EXALTED architecture, namely the M2M gateway and different types of non LTE-M devices in a capillary network. It can be shown that scalability, efficient and fast routing and reliable mobility management are achieved.
- Subtestbed 2.2 aims at the demonstration of end-to-end connectivity using different access technologies in capillary networks and heterogeneous devices. In particular, it integrates low cost and low power devices from different vendors controlled by enhanced M2M gateways. Various protocols and interactions developed in EXALTED can be shown.
- Subtestbed 2.3 tackles into the problem of maintaining connectivity when the network is composed of very constrained low power devices. Due to the simplicity of the end nodes, a special lightweight address translation mechanism is required. This is demonstrated through the seamless communication between M2M servers in the IP world and these low power devices. Further the subtestbed provides a proof of concept for the required scalability and efficiency within the capillary network.
- The main purpose of Subtestbed 2.4 is to protect application payloads from security attacks. This end-to-end security is demonstrated with a smart metering application. The business case behind is the control of energy produced by photovoltaic panels. In the subtestbed two different form factors of secure elements are utilized, a classical SIM card and a secure element developed for the purposes of EXALTED.

Finally, Testbed 3 aims at device management. Three different subtestbeds show different aspects. Similar as in Testbed 2, LTE-M radio access is not part of Testbed 3. Instead a

workaround is applied emulating the functionality of LTE-M. Testbed 3 exemplifies a smart metering use case. However, it is clarified that the shown solutions are generic. They can thus be applied equivalently for all use cases considered in EXALTED.

- Subtestbed 3.1 demonstrates a novel lightweight device management message encoding which aims to optimize both the message payload size and the processing performance. This is a contribution to the project objectives energy efficiency, spectrum efficiency, and support of a big number of devices, given the fact that in the real system these messages have to be transmitted over the LTE-M air interface.
- Subtestbed 3.2 implements a novel self-diagnostic, which guarantees device reliability and performance while enabling system scalability. EXALTED technologies for the support of fully autonomous devices equipped with self organizing and self healing capabilities for various use cases are shown.
- Subtestbed 3.3 is focusing on security aspects covering device pairing, authentication and provisioning in self organized capillary network. This is an additional proof of concept for the security framework developed in EXALTED complementing the experiments with Testbed 2.

As a general observation, it is claimed that all three testbeds together provide a significant contribution towards the achievement of the project objectives. Testbed experiments and measurements will complement the simulative and analytical evaluation of the proposed algorithms, protocols and features in the technical work packages. A detailed numerical evaluation and comparison with the technical requirements, which is not part of this report, will be disclosed in the final deliverable entitled “Final proof of concept validation, results and analysis”

Table of Contents

1. Introduction 7

2. Testbed 1: LTE-M 8

2.1 Overview and objectives 8
2.2 Components and interfaces 9
2.3 Algorithms and features 12
2.3.1 Generalized Frequency Division Multiplexing (GFDM) 12
2.3.2 Code Division Multiple Access (CDMA) overlay 13
2.4 Tracing of technical requirements 14

3. Testbed 2: End-to-End Communication 16

3.1 Overview and objectives 16
3.1.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario 18
3.1.1.1 Address translation strategy 19
3.1.1.2 Use cases addressed 19
3.1.2 Subtestbed 2.2: Heterogeneity and interoperability 20
3.1.3 Subtestbed 2.3: Connectivity for low-power devices 20
3.1.4 Subtestbed 2.4: End-to-end security 22
3.2 Components and interfaces 24
3.2.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario 24
3.2.2 Subtestbed 2.2: Heterogeneity and interoperability 28
3.2.2.1 Data Collection 29
3.2.2.2 Data Provision 29
3.2.2.3 Data processing 30
3.2.3 Subtestbed 2.3: Connectivity for low-power devices 31
3.2.4 Subtestbed 2.4: End-to-end security 33
3.3 Algorithms and features 35
3.3.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario 35
3.3.1.1 Non LTE-M devices 36
3.3.1.2 M2M Gateway 37
3.3.1.3 Application Server 44
3.3.2 Subtestbed 2.2: Heterogeneity and interoperability 44
3.3.2.1 Device Discovery and Registration 45
3.3.2.2 Semantic Data Dissemination 45
3.3.2.3 Semantic Data Aggregation 46
3.3.3 Subtestbed 2.3: Connectivity for low-power devices 48
3.3.3.1 Non LTE-M devices 50
3.3.3.2 M2M Gateway 51
3.3.3.3 M2M Server 54
3.3.4 Subtestbed 2.4: End-to-end security 55
3.3.4.1 Component features 55
3.3.4.2 LTE-M and Non-LTE-M devices 56
3.3.4.3 Alternative design 56
3.4 Tracing of technical requirements 57

4. Testbed 3: Device Management 59

4.1 Overview and objectives 59
4.1.1 Subtestbed 3.1: Lightweight device management 60
4.1.1.1 Solution overview 60

4.1.1.2 Scope 61
4.1.1.3 Objectives 62
4.1.2 Subtestbed 3.2: Self diagnostic 62
4.1.2.1 Scope 63
4.1.2.2 Objectives 63
4.1.3 Subtestbed 3.3: Secure element device management 64
4.2 Components and interfaces 64
4.2.1 Subtestbed 3.1: Lightweight device management 64
4.2.2 Subtestbed 3.2: Self diagnostic 66
4.2.3 Subtestbed 3.3: Secure element device management 67
4.3 Algorithms and features 67
4.3.1 Subtestbed 3.1: Lightweight device management 67
4.3.1.1 Data transfer 68
4.3.1.2 Managing devices behind the gateway 69
4.3.1.3 Messaging 69
4.3.1.4 OMA-DM v1.x compliant lightweight device management 71
4.3.1.5 Screen captures and Simulator 74
4.3.2 Subtestbed 3.2: Self diagnostic 75
4.3.2.1 Root failure cause detection 75
4.3.2.2 Assisted healing 76
4.3.3 Subtestbed 3.3: Secure element device management 77
4.3.3.1 Group key setting 78
4.3.3.2 Secure element Management 79
4.3.3.3 Key replacement 80
4.4 Tracing of technical requirements 81

5. Conclusion 82

Annex 83

A1. Technical Requirements 83
A2. Key Performance Indicators (KPIs) 84
A3. Mapping of requirements, use cases and testbeds 88
A3.1 Use Case Implementation with EXALTED Architecture 88
A3.1.1 Intelligent Transport System 88
A3.1.2 Smart Metering and Monitoring 89
A3.1.3 eHealth 90
A3.2 Mapping of Requirements to Use Cases 90
A3.3 Implementation of Use Case Functionalities in Testbeds 91
A4. Examples for communication protocol sequences 93
A4.1 Smart Metering and Monitoring 93
A4.2 ITS and eHealth use case 96

List of Acronyms 99

References 102

1. Introduction

The purpose of this report is to describe the progress towards the final specification of the EXALTED testbeds and to highlight the added value for the project.

In the EXALTED project three testbeds are being implemented, two of them are composed of various autonomous subtestbeds aiming at similar objectives but demonstrating different aspects of the big picture.

Testbed 1 addresses the LTE-M air interface. Lab equipment emulates the PHY signal processing of an LTE-M device (transmitter) and of an LTE-M enabled eNodeB (receiver). The added value of this testbed is the significant speed-up of the evaluation of two candidate LTE-M algorithms (Generalized Frequency Division Multiplexing – GFDM, and Code Division Multiple Access – CDMA) investigated in the respective technical work package. Moreover, with Testbed 1, an actual over the air transmission of LTE-M signals can be shown for the very first time. Based on its nature, Testbed 1 is not related to any particular use case, but claims general validity.

The two other testbeds cannot utilize LTE-M because this standard is still being specified, and naturally LTE-M hardware is not yet available. Instead, these testbeds use a workaround emulating the role of the LTE-M radio access network without violating the actual testbed purposes.

Testbed 2 aims at the experimental validation of aspects related with end-to-end connectivity of heterogeneous devices. It addresses scenarios, where these devices are located in capillary networks, which are designed for a particular application, e.g. eHealth devices used in an ambulance. An important further aspect in Testbed 2 is the end-to-end security between device and M2M server. The validation of Testbed 2 will be based on the technical requirements of the respective applications considered in the four subtestbeds.

Finally, Testbed 3 is dedicated for novelties in the field of device management. Considered aspects in the three subtestbeds are the lightweight device management message encoding, a novel self-diagnostic for reliability and performance, as well as security with respect to device pairing, authentication, and provisioning in self organized capillary networks. The validation of Testbed 3 will be done for a smart metering and monitoring use case as a representative example. However the demonstrated functionality exhibits general validity.

The description of all three testbeds in this report follows the same principle. In a first section, a brief overview of the characteristics of the respective testbed is given. Furthermore, the main objectives are highlighted to show the added value for the project. Afterwards, in a second section, the used components and interfaces are explained. The third section summarizes the different algorithms and features that can be shown with the testbed. Finally, in a forth section, the tracing of technical requirements, which are relevant for the testbed, is described. However, it is clarified that this tracing reflects the current status and not the final result. A detailed numerical evaluation is not part of this report.

Four annexes are added to this report. The first two summarize the technical requirements and the Key Performance Indicators (KPI). The third provides mapping tables showing the relationship between use cases, technical requirements and the three testbeds, and the last one exemplifies communication protocol sequences.

2. Testbed 1: LTE-M

2.1 Overview and objectives

The main purpose of Testbed 1 is the efficient evaluation of the LTE-M PHY layer algorithms Generalized Frequency Division Multiplexing (GFDM) and Code Division Multiple Access (CDMA)-overlay. Algorithms and protocols that belong to other layers are not implemented. Also, it is not the intention to demonstrate the applicability of the algorithms for one specific use case or application, but to show their generally valid functionality. To understand the general idea and the objectives of this testbed, and to justify why it is needed in the project, it is firstly described in a simplified manner.

Commonly used tools for PHY layer algorithms are link level simulation chains. They usually consist of three components, the transmitter, the radio channel and the receiver. All transmitter and receiver baseband algorithms, finally implemented on Field Programmable Gate Arrays (FPGA) or Digital Signal Processors (DSP) are firstly tested and evaluated with software tools, e.g. MATLAB. While the MATLAB versions of these algorithms never differ from their behaviour in the actual LTE-M device and the eNodeB hardware, this is not the case for the radio channel. Therefore, several software models for the radio channel have been proposed, e.g. the Spatial Channel Model Extended (SCME) [1], but independently which of these models is applied in the simulation chain, it is still a model and not the reality. Typical parameters of channel models are the number of propagation paths, their delays, their mean attenuation, their angles of departure and arrival, and of course the statistical metrics describing how all these parameters change over time. Some channel models consider typical scenarios like indoor propagation, dense urban or rural scenarios, ray reflection on a water surface, diffraction at roof tops and so on.

Realistic channel models add a lot of complexity and computational effort in the simulation tool. An analysis of the MATLAB link level simulation tool has shown that the channel model consumes nearly 80% of the overall computation time. This fact excludes a broad evaluation, i.e. the simulation with different channel models, different system parameters, different signal-to-noise-power ratio (SNR) values and different modulation and coding schemes is not possible in a reasonable time. A solution for this problem is the so-called Hardware-in-the-Loop (HaLo) principle. In a simplified view, HaLo replaces the computationally intensive MATLAB equivalent lowpass radio channel model, just by the real radio channel. The difference between a pure software simulation chain and the HaLo is shown in Figure 2-1. In both cases the transmitter algorithms are executed in the MATLAB simulation chain. Instead of feeding the time samples in the software channel model, the HaLo transmitter constructs a physical transmit signal including up conversion to the carrier frequency and radiation from an antenna. After the propagation over the air, the signal is received, filtered and down-converted by the respective inverse entity labelled with Sorbas in Figure 2-1. The result is a set of time samples in exactly the same format than the output of the software channel model. They are further processed in the MATLAB chain with the receiver algorithms.

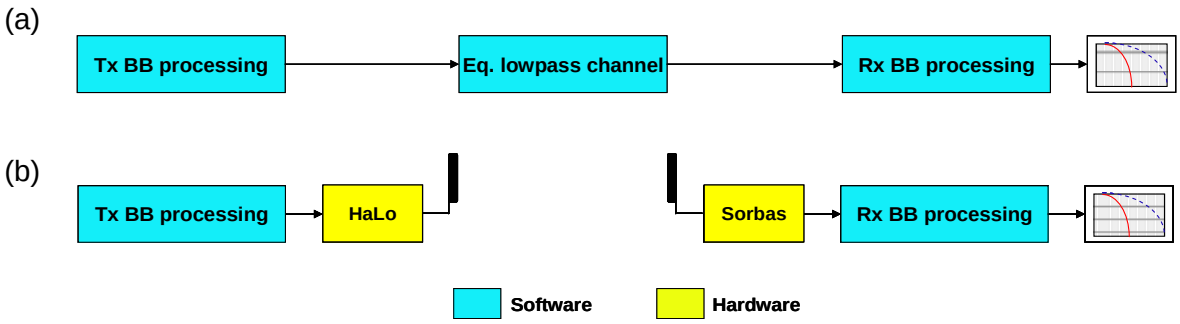


Figure 2-1: Performance evaluation with (a) link level simulation chain and (b) HaLo over the air replacing the software channel model

The added value of Testbed 1 to the project is twofold:

- Due to the fact that the time consuming equivalent lowpass software channel model is replaced by an actual over the air transmission, the computation time of the simulator with HaLo is significantly reduced by about 80%. Or in other words, in the same available timeframe, the fivefold amount of results can be produced, or alternatively the statistical accuracy of the results can be improved by averaging over five times more representatives of events like bit errors.
- The fact that the results are achieved by an actual over the air transmission instead of a more or less accurate model clearly stresses their relevance as long as the scope of the experiments fits to the considered scenario, in this case indoor propagation in the lab, where the HaLo system is installed. There are still degrees of freedom: Obstacles and reflectors can be placed between transmit and receive antenna to create Non-Light-Of-Sight (NLOS) conditions with multiple propagation paths.

On the other hand, it is clear that HaLo has also some disadvantages compared to a pure software simulator:

- The exact propagation conditions valid for one experiment cannot be reproduced for another experiment.
- The HaLo system is installed in a lab, i.e. only scenarios assuming indoor propagation can be investigated.

As conclusion, a broad evaluation of LTE-M algorithms requires simulations with a software channel model as well as additional simulations assisted by the HaLo principle with Testbed 1. Both tools complement each other, and both are available.

2.2 Components and interfaces

After the justification of the testbed by clarifying the purpose and the added value, a more detailed view of the components and interfaces is disclosed. They are illustrated in Figure 2-2. For a better understanding of the following description, it is important to know that only the uplink is considered in Testbed 1, i.e. the transmitter is an entity representing the LTE-M device and the receiver is an entity representing the eNodeB.

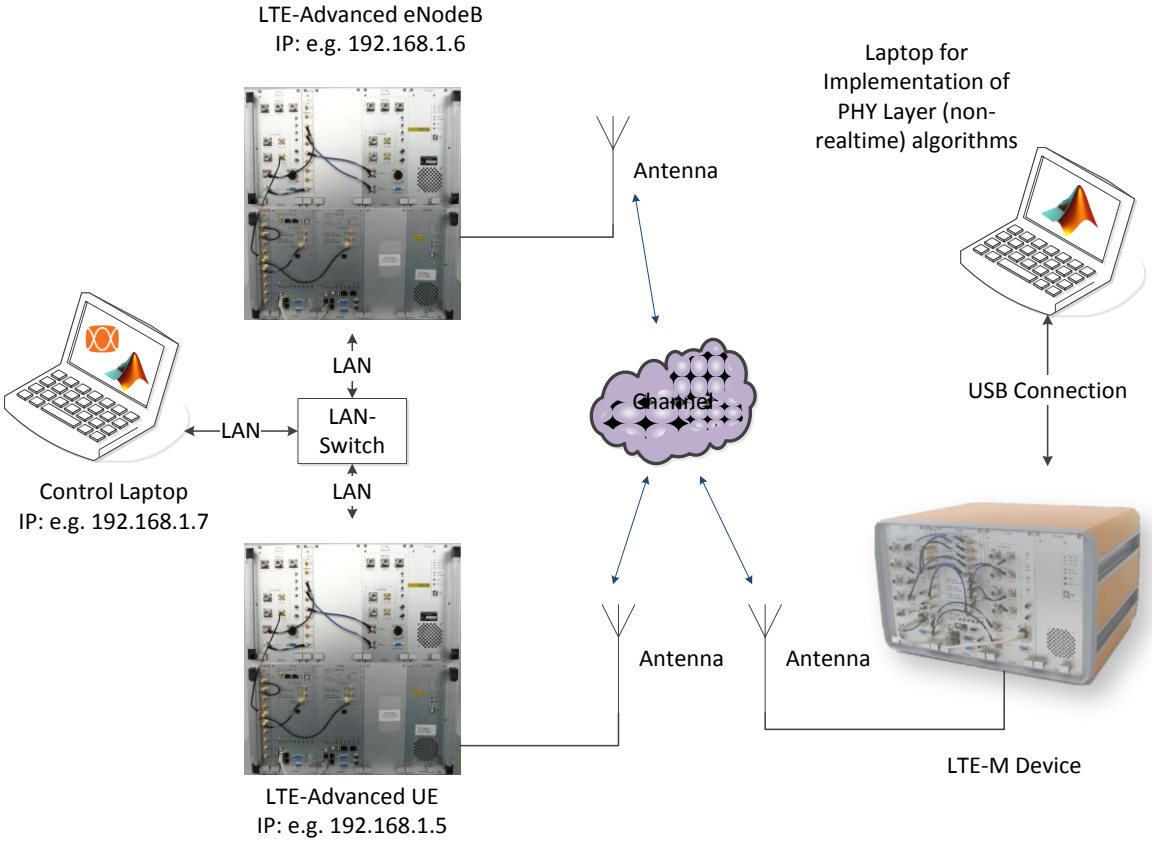


Figure 2-2: Schematic view of the components and interfaces of Testbed 1

The Testbed 1 setup shown in Figure 2-2 consists of the following components and interfaces:

Laptop for implementation of PHY Layer algorithms: It complies with the blue box in Figure 2-1 labelled with Tx baseband (BB) processing. On the laptop, a MATLAB simulation chain with all required PHY layer transmitter algorithms like coding, modulation, CDMA overlay, GFDM pulse shaping etc. is executed. Algorithms developed and investigated in the scope of EXALTED are part of this simulation chain. The outcome of this signal processing is a MATLAB vector with time samples representing the baseband LTE-M transmit signal. In a pure software simulator these time samples would be fed in the equivalent lowpass software channel model. According to the investigated algorithm, the MATLAB functions can be arbitrarily exchanged or configured. The computation time depends on the processor of the laptop, i.e. usually the provision of time samples to the HaLo cannot be achieved in real-time. As a consequence, Channel Quality Indicator (CQI) feedback and the adaptation of the link parameters at the transmitter according to the CQI are not possible with HaLo.

LTE-M Device: It has a USB connection with the laptop and complies with the yellow HaLo box in. All necessary processing steps to build a physical radio frequency signal out of the MATLAB time samples to be transmitted from the antenna shown in the diagram are executed in this device, which is controlled by the laptop with a powerful Application Programming Interface (API), written in MATLAB. Main functionality is the up-conversion to the carrier frequency. Hence, together with the laptop, the complete PHY layer functionality of a LTE-M device is present. The realization of this device is not part of the project, but existing equipment from Signalion available at the Technical University of Dresden (TUD) Mobile Communications labs. The access to the hardware for evaluation and testing purposes is restricted to the local premises of TUD.

Channel: It is a real radio channel for over the air transmission of radio frequency signals. It can be modified by placing obstacles and reflectors between transmit and receive antenna, or by changing the distance between the antennas. Apart from a Line-of-Sight (LOS) component, also propagation paths due to reflections at walls, floor, ceiling and other items in the lab room are present. The propagation conditions cannot be completely reproduced.

LTE-advanced eNodeB: It is not a real world eNodeB, but lab equipment emulating a part of eNodeB functionality together with the control laptop. This component is equivalent to the yellow Sorbas box in. It is equipped with an antenna that receives the signal that was sent from the LTE-M Device. Basic functionality is filtering, sampling and down-conversion of the physical signal. The result is a set of time samples representing the received baseband signal usable in MATLAB. The Sorbas is not an outcome of EXALTED, but existing equipment available at the Technical University of Dresden (TUD) Mobile Communications labs.

Control laptop: It is connected with the LTE-advanced eNodeB (Sorbas) by a LAN cable. It is equivalent to the blue box labelled with Rx BB processing in Figure 2-1. All baseband receiver algorithms like channel estimation and decoding are executed on this laptop. A part of these algorithms have been developed and investigated in the scope of EXALTED.

LTE-Advanced UE: LTE-M is a system that coexists with LTE in the same frequency band, i.e. LTE and LTE-M radio resources are multiplexed in time and frequency in the uplink frame structure. A detailed specification can be found in the EXALTED project report D3.3 [2]. The algorithms investigated in EXALTED aim at an optimization of this coexistence. Thus, the transmission of a LTE-M signal solely is for a broad evaluation not sufficient. This component, which is not shown in Figure 2-1 for the sake of simplicity, represents a regular LTE UE. It is utilized for the evaluation in EXALTED, but not a project achievement itself. It transmits a LTE conform signal on time-frequency resources that are not occupied by LTE-M. This means that the LTE-Advanced eNodeB receives a compound LTE / LTE-M radio signal, which can be separated in a LTE signal (with not avoidable LTE-M signal components) and a LTE-M signal (with not avoidable LTE signal components). Both of them can be processed and evaluated individually on the control laptop. The effects of mutual interference in terms of Out-Of-Band (OOB) transmission can be analyzed, which is also an important part of the work in EXALTED.

After this high level description of the main components of the testbed and their functionality, the focus is now on the most important interface that can bring together contributions from different partners. It is the already mentioned MATLAB API that is controlled by the laptop, that is responsible for the transmit signal processing. Its specification and implementation in MATLAB simulation chains from different partners are achievements of EXALTED. The interface is detailed in Figure 2-3.

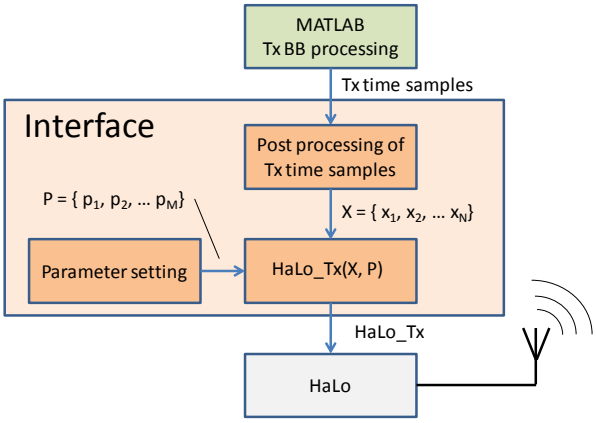


Figure 2-3: Interface between Tx BB processing and the HaLo

The simulation chain responsible for transmit baseband processing can be any proprietary tool. It does not depend on the specification of Testbed 1. The expected output is a vector with time samples representing the LTE-M baseband transmit signal in a format that can be read by MATLAB. Alternatively, instead of a simulation tool, the samples can be obtained also by a hardware signal generator.

Then, the samples are read by the MATLAB API. Eventually, they have to be post processed:

- Allowed sample rates are 5 MHZ, 10 MHz, and 20 MHz. If the original vector exhibits a different rate, a re-sampling has to be carried done.
- The length of the vector must be a multiple of 128. If needed, zeros have to be added to achieve this.

Before the HaLo can be called, some additional parameters must be set. They include

- Rate of the time samples (5, 10, 20 MHz)
- Carrier frequency
- Frame length
- Transmit power
- Continuous mode – if this option is active, the transmission of the data is repeated in an endless loop.

Finally, the HaLo function is called. The source code of this function is subject to intellectual property and won't be disclosed. Arguments are two vectors, namely the post processed time samples X and the set of parameters P. The respective RF signal is then radiated from the antenna connected to the HaLo.

2.3 Algorithms and features

The general functionality and features of Testbed 1 have been introduced in the previous subsection. In the following, the two PHY layer solutions GFDM and CDMA-overlay that will be evaluated with Testbed 1 are briefly described. Details of the algorithms are given in the EXALTED project report D3.3 [2].

2.3.1 Generalized Frequency Division Multiplexing (GFDM)

GFDM is a digital multi-carrier transceiver concept, derived from the general idea of filter banks. It aims to extend traditional Orthogonal Frequency-Division Multiplexing (OFDM) by introducing additional degrees of freedom that primarily address the spectral shaping of the transmitted signal as well as the spectral efficiency of a wireless communication. In this context, GFDM provides means to contain out of band radiation through pulse shaping with adjustable matched filters. However, sharp edges in frequency domain response have to be traded for a greater spread of the signal in time domain. To avoid increasing the length of the cyclic prefix with the pulse shaping filter length, GFDM relies on tail biting. Further, by combining several multi-carrier symbols as they are known from OFDM to one GFDM block, the amount of Cyclic Prefix (CP) per data can be reduced and thus spectral efficiency increases.

The GFDM approach is a generalized vision on single-and multicarrier access schemes and aims on a contemporary understanding of the mechanisms for radio access for different requirements. As M2M scenarios have such diverse requirements GFDM is the starting point for deeper investigations. It is a generalization of OFDM and singe-carrier frequency division multiple access (SC-FDMA) and therefore covers these solutions too while bringing into play a wide range of flexibility.

The GFDM approach enables the LTE-M system to provide the spectrum efficient and backward compatible data transfer from the LTE-M Device to the base station (eNodeB) and therefore aims at the fundamental objectives of the EXALTED project.

2.3.2 Code Division Multiple Access (CDMA) overlay

The mechanisms proposed in the EXALTED project report D3.1 [3] aiming at a reduction of complexity and cost of LTE-M devices, e.g. the restriction to one single antenna or a low transmit power, may lead to impairments in the link budget and in consequence to a downsizing of the coverage area. Hence, LTE-M must provide means for coverage extension, particularly for scenarios with severely changing radio propagation conditions. One promising and flexible approach is a CDMA overlay.

Figure 2-4 illustrates the addressed problem. Without CDMA overlay, only the transmissions originating from LTE-M devices close to the eNodeB (yellow and red points in the solid circle) can be received. The link budget of the green device outside of this circle is too bad. With CDMA overlay, the coverage area is extended (dashed circle), and all three considered devices can be supported. The superposition of transmitted data symbols with orthogonal codes is fully compatible with the LTE-M frame structure proposed in the EXALTED project report D3.3 [2].

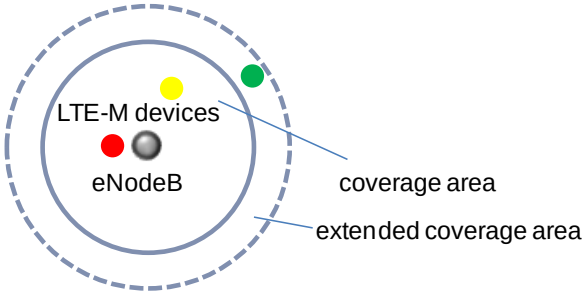


Figure 2-4: Range extension in LTE-M

The principle of the CDMA overlay is shown in Figure 2-5. We consider one single subcarrier and omit the frequency dimension in the following for the sake of simplification. The left figure depicts the consecutive transmission of three data symbols in the uplink. These symbols may originate from the three coloured points in Figure 2-4.

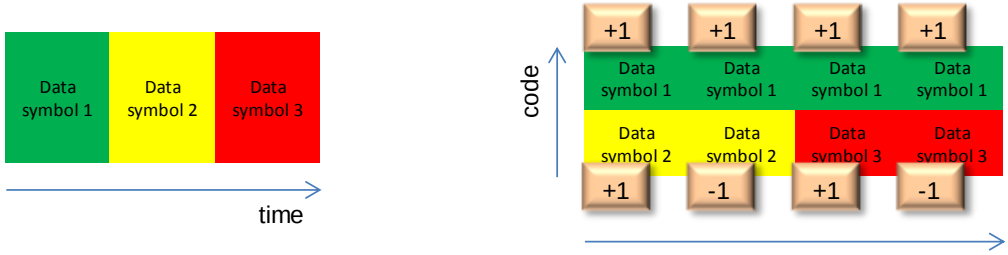


Figure 2-5: Transmission of data symbols (a) without and (b) with CDMA overlay

In order to achieve the required link budget improvement for the green LTE-M device, its transmission time is extended in the right figure from one to four GFDM symbols. The eNodeB accumulates the received energy until the message can be decoded. User separation is done by multiplying the data symbols at the respective transmitters with orthogonal codes and by the usage of a correlation receiver at the eNodeB. Assuming ideal conditions, there is a linear relationship between transmission time and accumulated energy at the receiver, i.e. expanding the transmission time by a factor of 2 yields a link budget improvement of 3 dB. A performance assessment based on link budget analysis is shown in the EXALTED project report D3.3 [2].

2.4 Tracing of technical requirements

The purpose of this subsection is to give an outlook which of the technical requirements relevant for the EXALTED use cases can be traced with Testbed 1, and how the evaluation will be done.

As already clarified in the introductory paragraph for Testbed 1, the purpose is to demonstrate the generally valid functionality of LTE-M on PHY layer only. Therefore, it is not intended to validate the suitability of the algorithms with respect to one specific use case or application.

Table 2-1 summarizes the technical requirements considered together with the aspired outcome of our experiments in a qualitative style. The last column indicates the current status of the evaluation. The final numerical results will be disclosed in a later report. An explanation of the requirements and the Key Performance Indicators (KPI) can be found in Annex A-1 and Annex A-2.

Table 2-1: Technical Requirements

ID	Title	Priority	Goal	Fulfilment
FU.2	Efficient spectrum management	Mandatory	The aim is to show that LTE-M GFDM signals can fit in very small spectrum gaps not occupied by LTE users (specification of joint LTE/LTE-M uplink frame see report D3.3 [2]). The used KPIs are (K1) BER, (K6) Peak-to-Average Power Ratio (PAPR), and (K7) OOB	Ongoing
SV.1	Overall QoS concept	Mandatory	As Testbed 1 is restricted to the uplink PHY layer of LTE-M, the tracing of this requirement is limited to a qualitative assessment based on (K1) – BER observations.	Not started yet
NT.2	LTE-M backward compatibility	Mandatory	Testbed 1 shows two indications: • Only negligible performance degradation for LTE users with respect to (K1) – BER • At least with respect to uplink PHY, the inclusion of LTE-M does not require additional hardware in the eNodeB	• Ongoing • Done
NT.3	Minimum number of modifications in network infrastructure	Mandatory	In can be shown qualitatively that the operation of LTE-M PHY uplink in Testbed 1 does not require any changes in the network infrastructure.	Done
NT.5	Half duplex transmission mode	Medium	It can be shown qualitatively that LTE-M PHY functionality supports the application of half-duplex operation.	Ongoing
NF.1, FU.1	Scalability, Support of large number of devices	Mandatory	The fact shown by experiment with Testbed 1 that GFDM can occupy very small, but scalable frequency resources indicates that at least LTE-M PHY uplink supports a big number of short messages transmitted simultaneously. This observation is complemented with the proof that a	Ongoing

			signal with CDMA-overlay consisting of components from more than one LTE-M device can be decoded successfully. The used metric is (K1) – BER.	
NF.2	Energy efficiency	Mandatory	The tracing of this requirement with Testbed 1 is limited to a qualitative assessment. The capabilities of GFDM enable efficient duty cycles in the sense of sporadic wake-up of devices and one shot transmissions in small spectrum junks.	Ongoing

It must be clear that a full proof for the fulfilment of these requirements with Testbed 1 solely is not possible because its functionality is limited to the PHY uplink. But it can give important indications that complement simulation studies and theoretical analysis.

Moreover, experimental findings will be fed back to WP3 and implemented in software models as radio link abstraction, i.e. a mapping function between SNR values and the corresponding BER (K1). Consequently a certain SNR distribution predicted by system simulation leads to estimates for throughput (K12) and spectral efficiency (K14). This mapping is also of particular interest for the evaluation of the CDMA-overlay with respect to the objectives range- (K25) and coverage extension (K26), which cannot be investigated in Testbed 1 directly, due to its nature being lab equipment.

3. Testbed 2: End-to-End Communication

3.1 Overview and objectives

This scenario will focus on demonstrating the novelties developed in three main fields:

- Concepts developed within capillary networks
- Providing end-to-end connectivity between M2M devices, not only belonging to the same capillary network, but also from different ones connected to the LTE/LTE-M network.
- Providing end-to-end security between M2M devices and M2M server

For these purposes and in order to address every scenario and use case, this second testbed can be seen as a composition of four subtestbeds each one centred on its capillary network. This represents the real-life sub-networks each one addressing a set of requirements for the final user.

Figure 3-1 shows the different pieces of hardware and their relationship with the proposed algorithms, as well as the way connectivity is shown:

- **Subtestbed 2.1** is ITS and eHealth oriented. The objective is to prove connectivity when energy efficiency and high compression is not critical, but mobility, reliability and delay are crucial aspects to take into account. eHealth use case is intended to be built on top of a vehicular use case, using different elements in the architecture combining cellular radio communications technologies and capillary vehicles, with an E2E IPv6 framework that assures scalability, efficient and fast routing and reliable mobility management.
- **Subtestbed 2.2** is aimed to demonstrate connectivity for a Smart Metering and Monitoring (SMM) scenario using different technologies and heterogeneous devices. Several radio interfaces will be used, integrating low cost and low power devices controlled by enhanced M2M gateways implementing software capabilities developed within EXALTED. The result is a common testbed using different vendor devices interacting together and managed via purpose specific protocols derived from the work in the project.
- **Subtestbed 2.3** tackles into the problem of maintaining connectivity when the network is composed of very constrained low power devices, again using the example of SMM. Due to the simplicity of end nodes, they cannot implement a traditional communication protocol stack as it is, so there is a need for defining a lightweight address translation mechanism able to provide seamless communication between M2M servers in the IP world and these low power devices. With that implementation, E2E connectivity and scalability are achieved while maintaining the efficiency required in the capillary network.
- Purpose of the **Subtestbed 2.4** is to protect application payloads from security attacks. This End to End security is performed in the context of a SMM application deployment where some meter index values are sent by devices to the M2M server to be analyzed. Business case is to control on a server the energy produced by photovoltaic panels. Service provider wants to protect the collected values against malicious tampering. Two different Secure Element form factors are used with this subtestbed: either a SIM card or the Secure Element developed for the purpose of the project. In the former case the SIM of the device used to manage the communication with the network is also used to protect messages while the EXALTED Secure Element is embedded on a non LTE-M device in the latter case.

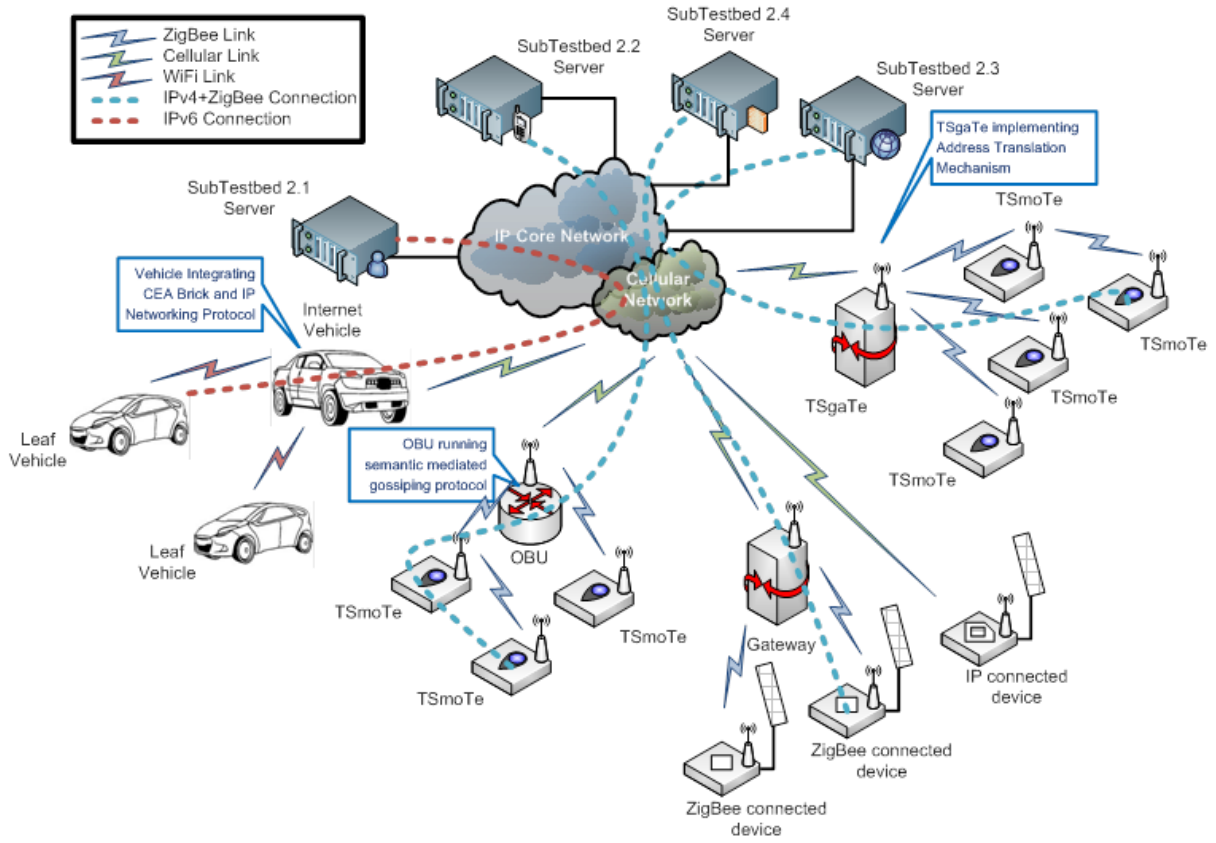


Figure 3-1: Hardware, algorithms and connectivity mapped for Testbed 2

The combination of the four subtestbeds makes this proof of concept applicable to all key use cases assumed by the project. Moreover, all of them can coexist and share the same location sharing the same physical resources, so it enables demonstrating the scalability, interoperability and heterogeneity of the applications.

As a summary, Table 3-1 shows the connection between each testbed and its main goals and the applicable use cases from EXALTED.

Table 3-1 Summary of connectivity oriented testbed

Subtestbed	Goals	Use Cases
2.1	• Scalability • Efficient routing for complex use cases • Mobility management • Reliability of data transferred	ITS and eHealth
2.2	• Heterogeneity • Energy efficiency • Implementation of complex gateway routines • Interoperability	SMM
2.3	• Scalability • Energy efficiency • Payload compression • Efficiency on capillary transmissions • Leverage of LTE-M data transferred	SMM
2.4	• End-to-end security	SMM

The following subsections describe in detail the components and objectives of each subtestbed. Furthermore they set up a relationship with the use case that is applicable to each of them.

3.1.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario

This subtestbed, while combining vehicular and eHealth use cases, focuses on IPv6 connectivity with its relevant addressing scheme and on the reliability of the transferred data.

Intelligent Transportation Systems (ITS) [4], [5] are envisioned to play a significant role in the future, making transportation safer and more efficient. With respect to these expectations, Vehicle-to-Infrastructure (V2I) interactions as advanced by the EXALTED project (capillary to infrastructure), have evolved to include various types of applications, safety-related and user-oriented (infotainment).

The Subtestbed 2.1 demonstrates eHealth as one very pertinent application example of vehicular networking amongst other possible V2I and Vehicle-to-Vehicle (V2V) examples which can be roughly classified in two major types [6]:

- Safety-oriented: they are clearly time-critical tasks, where message delivery with short delay guarantee is the first design goal. In these use cases including eHealth and safety on road, non-IP technologies are often considered for their reliability [7].
- User-oriented: these applications are non-time-critical tasks, in which category fall infotainment and other prevention on road applications. The use of IP (best effort) to extend the supported geographic area for these applications is possible [8].

The use of IPv6 in current standardization work for vehicular communication technologies guarantees a better integration in the Future Internet. For example, LTE-M technology with a support to IPv6 could open new capillary to infrastructure services perspective [9].

Figure 3-2 shows the overall picture of the joint testbed where the IPv6 link towards the infrastructure is established over Ethernet for early experimentations. If IPv6 is supported over future LTE/LTE-M deployments, experiments would be possible over such networks.

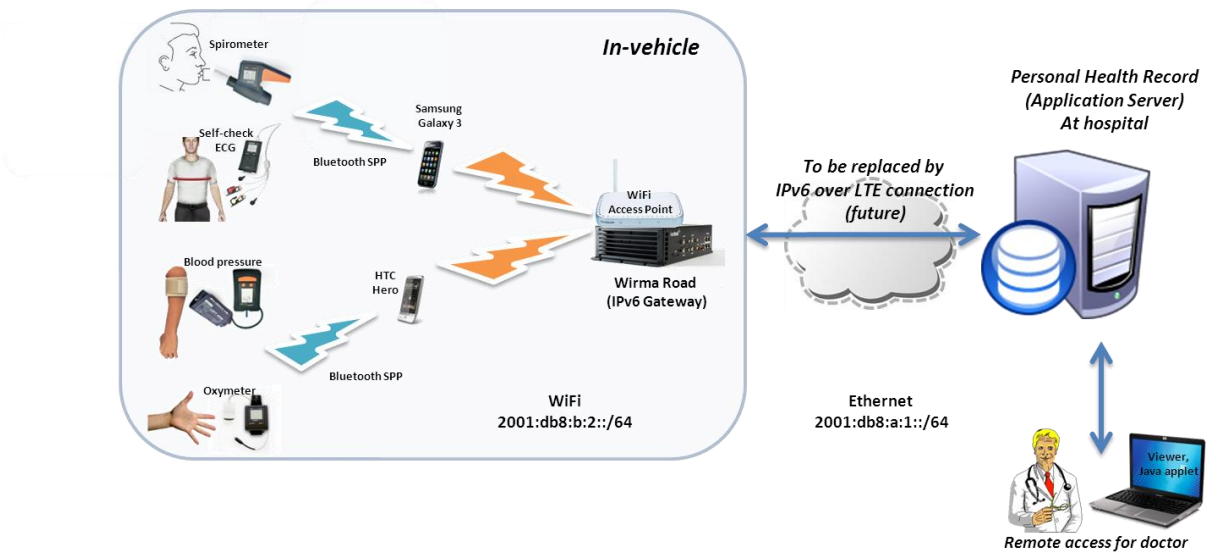


Figure 3-2: Overall picture of eHealth and ITS subtestbed integration

Vehicular networking and eHealth technologies are combined as depicted in Figure 3-2 in the form of an ambulance equipped with special telemedicine devices that can record as well as transmit a patient's vital signs (body temperature, pulse rate, respiration rate, blood pressure) and critical physiological parameters (Electrocardiograph - ECG, blood glucose levels,

oxygen saturation levels) to the nearest hospital in order for the resident health professionals to optimally prepare the patient's admittance. This is a typical Vehicle to Infrastructure communication scenario that can be greatly enhanced through IPv6 connectivity.

In this joint testbed a road accident is assumed where a serious trauma should be taken into consideration. The ambulance crew has in its disposition a set of handheld lightweight devices that can transfer emergency data to the hospital. The objective in such a situation is to maximize clinical value through a limited set of measurements. All involved devices communicate via Bluetooth to an Android smart phone providing IPv6 connectivity.

However in an emergency situation (natural disaster, road accidents) where numerous vehicles of different functions (ambulances, fire brigade, police cars) are involved, the scenario could differentiate in order to accommodate for the optimum data transfer to the interested parties (health care provision, law enforcement) via Vehicle to Vehicle communications. Although this functionality is possible with our V2V2I proposal, the joint testbed (Subtestbed 2.1) will focus on the case scenario where recorded health-related data is sent to the infrastructure for the sake of clarity.

3.1.1.1 Address translation strategy

The small devices (including eHealth devices) forming the Internet of Things (IoT) are generally limited in terms of computing and networking capabilities. Short range communication technologies, such as Radio Frequency Identification (RFID), Bluetooth, or IEEE 802.15.4 standard are much more common than long range communication technologies (3G, LTE or WiMAX). Therefore, an additional functional element, the gateway (GW), translates between both short and long range communication technologies and helps expanding the boundaries of the current Internet. From an addressing perspective, these gateways are called Address Translation Gateways [10] due to their dual addressing function (IP and IEEE 802.15.4 in 6LoWPAN, for instance).

The vehicular eHealth scenario considers IP devices deployed in a vehicle equipped with a Gateway offering long range connectivity. In such a scenario, mechanisms of auto-configuring simultaneously a multiplicity of addresses are needed: the Gateway needs not only one address for itself but a set of addresses for the IP eHealth devices. The mechanism to achieve such auto-configuration is named Prefix Delegation. This is an extension to the Dynamic Host Configuration protocol (DHCPv6) [11].

In addition to the typical functionality of DHCP to assign IP addresses, this extension allows the assignment of a set of prefixes to a Client. The DHCPv6 protocol is specified to work with Relay and Server entities; when assigning a prefix instead of a set of addresses the routing paths need to be updated on Relay and Server. A recent reference [12] describes a mechanism to support Prefix Delegation when the Relay and Server are used.

Prefix Delegation for Network Mobility [13] is a specification of behaviour for the existing DHCPv6 Prefix Delegation such as to work in the context of network mobility. Network Mobility (NEMO) is an extension to the Mobile IP protocol to support groups of devices moving together; such a group can be understood as a capillary network and/or as a vehicular network. This particular prefix delegation mechanism specifies the roles of the Requesting Router (Mobile Router) and of Delegating Router (Home Agent), as well as the placement of the DHCP Relay (Mobile Router).

3.1.1.2 Use cases addressed

Subtestbed 2.1 merges two worlds: eHealth and vehicular communications so the obvious use case addressed is that of ambulances and other public service vehicles especially in a crisis situation e.g. natural disasters etc. However as the basic principle it serves, is that of IPv6 connectivity as long as the capillary network in place is that of Bluetooth it can be used for other scenarios regardless of specific application domain.

The eHealth protocol messages carry sensible data and require integrity, confidentiality and availability. Privacy is one of these security issues and has been addressed by proposing pseudo-anonymization of medical data [14].

The use case chosen for the specific subtestbed only for reasons of simplicity as well as clarity it allows for the recording of vital clinical parameters of an individual on site. These parameters are then transmitted in a secure way and via the vehicle's set up to the nearest hospital thus allowing for the optimum preparation when expecting a trauma patient. This is a typical Vehicle to Infrastructure scenario (V2I) and while in the considered ambulance setting the customary devices to be found are only two, namely an electrocardiograph and an oxygen saturation level meter, in our case we have opted for a greater variety of devices in order to demonstrate the connection capabilities of the set up.

3.1.2 Subtestbed 2.2: Heterogeneity and interoperability

An intelligent M2M gateway component for LTE-M networks to support communications amongst M2M devices, high-level application and service layers is developed. The M2M gateway design and architecture in particular focus on three main aspects: connectivity of heterogeneous resources, data processing and optimization of access and resource usage for resource/power constrained devices, provisioning and interaction with network and service layers.

The planned architecture for the M2M gateway is in line with the testbed design and scenarios that were selected for the testbeds. The communication between the M2M gateway and capillary networks and intelligent decision making within the components will enable the EXALTED components to provide connectivity and interoperability for the underlying heterogeneous networks and at the same time optimize and manage resource access for constraint devices using context-aware and intelligent machine learning mechanisms.

The current M2M gateway software supports the integration of 802.15.4 enabled devices in the capillary network. The M2M gateway groups the nodes in the capillary network according to semantic context information, which corresponds to different QoS requirements, to reduce the communication overhead. Two main features are introduced, namely data dissemination and data aggregation. The data dissemination feature utilizes the semantic context information to distribute messages in the capillary network to the suited sensor nodes. The data aggregation feature aggregates data of nodes within a group for message-reduced data provision. The software runs on Advanced RISC Machine (ARM) and x86 devices and is ported for the Sagemcom On-Board Unit (OBU).

3.1.3 Subtestbed 2.3: Connectivity for low-power devices

Due to the fact that devices in capillary networks are constrained in resources, they usually operate using different addressing schemes depending than the ones used for traditional IP communications. Meanwhile LTE-M network will handle IP addresses, either IPv6 or IPv4. As one of the key points of the EXALTED project, it is necessary to provide continuous connectivity among nodes, outside and inside the capillary networks. This duality requires the definition of mechanisms that allow the following:

- Accessing nodes within a capillary network from external nodes.
- Accessing to public servers from the capillary network.

Up to day, there is some research focused on how to provide E2E connectivity between nodes in the capillary networks and out of it, on one side IP and on the other one ZigBee or similar. The most common solution is the implementation of a web server that maps ZigBee addresses into ports in the IP interface of the gateway node. Recently, ZigBee Alliance has proposed a methodology based on Extensible Mark-up Language (XML) Remote Procedure

Calls (RPC) aiming at providing continuous connectivity from the IP networks to capillary [15]. Due to the particularities of M2M communications analyzed in EXALTED both schemes are being investigated in order to select the one that better fits in the project framework.

Figure 3-3 depicts how this translation is done at the gateway node for the web server approach.

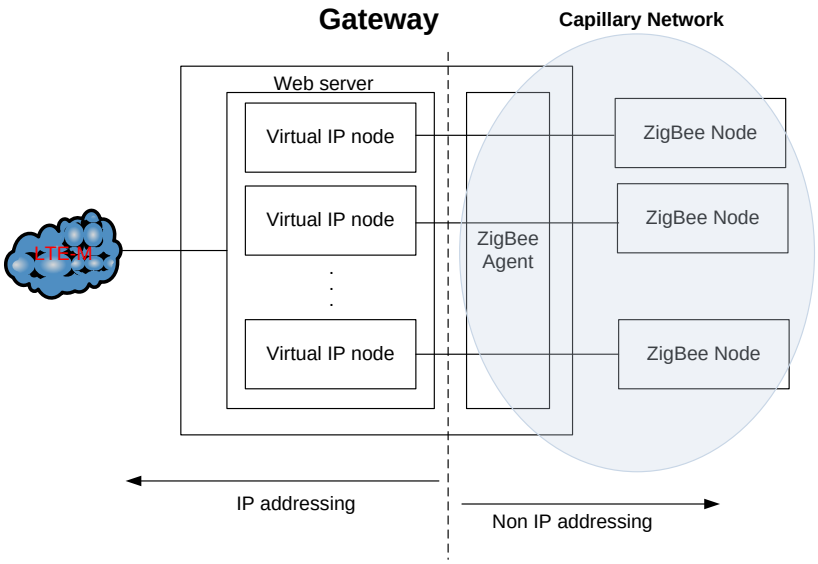


Figure 3-3: Address translation between LTE-M and capillary networks

The resulting mechanism that is described in section 3.3.3 aims to provide the same functionalities identified on the previously mentioned literature, while reducing as much as possible the overhead and resource consumption at gateway level.

In order to be able to properly study which is the best option, it is important to describe firstly all elements and technologies involved in the path from Non-IP devices to elements placed in the core IP network.

Adapting the elements to fit into EXALTED architecture, and looking always for a solution suitable to be implemented and measured over real equipment available in the project, the following technologies have been identified as key for the definition of the algorithm:

- Near Field Communication (NFC). It is not directly related with the address translation mechanism, but it is used as user interface. The way to initiate and force communications from devices is done by NFC tags. This way, the amount of data transmitted from nodes is tightly related to this standard, so it is depicted in section 3.3.3.
- 802.15.4 standard, responsible for the communication between end nodes and the gateway in charge of the address translation. It is necessary to study the format of frames related to this standard, so as to be able to parse them and then translate it into IP packets on the gateway, and vice versa. This study can be seen in section 3.3.3.Over this standard, DigiMesh protocol will be implemented. This protocol is a proprietary approach embedded on 802.15.4 radio modules manufactured by Digi. It is very interesting for testing new algorithms as it considers each node in the network as an equal, without distinguishing among profiles as in ZigBee.

More details about both standards and its application to the address translation mechanism can be seen in [16].

Use cases addressed:

The mechanism presented in Subtestbed 2.3 is suitable to be included into any of the scenarios identified by EXALTED project, namely ITS, Smart metering and monitoring and e-Health, although it is more suitable to be implemented for SMM applications.

There is a strong need for all of them (which is derived from a main objective of the EXALTED project) of addressing high number of devices behind the gateway. This need, combined with the fact that Non-LTE-M devices are constrained in terms of computing capabilities and battery, and with the need of seamlessly addressing in a unique way each device in the network, makes the address translation mechanism a key algorithm to prove the EXALTED premises.

In an attempt to give more visibility to the proof of concept performed in this testbed, one particular use case has been chosen among the plethora described in the project. It maps a real world procedure that is being addressed now around the world: automation of hospital logistic procedures.

Current logistics systems in hospital environment are usually based only on large dedicated equipment [17]. In the recent years, some level of automation has been introduced enhancing overall system performance [18]. However the advances in technology allow a higher degree of autonomy for such systems. Scalability has been stated as a key issue to be addressed by these systems and the best option is extract all the power for smart devices [19]. The combination of the potential of all existing solutions led us to create a full system integrating multiple communication technologies for different purposes, in a smart low-cost device.

Nowadays taking advantage of the efforts done in the field of M2M communications [20] and the concepts proposed by IoT, it is possible to create a scalable system based on smart devices. The possibilities offered by the combination of both concepts have been exploited in the system we are proposing.

3.1.4 Subtestbed 2.4: End-to-end security

Together with the end-to-end connectivity demonstrated by other subtestbeds the purpose of this subtestbed is to demonstrate end-to-end security. It is shown in Figure 3-4. Among all scenarios identified within the project this subtestbed explicitly illustrates the Smart Metering and Monitoring scenario with photovoltaic panel hosted by devices and Kilo Watt / hour (KW/h) values captured by devices and sent to the servers.

Two use-cases are demonstrated in the subtestbed. In the first use-case smart metering data captured by devices are put in an application payload that is protected against unexpected updates and then sent to the server. Devices can be either LTE-M devices with connection with the Radio Access Network or non LTE-M devices in a capillary network. Different secure elements are used according to the device type. Assumption with this subtestbed is that keys used to protect the payloads have already been deployed and are ready to be used.

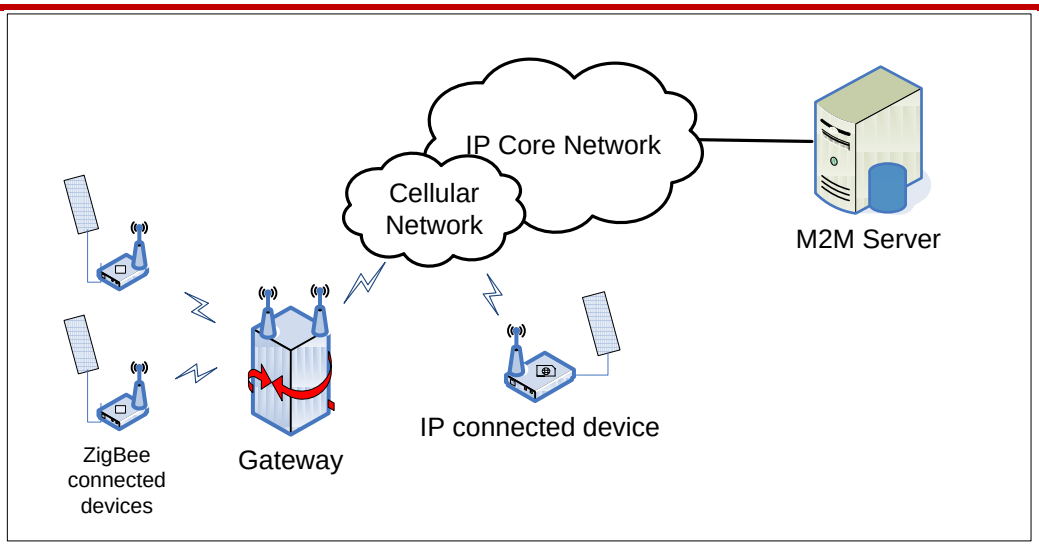


Figure 3-4: Components of Subtestbed 2.4

With the second use-case it is the server that sends a command to devices. For various reasons the server may ask some devices to stop generating electricity either because the electric network is near overloading or because of security reason like a fire alarm in the installation not more electricity must be produced. After the alert the server can ask devices to restart production again.

Only data integrity is used in this subtestbed because it is the most adequate security protection with the smart metering scenario. There is no real need to cipher data exchanged between the server and the devices because the data transmitted are not very sensitive. Both ends have interest to tamper with metering data: on the energy producing end it would be nice to artificially increase the amount of electricity produced while the electricity operator could be tempted by under evaluating the amount of produced electricity. In the real world electricity dissipation should also be evaluated and integrated but this is out of the scope of the project.

Integrity is provided by hashing the application payload and then ciphering the resulting hash value with a secret key. The result is called a Cryptographic Message Authentication Code (Crypto-MAC). It has nothing to do with the Media Access Control address denoted with MAC in all documents of the project. It is just a similar acronym. A Crypto-MAC is similar to a signature in a way that a message is hashed and then this hash is encrypted with a key. Difference is that a symmetric key is used to encrypt the hash and with this regard is different from a digital signature where is must be the private key to be used to sign the hash. Advantage of the signature is that in addition of the data integrity it also provides the authentication of the peer that signed the data and the non-repudiation. The drawback is that it imposes that the secure element is able to provide asymmetric cryptography that is more energy demanding and needs a more expensive chip.

In both use-cases of this subtestbed the message flow is initiated by the M2M server with an SMS. There is no other way for the server to reach devices using TCP/IP packets because devices have private IP addresses in the operator domain and cannot be reached from the IP core network without an initial request from the devices. In this initial SMS message the M2M server gives its IP address and then IP packets can be exchanged with always first packets sent from devices to the M2M server.

For this subtestbed the capillary network is a ZigBee network.

The data protection used with this subtestbed does not address the integration with the data aggregation algorithms implemented in the project. Only end-to-end security is covered here.

3.2 Components and interfaces

This section gives a brief and high level description of the main components and interfaces of the testbed. It is not the intention to provide a complete specification of the testbed. The reader shall get an impression about the main functional blocks and how they are linked together.

3.2.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario

This subtestbed assumes the existence of an underlying core network infrastructure supporting IPv6 communications and thus allowing the establishment of an IPv6 path from the M2M Gateway (first endpoint) to the Application Server (second endpoint).

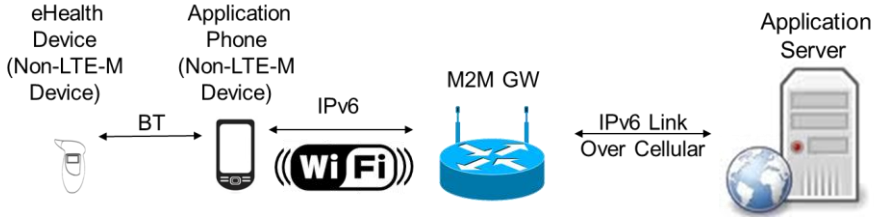


Figure 3-5: Subtestbed 2.1 high level architecture

Figure 3-5 describes the overall system architecture of Subtestbed 2.1 highlighting the LTE-M and Non-LTE-M functional elements of the testbed. These elements are located in the in-vehicle network. The vehicle can be Leaf Vehicle (LV) or Internet Vehicle (IV) depending on the case study. For our experiments, we use a backward compatible LTE/3G dongle on our M2M Gateway to validate the testbed requirements.

- **Non-LTE-M devices**: There are two types of Non-LTE-M devices in this testbed: eHealth devices and Application Phone. The eHealth device provides real time health-related measurements. These measurements can be of different nature such as blood glucose levels or oxygen saturation levels. These M2M devices are provided with Bluetooth communication technology to send recorded data to another authorized peer. The Application Phone is in the middle of two different communication technologies. On one hand, short-range Bluetooth technology to communicate with M2M Devices and capture the eHealth data and on the other hand, short-range WiFi technology to send IPv6 TCP packets to the server via the Gateway. The application on the phone allows adding comments along with the data to send (transferred to the server as another type of application payload).

  To achieve heterogeneity objective of EXALTED, two radio technologies are supported in the capillary network (inside the vehicle): Bluetooth and WiFi (IEEE 802.11b/g/n).

- **M2M Gateway**: Also called M2M Gateway, or Mobile Router depending on the context. The main functionality is to provide IPv6 connectivity to in-vehicle devices and a default-route through cellular network towards the server in the IPv6 Internet. As energy-consumption is not a critical criterion for our testbed, we demonstrate the use of two types of M2M gateways. The first gateway is energy-efficient and of limited power. This gateway is used in the V2I scenario of eHealth. The second gateway which is more powerful and more energy-consumptive than the first, implements V2V2I algorithm and allows the demonstration of eHealth over multi-hop topology, involving inter-capillary networks communication, and capillary-to-infrastructure communication. With an LTE/LTE-M technology deployment, both gateways use LTE-M cellular interface as a long-range communication interface. The gateway implements DHCPv6-Default Route to provide a default route to in-vehicle devices, and V2V2I algorithm to provide cellular coverage for the neighbouring vehicles when they are not provided with an LTE-M interface.

This part of the testbed will achieve the EXALTED objective of addressing and connecting numerous devices behind the gateway. These devices should also be accessible from outside by the use of Global IPv6 addresses.

- **Application server**: In our demonstrator, the Application Server collects the data from patients and provides a web interface for doctors to support diagnostic. The software running on the server includes a web server accessed over a secure connection (Secure Socket Layer - SSL) and a limited-access database server to gather the data by patients. A Java Applet is required to view ECG graphs on the doctor screen. The server must be IPv6-capable and use a global IPv6 address. For the sake of simplicity, a Fully Qualified Domain Name (FQDN) should be used and resolved to the actual IPv6 address of the server by a DNS entity, and the entry should be updated whenever the server address changes. This assures that the Application Phone is always able to access the server using its FQDN.

  The server is connected to the IPv6 core Network. For the eHealth V2I proof of concept purpose, we used a remote Ethernet connected server on the M2M GW.

Basic operation

With respect to IP, each device within the capillary network may have an IP address assigned to it, although it is not mandatory. In the ITS and eHealth scenario, it is mandatory that at least one device within the capillary network (not the Gateway) to be assigned at least one public IP address using the IP auto-configuration mechanisms. So far in our proof of concept we used the application phone as a proxy for actual M2M eHealth devices. The application phone is thus the one having a complete IP configuration on behalf of other devices.

The Gateway must be assigned at least two IP addresses (or prefixes): one for each of its interfaces (minimum two interfaces). The Gateway performs IP forwarding functions, and must decrement the Hop Limit field of the base header of an IP packet when forwarding packets to/from the capillary network. Within a capillary network there may exist one or several distinct IP subnets. In the Subtestbed 2.1, only one subnet is considered.

We can split the basic operations happening in Subtestbed 2.1 into 3 types: Auto-configuration mechanisms, Server Registration, and Server Communication.

The Subtestbed 2.1 includes a set of auto-configuration mechanisms that provides M2M devices into the capillary network with a correct IPv6 configuration. These mechanisms are important to achieve some of the EXALTED objectives:

- Support of a large number of devices: by configuring them with a Global IPv6 prefix, we are able to support up to (in theory) 2^64 device per prefix used.
- Support of heterogeneous networks: this testbed integrates Bluetooth communicating devices with a WiFi and LTE-M M2M GW.
- Autonomous IPv6 parameters configuration: the internal devices configure themselves with correct IPv6 parameters upon the reception of M2M GW Router Advertisements.

A basic IPv6 configuration includes IPv6 prefix and a default route to global Internet. Devices with limited CPU and memory capacities can benefit from the sole presence of a default route in their routing tables: it is sufficient to store *only* the default route in order to be able to reach any other node in the Internet. In this sense, the default route is a very strong candidate for implementation in small devices (it is possible to not store the other routes). Using a default route instead of a large number of specific routes helps keeping routing table sizes small, which is especially advantageous for machine-type communications.

For this reason, it is important to have a good mechanism of assigning default routes to end nodes: M2M Device and M2M Gateway (Gateway can be considered as an *almost*-end node: it is situated one or a few IP hops away from the end). Recent works at IETF propose extensions to the DHCPv6 protocol to provide default route to a M2M Gateway. A preferred such method of assigning a default route is described in Figure 3-6.

Figure 3-6 depicts the overall picture of a capillary network connected to the LTE-M access network to obtain the default route.

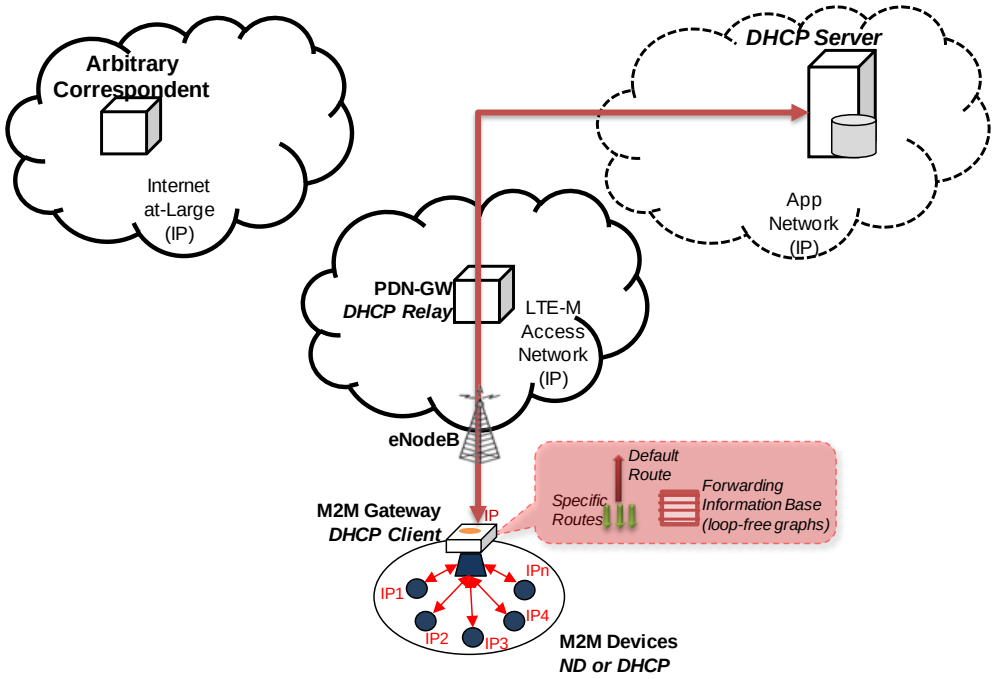


Figure 3-6: Topology for default route assignment within the EXALTED architecture

Figure 3-6 shows the following entities:

- An arbitrary correspondent node situated in the fixed Internet infrastructure; at the leftmost upper corner. It represents an IP server which cannot be modified.
- The DHCP Server situated in the Application Network of the EXALTED (rightmost upper corner); this server provides with DHCPv6 default route among other possible parameters.
- DHCP Relay, co-located with the Packet Data Network Gateway (PDN-GW) (gateway typical in the LTE architectures), situated in the joint LTE/LTE-M core network; this represents the first IP hop when looked at from the DHCP Client.
- M2M Gateway which controls the capillary network, and which moves together with it. The M2M Gateway runs software implementing DHCP Client functionality.
- eNodeB situated in the LTE-M network, represents the radio receiver for the radio emitter present in the M2M Gateway.
- Nodes within capillary network are named “M2M Devices”. In ITS-eHealth testbed, the Application phone and eHealth devices are considered capillary M2M Devices.

In order to self-configure the M2M Gateway, it needs to run a protocol to obtain the following parameters:

- An IPv6 route to be used as a *default* route in the routing table of the M2M Gateway.
- A set of IPv6 prefixes to be used for address auto-configuration on the M2M Devices.

The details of DHCPv6 extension to assign the default route are described further in section 3.3.1of this document.

Figure 3-7 and Figure 3-8 show the M2M Gateways on which the above proposals are currently implemented.

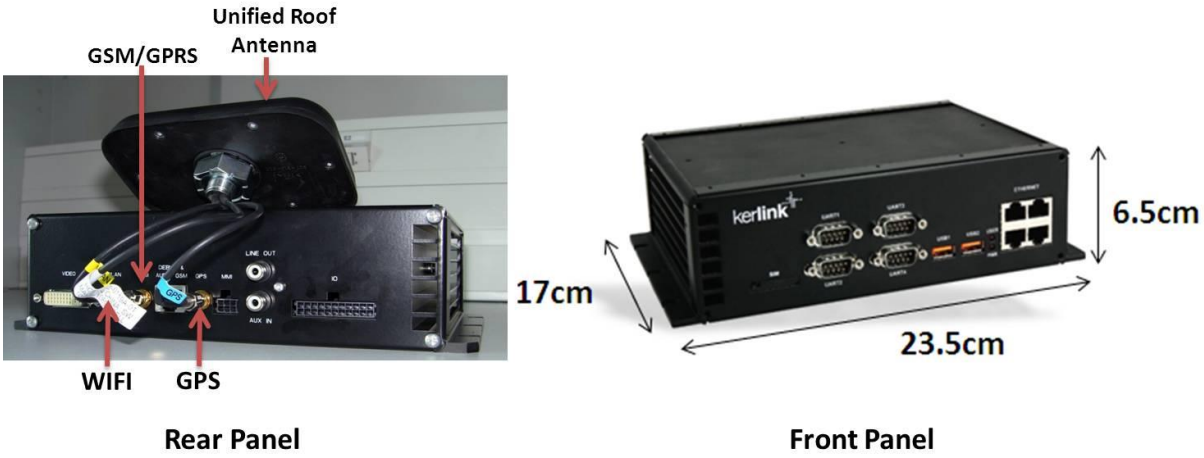


Figure 3-7: Energy-efficient M2M GW for vehicular communications

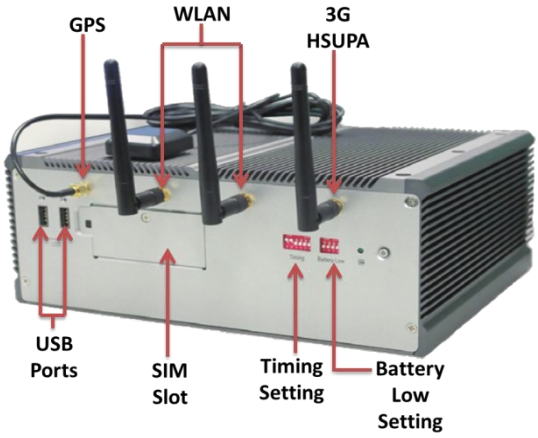


Figure 3-8: Powerful M2M GW for vehicular communications

Figure 3-9 represents the registration process of eHealth devices on the Application server. TCP messages are sent from the Application phone to the Application server through the M2M Gateway. The messages consist of two types: Control and Data. The control plan includes registration messages sent from the Phone to the Personal Health Record (PHR) Server to acquire the user ID. The PHR server acknowledges by sending the requested information to the phone. This information is sent over IPv6 and traverses the M2M Gateway attached to the cellular infrastructure.

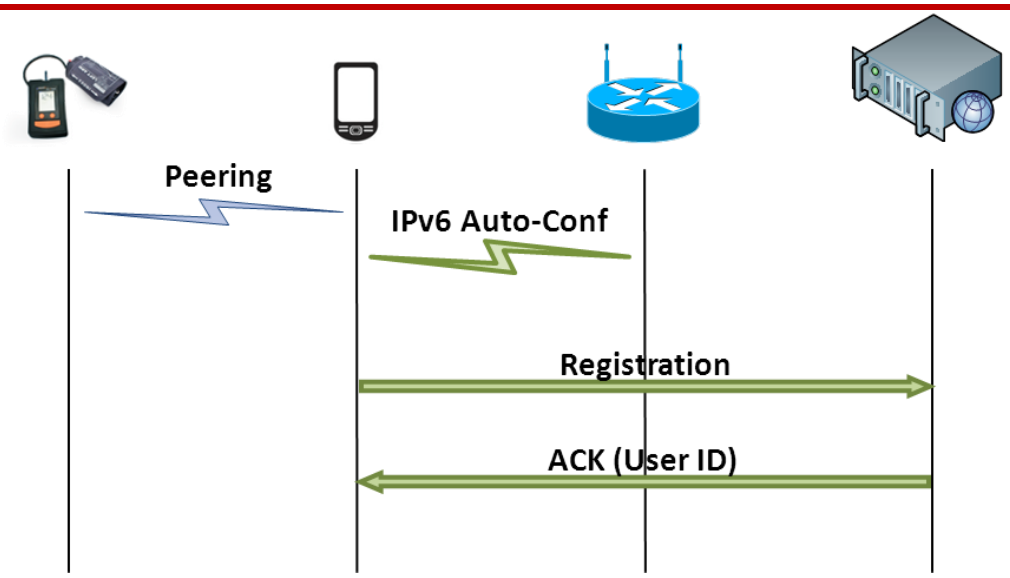


Figure 3-9: eHealth devices registration process

Figure 3-10 depicts the communication process (that is the data plane) involving the eHealth device, the Application phone and the Application server through the M2M gateway. The eHealth recorded data at the devices level are first gathered at the phone level, and then sent to the server traversing the M2M Gateway. After reception at the server level, the remote access of a doctor is possible. This is to issue a diagnostic over the recorded data basis.

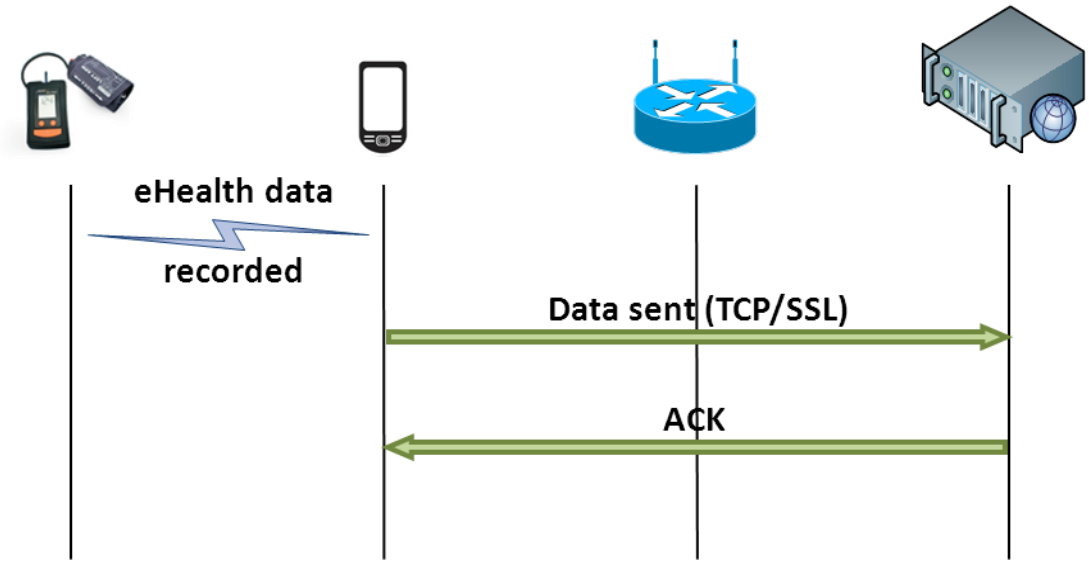


Figure 3-10: Server communication

3.2.2 Subtestbed 2.2: Heterogeneity and interoperability

The M2M gateway is implemented by three main functions as shown in Figure 3-11. The **data collection** is responsible to make the connection to the capillary networks and provide the data to the other functions. The **data processing** processes the data; in our approach we introduce two main processing features, data dissemination and data aggregation. Both rely on the context database which stores semantic information, about the sensors and their relations, in an ontology. The **data provision** is the interface between higher network services and the processed data by the M2M gateway.

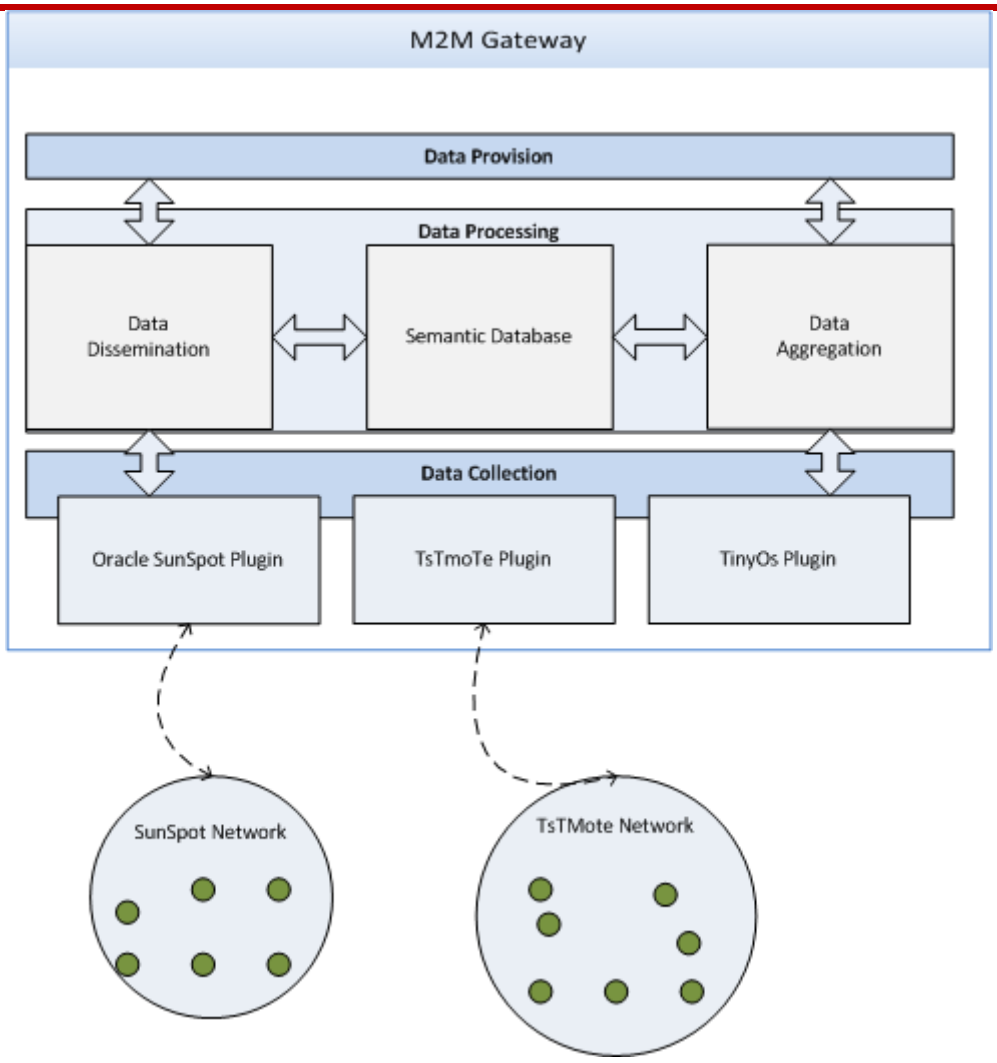


Figure 3-11: Overview of M2M Gateway architecture

3.2.2.1 Data Collection

The data collection function establishes a connection between different sensor platforms and makes the sensor data available to the other, higher M2M gateway functions. The sensors can communicate via multi-hop connections to the sink node connected to the M2M gateway. The M2M gateway (i.e. the OBU) implements different plug-ins according to the different hardware and software features of the used sensor platforms to communicate and process the data from the sensors. The data collection is designed in a modular plug-in approach. In the case that new sensor hardware or software commences, new plug-ins can be included to integrate the platform into the OBU gateway. In the current designs, plug-ins for the TsTmoTe, Oracle Sunspot and TinyOs devices are envisaged. We have developed a zero-configuration approach, to discover sensor nodes immediately after the devices power up or get in range of the gateway's beacon signal. This feature is described in the following section.

3.2.2.2 Data Provision

The data provision function is responsible for making the connection between the M2M gateway and higher network services and applications. In this function several interfaces can be introduced to make the data gathered by the sensors and processed by the gateway available to others. This function has to provide simple programming APIs to be accessible for developers but also standardized interfaces such as commonly used Web-Service (WS-*)

interfaces such as RESTful and WS-* based components used in M2M communications. This function is designed similar as the data collection function, plug-ins for different interfaces and technologies can be neatly attached.

3.2.2.3 Data processing

The proposed M2M gateway supports processing of the data gathered by the heterogeneous capillary network. The novelty of this design is the usage of a semantic context database, which stores not only the sensor information from the M2M device but also context information such as location of the sensor, observed feature of interest and other meta-information related to the sensors. This cross-linkage of data is exploited in two data processing mechanisms namely data dissemination and data aggregation algorithm.

3.2.2.3.1 Semantic Database

The semantic database represents the cross-linked Meta information of the capillary network. The information ranges from sensor platform information such as which sensor uses which hardware and provides which sensor devices up to the information where the sensors are deployed and what their current feature of interest (topic) is. To map the real properties of the network to a semantic description the W3C Semantic Sensor Network Ontology is used. In Figure 3-12 an extract of the used ontology is shown. The information is stored on gateway level and maintained each time a sensor connects to the gateway as described in [21].

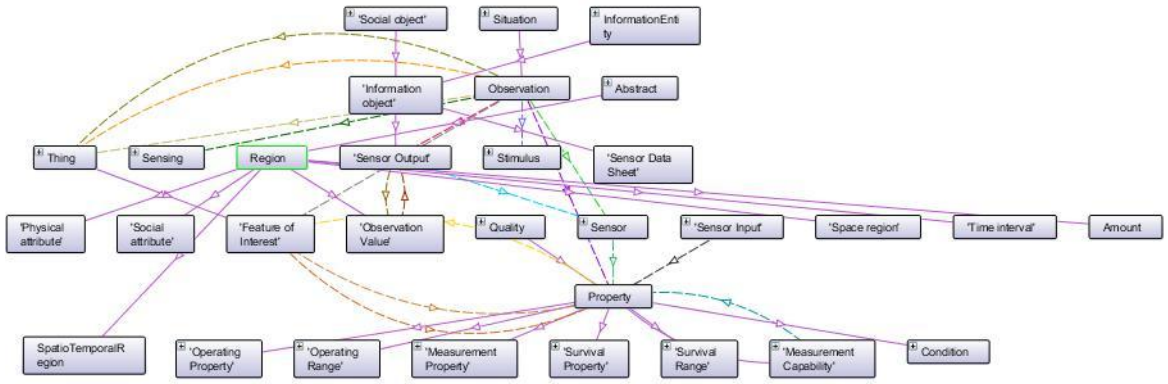


Figure 3-12: Extract of the semantic model

The semantic database is the foundation of the proposed features of the M2M gateway which are introduced in the following sections.

3.2.2.3.2 Semantic Data Dissemination

To notify the capillary network about changes in the topology or to traverse the network to retrieve certain information, flooding algorithms have been used in the past. In our work we introduce a smart gossiping algorithm based on the semantic database and the cross-linked information network of the whole network. Instead of traversing or flooding the whole network, a query can be limited to semantic-related sensor nodes which are grouped according to the context model.

3.2.2.3.3 Semantic Data Aggregation

The data gathered by the sensors in the capillary network are usually transmitted as raw data. Depending on the sampling rate of the sensors, the transmission is energy and process

intensive. We introduce a new approach based on the Symbolic Aggregate Approximation (SAX) [22] linked with our semantic model to abstract from raw time-series data to transmit abstracted information compressed from sensor to gateway and from aggregated gateway to higher-layer services.

3.2.3 Subtestbed 2.3: Connectivity for low-power devices

Apart from the underlying infrastructure assumed about LTE-M and core IP network in charge of making possible to interconnect LTE-M enabled gateways and any server connected to the Internet, this subtestbed defines the behaviour and needed programming on three kinds of devices as presented in section 3.1.3. Since LTE-M equipment is not available, the cellular network used in this subtestbed is GPRS. According to EXALTED notation in the report D2.3 [23], these devices are the following ones:

- **Non-LTE-M devices**: constrained devices in terms of battery, cost and capabilities. They are in charge of performing simple sensing or acting task, and reporting their status to the gateway (maybe through multiple hops over mesh networks or through CH devices aggregating traffic).

  Aiming to prove one of EXALTED's goals, several radio interfaces have been placed in order to show heterogeneity: NFC is included for interacting with the devices and 802.15.4 is available for wireless communication of data to the gateway.

- **M2M Gateway**: A more powerful and intelligent device, in charge of handling communications with the outside world using a cellular interface. One of its main functions is mapping devices behind the gateway into virtual IP nodes reachable from the IP core network.

  The interfaces placed in this device are 802.15.4 to communicate inside the capillary network, and GPRS interface for sending information from Non-LTE-M devices to the outside world.

- **M2M Application server**: This is the machine where devices report their information. Its features are strongly dependent on the use case selected. This subtestbed deals with hospital logistics, so its main activities are tracing medicine stock, monitoring battery status of devices and trigger alarms.

- It is accessible through a public IP address, so the communication with M2M gateway can be established directly.

Basic operation

There are two main types of communication regarding the address translation mechanisms: One related to device initialization and the other for the normal operation.

It is important to note that the way it is conceived, integrating NFC capabilities, pursues two big EXALTED goals:

- On the one hand, it proves heterogeneity on the technologies handled by the network.

- On the other side, it gives visibility to the operations performed. Both initialization and sending of the data could be programmed on nodes, but this way it can be shown on demand. This way anyone can test the correct behaviour of the algorithm at a certain moment.

Registration is the first task needed to be performed on the devices. If it is not done, no other operations are enabled by the system.

Once the INIT device is powered on, it sends an 802.15.4 message to the gateway containing the IP address of its default application server. This info is translated into an IP datagram and transmitted to the application server through a new socket session started for

this purpose (further considerations about the behaviour of the gateway are treated in section 3.3.3.2). The M2M Server then registers the node and sends back the confirmation, firstly to the gateway and, then, to the node, as shown in Figure 3-13.

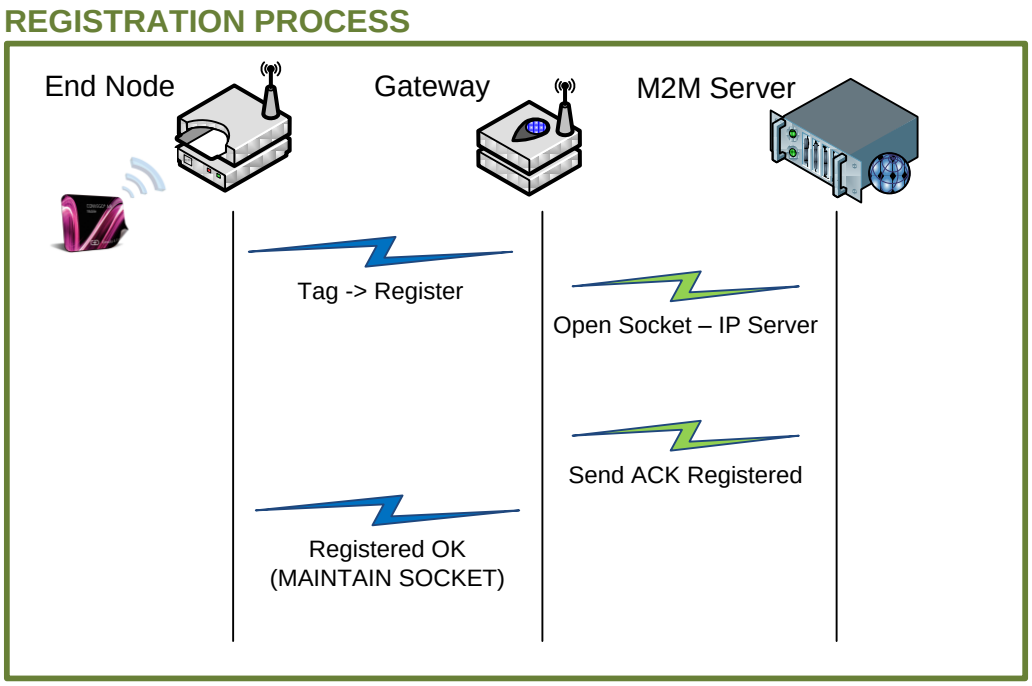


Figure 3-13: Registration process

Once the node is registered, any tag can be approached to the nodes, resulting on a different response from the server. The info is sent via 802.15.4 air interface to the gateway which capsules it into IP and sends it towards the application server using the previously created socket connection. The same way, the response is routed back to the node, as can be seen in Figure 3-14.

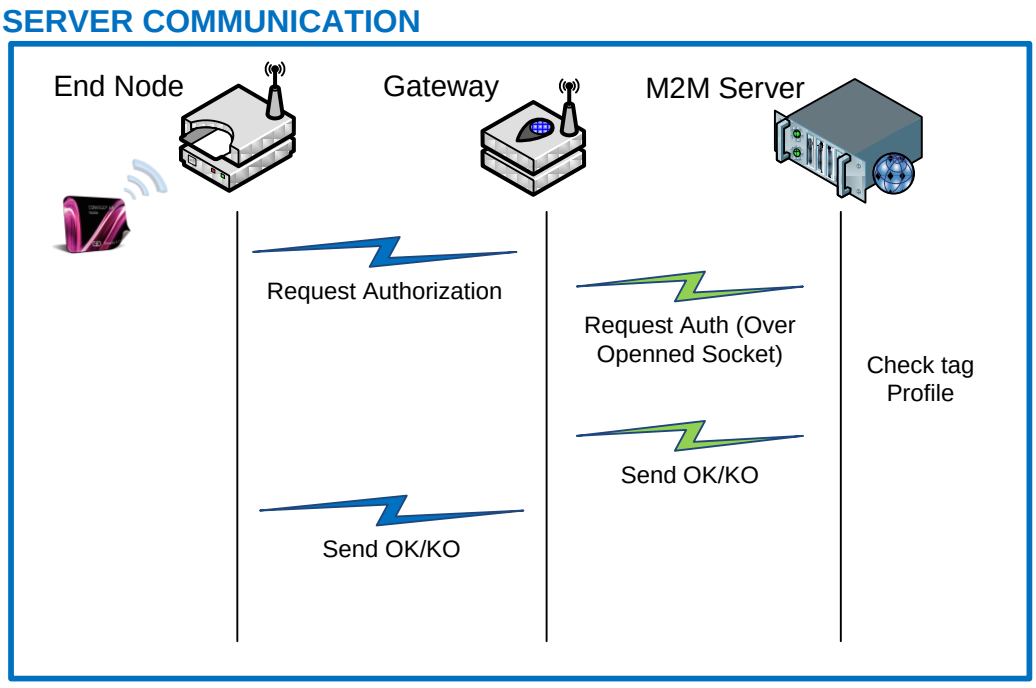


Figure 3-14: Server communication

There are two ways of transmitting information through DigiMesh protocol: transparent mode and API mode. The one selected for this application is API mode.

When operating in transparent mode, the modules act as a serial line replacement, whereas API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a Universal Asynchronous Receiver Transmitter (UART) port.

The frame structure for API mode can be seen in Figure 3-15. Note that the frame varies depending whether it is sent or received.

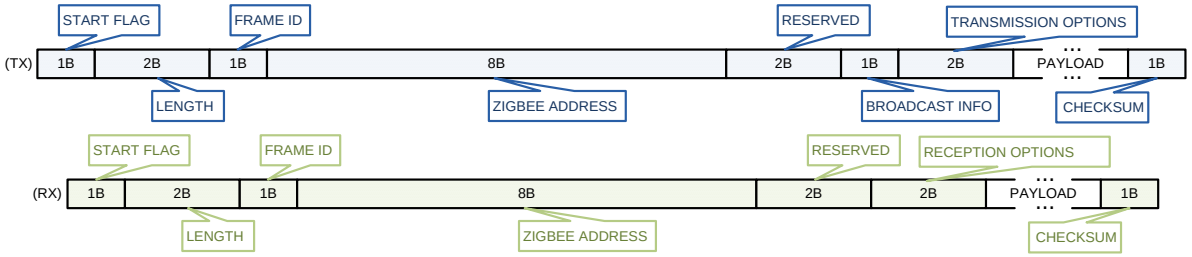


Figure 3-15: DigiMesh frame structure for transmission and reception

The payload data included on DigiMesh fames, depending whether it comes from the nodes (Tx) or from the M2M Server (Rx) is pictured in Figure 3-16.

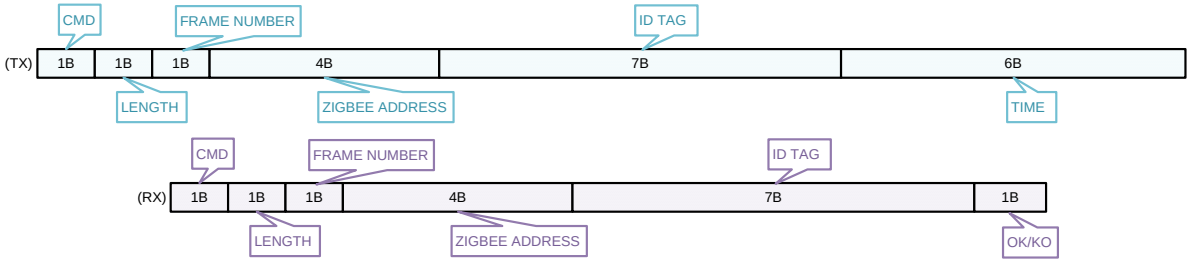


Figure 3-16: Payload structure

With this architecture in mind, the only thing left to define is the specific behaviour programmed on each element of the architecture (please refer to section 3.3.3).

3.2.4 Subtestbed 2.4: End-to-end security

Figure 3-17 shows the components and the two different types of devices of this subtestbed.

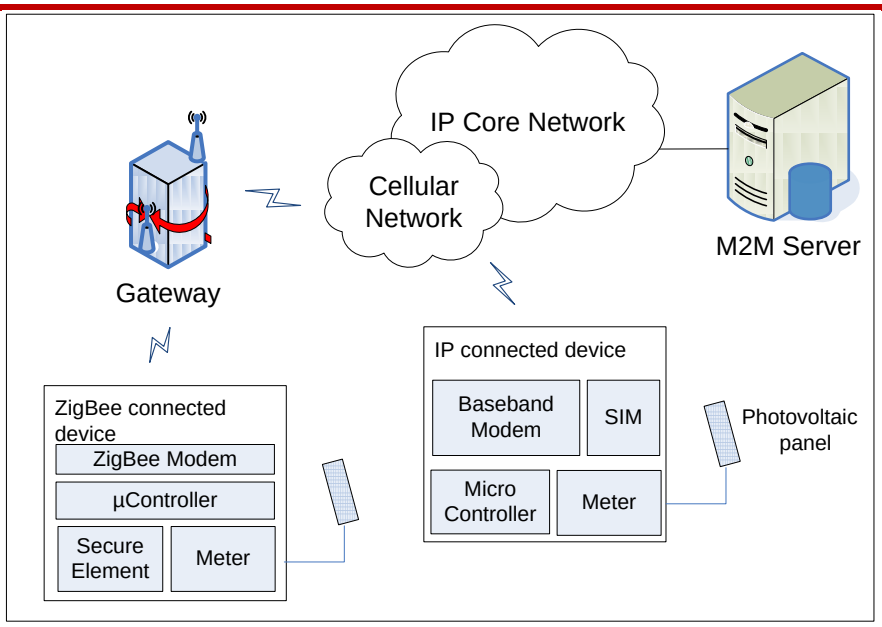


Figure 3-17: Subtestbed 2.4 components

Using the EXALTED notation in the report D2.3 [23], the following components are involved in this subtestbed:

- **Non-LTE-M device**: This device communicates with the gateway through 802.15.4 wireless communication. It hosts a ZigBee radio Frequency chip, a photovoltaic panel, an electric meter, a serial connection with the EXALTED Secure Element, a relay that monitors the electric connection to the photovoltaic panel and a controller.
- **EXALTED Secure Element (SE)**: it is a bi-chip module that embeds a configurable bridge and a Machine Identity Module (MIM). This latter component is described below. The SE is accessible through a Serial Peripheral Interface (SPI) bus in a master-slave mode. The purpose of the configurable bridge is to adapt the communication protocol between the MIM and the device. Configuration of the bridge is set at boot time from the flash memory of the MIM.
- **Machine Identity Module (MIM)**: This is a classical Subscriber Identity Module like the one used in mobile handset except there are no ISO contacts.
- **LTE-M device***:* Because there is no way to connect to a real LTE-M network the security is demonstrated with a GPRS/3G modem capability. The modem embedded by the device cannot communicate over LTE. This device also hosts a photovoltaic panel and a similar configuration as the Non LTE-M device except that the security features are provided by the SIM card hosted by the device to enable wireless connectivity**.**
- **LTE-M Gateway**: This device shown in Figure 3-18 is able to communicate either with the MNO wireless network or with the non LTE-M devices through IEE 802.15.4. A very basic address translation is implemented in this gateway only for the purpose to demonstrate the security. The gateway is made of two Systems on Chip (SoC). The ZigBee part is a Texas Instruments component and the GSM part is provided by Cinterion. These two components communicate through a serial line.
- **M2M server**: HTTP server located on the IP core network. SMS messages are sent to devices using an SMS gateway. The server is able to serve request from devices and also to answer to user request using a business oriented graphical user interface.

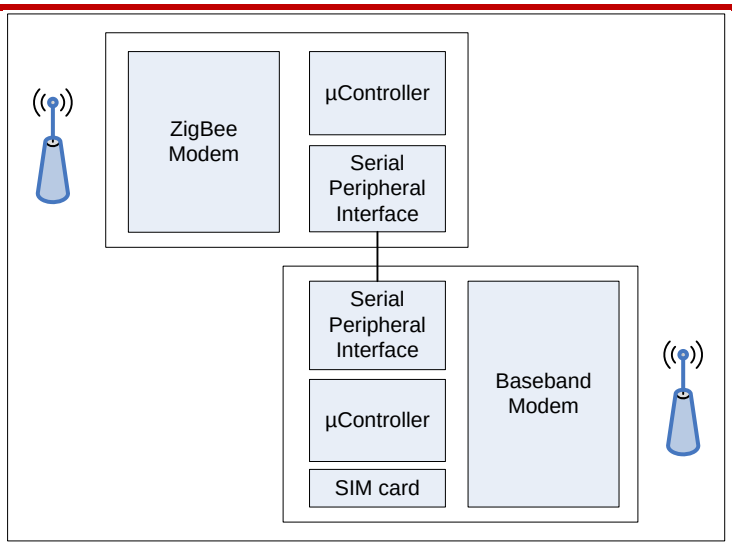


Figure 3-18: Subtestbed 2.4 Gateway

3.3 Algorithms and features

This section shall give a complete description, what exactly can be demonstrated or validated with the testbed, an overview of algorithms and features that can be shown together with a justification, why this is an added value for the project

3.3.1 Subtestbed 2.1: Connectivity for a combined ITS and eHealth scenario

This section relates the technical details of the high level features described in subsection 3.2.1. The technical contribution is presented according to the different requirements on each device.

- **Non-LTE-M devices**
    - eHealth devices record and transmit the patient's vital signs (body temperature, pulse rate, respiration rate, and blood pressure) and critical physiological parameters (ECG, blood glucose levels, oxygen saturation levels) to the application phone they are peered with.
    - The application phone should be configured to handle the eHealth device messages and transmit the data to the Personal Health Record (PHR) Server in the Infrastructure traversing the M2M GW. The phone is set up with a complete IPv6 configuration and uses the M2M GW as a default route.
- **M2M Gateway**
    - It is in charge of setting the M2M Devices with a valid IPv6 configuration:
        - The IPv6 global prefix can be obtained from the infrastructure with DHCP-Prefix Delegation and announced in the capillary network with router advertisement.
        - In the V2V2I scenario the M2M GW should implement Prefix delegation with neighbour discovery as described in the EXALTED project report D4.2 [16].
        - The gateway should implement DHCPv6-Default Route as described previously.
    - It is in charge of the reliable delivery of messages to the M2M Server.

- **M2M Server**
    - It must be accessible from a global location and IPv6 capable.
    - The application must run over IPv6 sockets.
    - It must store PHR containing the history of a patient.
    - It must allow remote access from a health care specialist to issue diagnostics.

3.3.1.1 Non LTE-M devices

The eHealth devices shown in Figure 3-19 used for the testbed are manufactured by Card Guard [24]. The oxygen saturation level is measured by OxyPro, a wireless pulse oximeter. It provides for real time measurements and can be operated in continuous mode. It also provides for pulse monitoring. It displays oxygen saturation and pulse rate averages with the absolute maximum and minimum measurements.

The blood glucose and pressure measurement is performed by Easy2Check [24] device. Blood glucose is measured with the use of an amperometric biosensor where fresh capillary blood is deposited. Its accuracy is for both measurements performed of high level similar to the non portable devices.

Self-check Electro-Cardiographer (ECG) [24] offers 1 to 12 leads ECG events monitoring. It is intended for monitoring symptoms that may suggest abnormal heart function: skipped beats, palpitations, racing heart, irregular pulse, faintness, lightheadedness, or a history of arrhythmia. The recording period is set at 32 seconds while the bandwidth is 0.05 - 35 Hz for the 12 Leads and 0.4 – 35 Hz for the 1 Lead.

Spiro Pro [24] is a spirometer that records exhalation air flow curves (volume over time) according to international performance standards. It measures lung ventilatory functions during Forced Vital Capacity (FVC) tests. The recording lasts for 17 seconds and its accuracy for the FVC and FEV 1 is +5% or +0.1L. It is mostly used for asthma or Chronic Obstructive Pulmonary Disease (COPD) monitoring.

A medical application is installed on an Android smart phone (IPv6-capable), which receives the vital signs from the portable monitoring devices via Bluetooth. This application is also responsible for first registration of the patient to the system hence it allows for the creation of the dedicated Electronic Health Record on the application server's side.

Figure 3-19: eHealth devices

3.3.1.2 M2M Gateway

The M2M GW handles the network discovery, auto-configuration, and routing aspects of this subtestbed. Two protocols are implemented with respect to these objectives: DHCPv6-Default Route extension, and V2V2I method. These proposals are fully described in the EXALTED project report D4.2 [16].

The auto-configuration protocol is lightweight and provides IP addresses to the eHealth devices as well as a default route to the Gateway deployed in the vehicle. The protocol used for configuring default routes with DHCPv6 is illustrated in the exchange diagram of Figure 3-20.

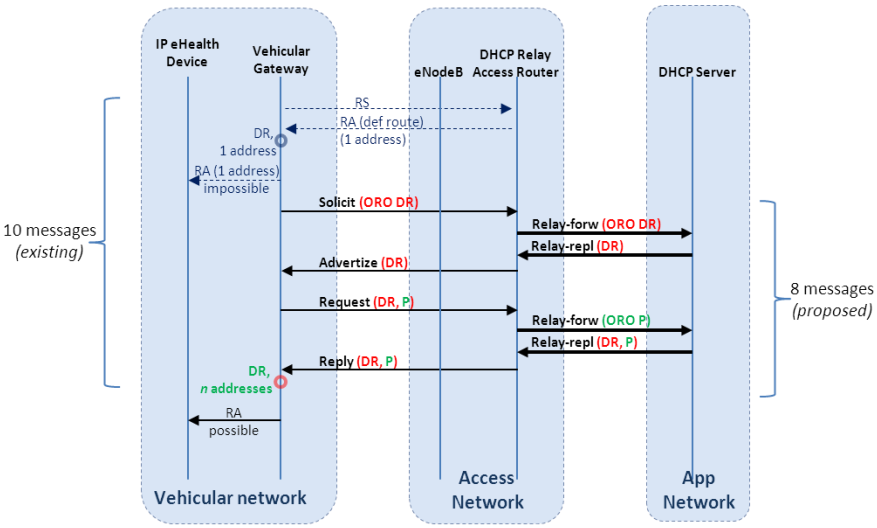


Figure 3-20: Auto-configuration protocol messages

Figure 3-20 describes the extended message exchange performed by the vehicular Gateway and the DHCPv6 entities in the infrastructure. In the original DHCPv6 protocol, to obtain a set of addresses and a default route, 10 messages are necessary (including Neighbor Discovery messages). The initial Router Solicitation (RS) / Router Advertisement (RA) offer the default route, whereas the subsequent DHCP Solicit/Advertize/Request/Reply offer the set of addresses to the Gateway (for the eHealth devices).

We propose to use only the DHCPv6 messages to provide the default route in addition to the set of addresses. Thus, the total number of messages is reduced from 10 to 8, providing optimized use of bandwidth and a reduced Round Trip Time (RTT). The measurements related to this gain depend on the cellular link quality and won't be demonstrated further in our future experiments results.

In a Solicit/Request packet a Client lists the wanted options in the Option Request Option (ORO), composed of a list of option codes. The DHCPv6 Server answers those packets with Advertise/Reply packets containing values for the options asked by the Client. The relay receives the message from the client and forwards it to the server in a Relay-forward message. The server replies to the relay with an advertise/reply message encapsulated in a Relay-reply message. The content of this message is extracted by the relay and sent to the client. In its DHCPv6 requests, the client sends a list of required options in the ORO. This option contains three mandatory fields: OPTION ORO, option-len and requested-option-code, followed by new option fields. The proposed option is named here OPTION DEFAULT ROUTER LIST. It is possible to concatenate this value with several other existing requested-option-codes. The value of this code in this option is to be assigned. Obviously, this option needs to be understood by the server as well. In the server side, the default router list option of DHCPv6 contains: OPTION DEFAULT ROUTER LIST, option-len, router-address, router-lifetime, lla_len (link-layer address length) and optionally router link layer address. As this

option contains a list, the pattern containing router address, router lifetime, lla_len and optionally router link layer address can be repeated.

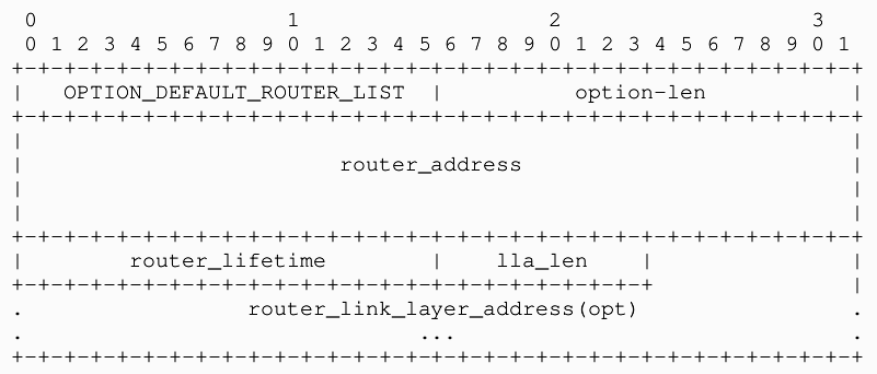


Figure 3-21: DHCPv6 default router list option format

The V2V2I communication scenario involves two kinds of vehicles: a Leaf Vehicle (LV) and an Internet Vehicle (IV). Basically, the difference between those two vehicle lies in the fact that the LV cannot access directly to the infrastructure for two reasons: (1) the M2M Gateway in the LV is not equipped with a long range egress interface (LTE/LTE-M) but only with a short range egress interface (typically WiFi) or (2) the LV is too far from the eNodeB[1] and cannot use its LTE/LTE-M interface directly. In both situations, the LV is not able to connect to the infrastructure. Therefore, it uses its short range egress interface to connect to another vehicle (the IV) that provides access to the infrastructure.

The proposed V2V2I algorithm can be explained through four steps that are presented in Table 3-2. Each step is completed using different addressing and routing protocols functionalities that are detailed in the EXALTED report D4.2 [16]. The actual testbed demonstration should be on table (not vehicles). There may exist differences between the protocol behaviour of computers on a table and the behaviour of naturally occurring events in the case of a demonstration using real vehicles. Events such as unpredictable movements, wave reflections, interference of other sources of wireless emissions, may affect the behaviour of IP protocol. However, it is widely accepted that a demonstration with a prototype in a closed setting (on a table) is a good first step towards a real demonstration.

Table 3-2: V2V2I communications

Step	Active communication	Brief description
1	V	Communications between end devices and the Mobile Router, all within one vehicle, is possible.
2	V2V	Communications between end devices in a LV and end devices in an IV is possible.
3	V2V2I	Communications between end devices in a LV and an application server in the infrastructure is possible.
4	V2V2I with mobility management	Communications between end devices in a LV and an application server in the infrastructure are maintained in a mobile environment, i.e. the Mobile Routers (MR)s change their egress address whereas end devices don't, thus maintaining ongoing communications.

[1] A vehicle is mobile across very large geographical areas; in some cases it may drive through a zone not covered by LTE/LTE-M eNodeBs. They are too far from the vehicle.

Step 1: V communications

The first step consists of enabling local communications between end devices and the Mobile Router (MR) inside a vehicle. The *routing scheme* is completed by the Neighbor Discovery Protocol (NDP). The MR informs the end devices via Router Advertisement (RA) messages that it can be used as a default route. Thus, upon reception of the RA messages from the MR, the end nodes add a default route in their routing table with the MR as next-hop.

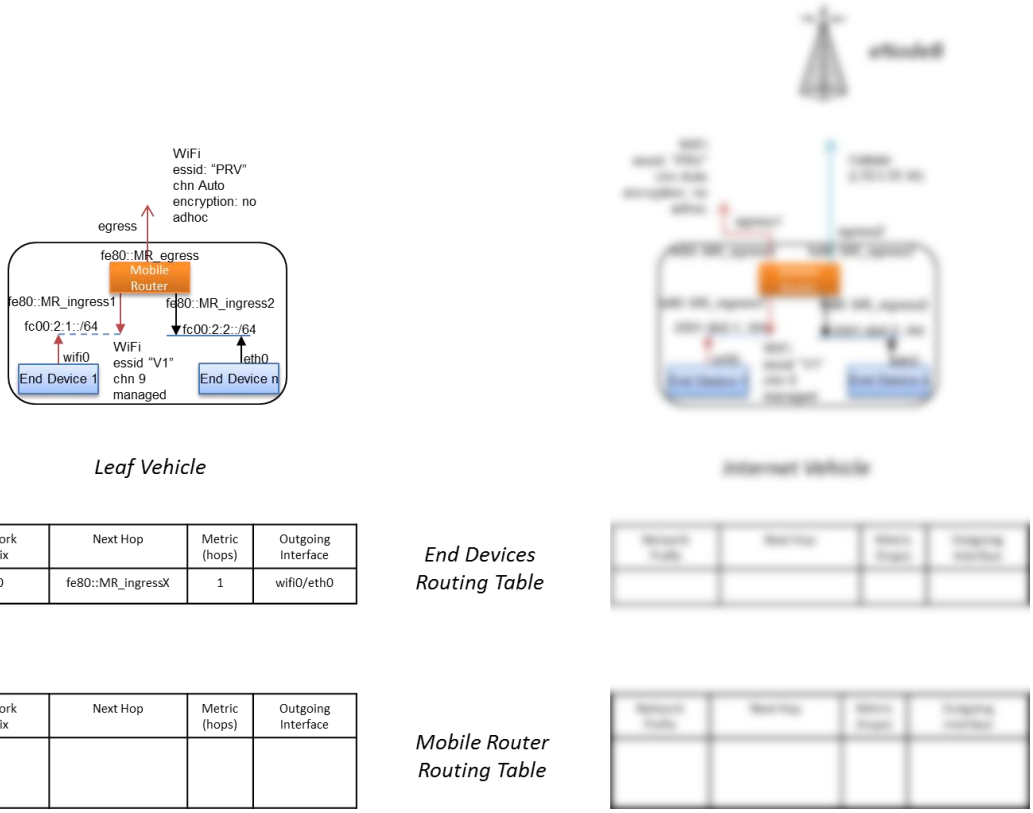
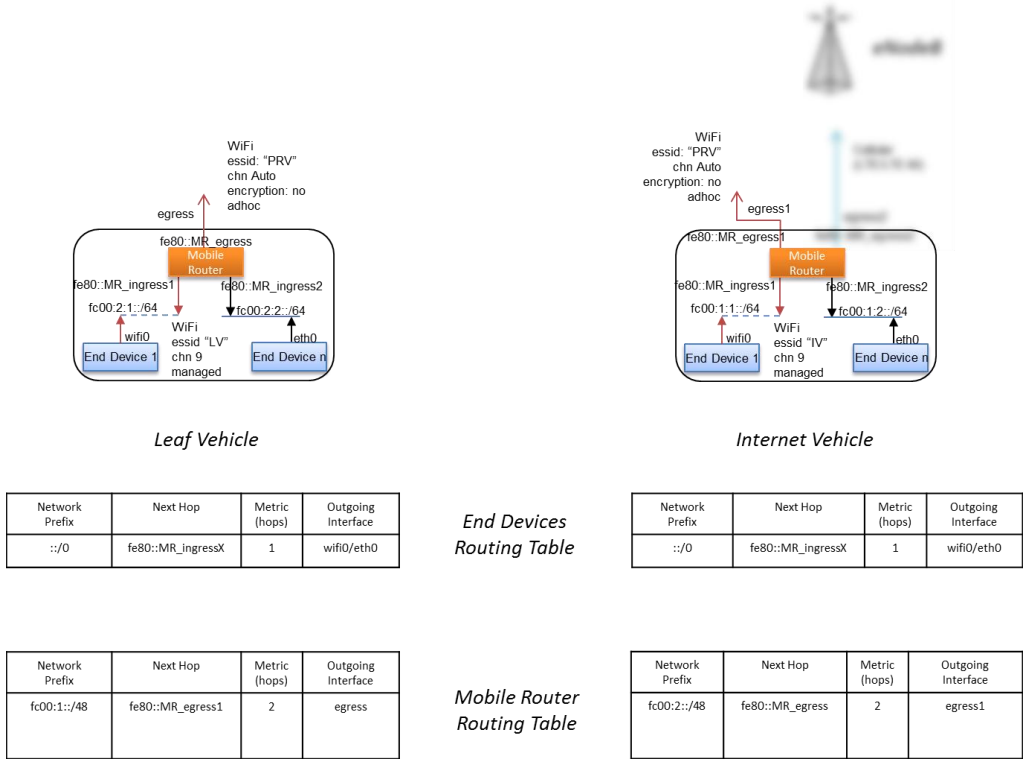


Figure 3-22: Configuration details of the V communication

Step 2: V2V communications

The V2V step starts when a LV approaches an IV. Each vehicle has already set its own private network based on Vehicular Unicast Local Addresses (VULA). Also, each end device has already set its default route. Therefore, in order to enable communications between the end devices of one vehicle and the other, the routing tables and the egress interfaces of both MRs have to be configured. When the MRs can communicate with each other locally, they will have to share the prefix of their private network in order to enable end devices communication from one vehicle to the other. More precisely, the LV MR needs to add in its routing table a route to the capillary network (based on VULA) of the IV and vice versa. This *routing scheme* is completed using a new extension of the NDP protocol: a new option in the RA messages which enables prefix exchange between nodes.



Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

End Devices Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:1::/48	fe80::MR_egress1	2	egress

Mobile Router Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:2::/48	fe80::MR_egress	2	egress1

Figure 3-23: Configuration details of the V2V communication

Step 3: V2V2I communications

The V2V2I step starts when the V2V communication is established between a LV and an IV. The IV can now share its connectivity to the infrastructure with the LV. Although the end devices of the LV are already configured with VULA-based addresses that enable them to communicate with the end devices in the IV, the site-scope limitation of VULA addresses does not allow the use of these VULA-based addresses to communicate with the infrastructure. Therefore, the end nodes have to be configured using a global prefix.

The DHCPv6 protocol relies on the fact that there must be at least one DHCP Server (or DHCPv6 Relay) on-link, it may not be a reliable protocol in some cases. For example, in the V2V2I context, relying on the DHCPv6 protocol to provide a global prefix to the LV assumes that the IV provides a DHCPv6 service (i.e. the MR in the IV implements a DHCPv6 Relay or Server). In order to avoid this case, one solution would be to extend the NDP protocol such that it also provides the Prefix Delegation option. Indeed, the NDP protocol is a native protocol of the IPv6 stack: A node that does not implement the NDP protocol cannot configure its link-local address and therefore is unable to connect to an IPv6 network. Therefore, it seems more appropriate in our context to improve the NDP protocol with a Prefix Delegation extension rather than relying on a potential DHCPv6 implementation at the IV. Nevertheless, our algorithm relies on both protocols in order to provide a global prefix to the LV: NDP Prefix Delegation is first tried and if it fails then the DHCPv6 is tried. Therefore, the *addressing scheme* in this step is provided by either NDP or DHCPv6.

The *routing scheme* part of this step consist of only one action: in order for the MR in the LV to forward IP datagrams coming from the end devices to the infrastructure, it has to add a new route entry in its routing table. This route entry is its default route and points to the MR in the IV. The MR in the LV adds this route in its routing table when it has received a global prefix from the IV (i.e. when the addressing scheme part is successfully done).

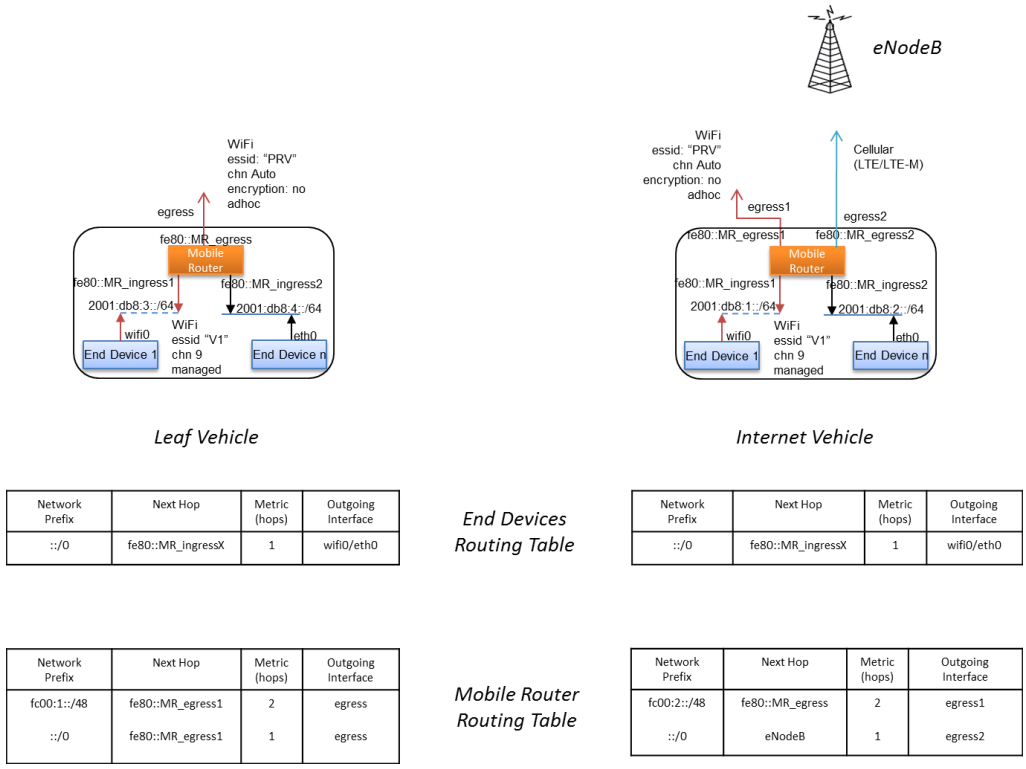


Figure 3-24: Configuration details of the V2V2I communication

Step 4: V2V2I communications in a mobile environment.

The last step of the proposed algorithm deals with mobility management. This step intends on supporting the mobility for the LV. Although it is described in sufficient details in the EXALTED project report D4.2 [16], it is unlikely to be shown in the testbed

In the following we illustrate the complete V2V2I method by means of illustrative message exchange diagrams.

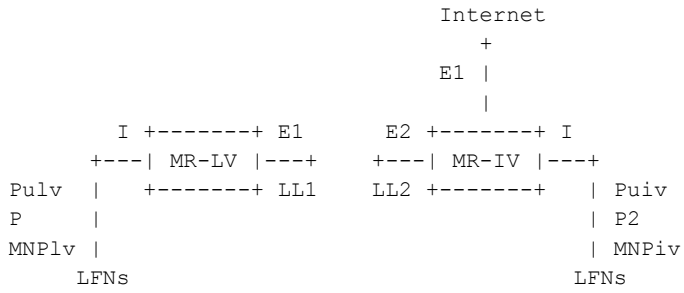


Figure 3-25: Topology for the V2V2I algorithm

In the topology shown in Figure 3-25, the following notation is used:

- Pulv: prefix based on VULA, used in the capillary network of LV.
- P: globally scoped prefix, assigned by MR-IV to MR-LV.
- MNPlv: the Mobile Network Prefix, used for mobility management.
- MR-LV: the Mobile Router of LV.
- I: the ingress interface of LV.
- E1: the egress interface of MR-LV.

- LL1: the link-local address of E1.
- E2: the egress interface of MR-IV.
- LL2: the link-local address of E2.
- MR-IV: the Mobile Router of the IV.
- E1: the egress interface of the MR-IV.
- I: the ingress interface of MR-IV.
- Puiv: the prefix based on VULA, used in the capillary network of IV.
- P2: the globally-scoped prefix used in the capillary network of MR-IV, topologically correct at the fixed infrastructure.
- MNPiv: the Mobile Network Prefix valid in the MR-IV, used for mobility management.
- LFNs: Local Fixed Nodes, the end nodes in the capillary networks.

The algorithm is described in a condensed manner by the following Figure 3-26:

```
IV->LV[Puiv]                                  [1]
LV[Puiv,LL2,E1]                               [1]
LV->IV[Pulv]                                  [1]
IV[Pulv,LL1,E2]                               [1]
V2V (Pvula)
If [P,DR] IV[ND]                              [2,5]
      V2V2I (Pglob)
      If [CoA] IV[DHCPv6]                     [7]
            V2V2I (MNP)
      End
Elseif [P] IV[ND]                             [2]
      IV[P,LL1,E2]                            [2]
      If [DR] IV[DHCP_DR]                     [4]
            V2V2I (Pglob)
      If [CoA] IV[DHCPv6]                     [7]
            V2V2I (MNP)
      End
      Else
            IV->LV[P2]                        [1]
            LV[P2,LL2,E1]                     [1]
            V2V (Pglob)
      End
```

```
Elseif [DR] IV[ND]                            [5]
      If [P] IV[DHCP_PD]                      [3]
            V2V2I (Pglob)
            If [CoA] IV[DHCP]                 [7]
                  V2V2I (MNP)
            End
      End
Else
      If [P,DR] IV[DHCP]                      [3,4]
            V2V2I (Pglob)
            If [CoA] IV[DHCPv6]               [7]
                  V2V2I (MNP)
            End
      Elseif [P] IV[DHCP_PD]                  [3]
            IV->LV[P2]                        [1]
            LV[P2,LL2,E1]                     [1]
            V2V (Pglob)
      End
End
```

Figure 3-26: Condensed description of the V2V2I algorithm

This description of the algorithm is composed of two columns. The execution starts with the left column, read top-down, and then the right column in a similar manner.

The algorithm represents a top-level view of the V2V2I solution: It describes in a mixed manner the behaviours of both LV and IV. For implementation, a bottom-level view is necessary, where the behaviour is described only from LV perspective, or only from IV.

The algorithm is organized as an alternation of preferences. Mainly, the LV acts depending on the information it receives from IV. For example, at line [2,5] on the left column, if the IV delivers prefix and default route by using NDP then LV takes a number of actions. The same kind of behaviour is repeated for all cases of using DHCPv6 or ND.

The left column concentrates on the case where IV uses ND to deliver various data to the IV (prefix, default route) and LV insists by using DHCPv6 when one such parameter is not delivered, whereas the right column is vice-versa.

First, IV informs LV about its Puiv; then LV inserts a routing table entry corresponding to Puiv, LL2 and E1. Similarly, LV informs IV about Pulv and IV inserts a routing table entry. At this point, V2V is possible based on Pvula.

Then, on the line [2,5], if IV delivers a prefix P and a Default Route (DR), using ND, then V2V2I communications are possible, using Pglob (or otherwise simply named P). Further, if a Care-of-Address (CoA) is delivered by IV to LV by using DHCPv6, then V2V2I with mobility management is possible for LV, using its MNP. Otherwise (line [2]), if only the prefix P is assigned by IV to LV with ND (and not the default route), then the IV inserts a routing table entry for P towards LL1 on E2. At this point V2V communication is possible.

In addition, if a default route is offered by IV by using DHCPv6, then V2V2I communications are possible, using Pglob. Moreover, of a CoA is offered by IV using DHCPv6 to LV, then V2V2I with mobility management is possible, based on the MNP.

Finally, the right column represents the behaviour whereby the IV prefers the use of DHCPv6 and, optionally, ND.

The following message exchange diagram in Figure 3-27 illustrates the behaviour of MRs when ND is preferred initially for delivery of prefix and default route.

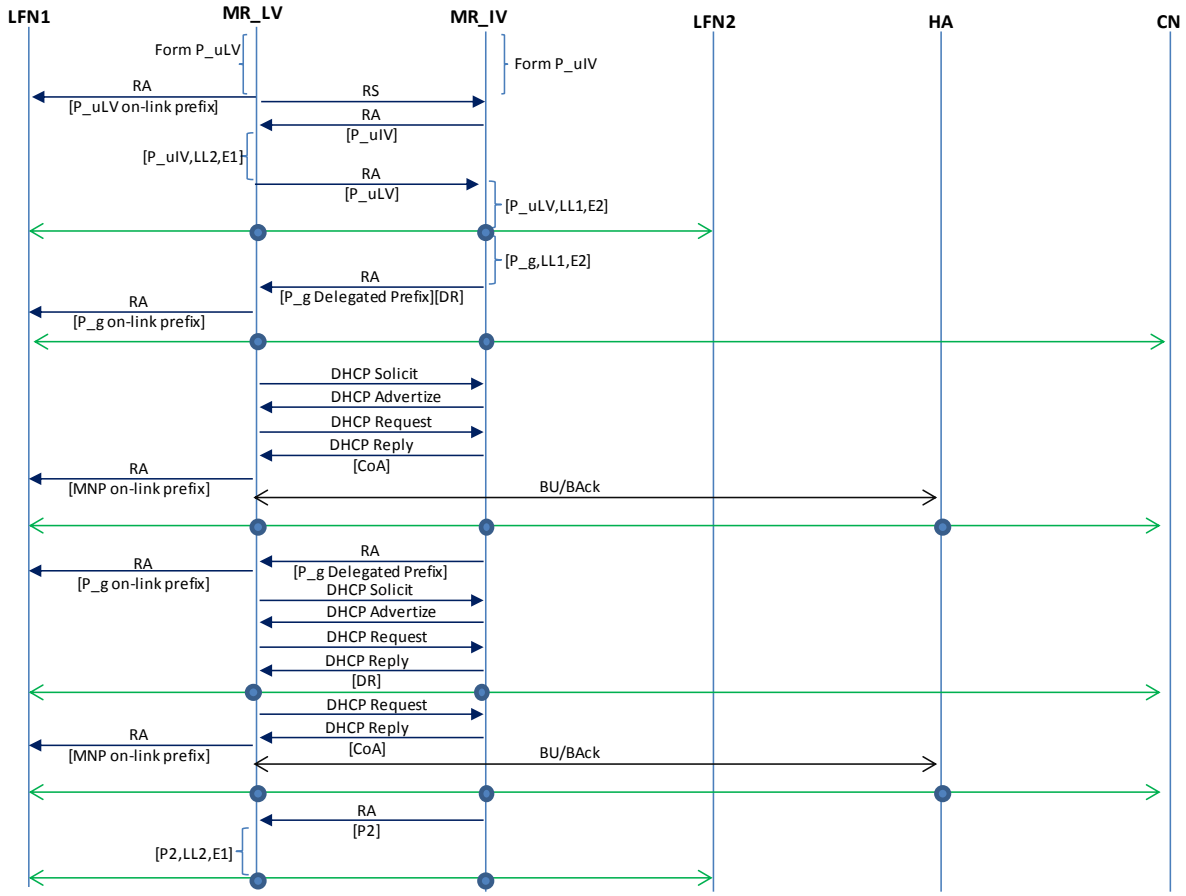


Figure 3-27: Message exchange diagram for ND preference (left column)

The columns are representing the entities in the topology and are named as follows:

- LFN1: a Local Fixed Node, an end node, within the capillary network of LV.
- MR_LV: the Mobile Router of the LV.
- MR_IV: the Mobile Router of the IV.
- LFN2: a Local Fixed Node within the capillary network of IV.
- HA: the Home Agent of MR_LV.
- CN: a Correspondent Node of the LFN1.

Initially, MR_LV and MR_IV form prefixes based on their respective VULAs: P_uLV and P_uIV Once the P_uLV is formed, it is advertised by the MR_LV to the LFN1, by using a Router Advertisement (RA). This is advertised as on-link prefix within the capillary network of the LV.

Subsequently, or simultaneously, MR_IV and MR_LV exchange their respective prefixes using RAs on their egress interfaces. Upon reception of such an RA, a MR inserts a routing table entry corresponding to the received prefix. Upon completion of this message exchange, communication between LFN1 and LFN2 is possible (V2V communication).

Next, it may be possible that MR_IV has a globally-routable prefix (P_g) that it can allocate to MR_LV. Before performing this allocation, it must add an entry in its routing table stating that P_g is reachable at the egress interface of MR_LV. After this addition, it will send a RA containing that P_g and the indication of being a default router. Upon reception of P_g, the MR_LV will advertise P_g to the nodes in its capillary network. After this exchange, the LFNs in LV's network are able to communicate with arbitrary CNs in the infrastructure.

Further, if MR_LV is able to obtain a Care-of Address from MR_IV, by using DHCPv6, then MR_LV will be able to advertise the MNP (Mobile Network Prefix) to the nodes in its capillary network and, subsequently, send a NEMOv6 Binding Update to its Home Agent. At this point, the nodes in LV may use their globally-scoped permanently reachable addresses and session continuity – mobility management with V2V2I. The communication from LFN1 to CN will tunnel through MR_LV's Home Agent.

On another hand, if in the earlier step the RA only sent the P_g to MR_LV, and not the default route, the MR_LV will request the assignment of default route by using DHCPv6. After the 4 messages of the DHCPv6 exchange containing the default route, the LFN1 will be able to communicate with CN. Additionally, if the DHCP exchange provides a CoA to MR_LV, then MR_LV is again able to advertise MNP to its nodes and perform BU/Back (Binding Acknowledgement) with its Home Agent.

The last two messages show what happens in case neither the Delegated Prefix P_g nor the default route are provided neither by DHCP nor by ND. In this case it is possible for MR_IV to advertise its globally routable prefix (P2) using Router Advertisement, such that the two vehicles may still communicate in a V2V manner (as a safety case when MR_IV may not be able to generate its P_uIV).

3.3.1.3 Application Server

The recorded data from the devices are transferred automatically (in the absence of the Mobile Router) through the smart phone via GPRS, Ethernet or WiFi to a designated web centre (over IPv6). The application provides a simple Electronic Health Record (EHR) for disease management and treatment and initiates patients' active involvement in healthcare. Analytically, it features browsing on the exams history, viewing of the recorded data, downloading of a diagnosis or advice from a doctor, additional comments and more. The final destination of these data is the EHR of the patient who uses the devices and it is resident in a dedicated server from where it is accessible for reviewing under secure credentials by the treating physicians. It has to be noted that the application server only accepts data from the vehicle while other forms of communications, such as emails, online chats etc, with the ambulance crew are not supported as it is out of scope for the current set up.

3.3.2 Subtestbed 2.2: Heterogeneity and interoperability

In the following we describe the main contributions, namely device discovery and registration for M2M devices into the M2M gateway, a zero-configuration approach; Semantic data dissemination for query and information dissemination to reduce the traffic in/out the capillary network and Semantic data aggregation, for data aggregation in the capillary network and the gateway to minimise the transmitted data and submit only relevant information.

3.3.2.1 Device Discovery and Registration

To establish a reliable connection between M2M device and M2M gateway, a similar approach as the association and negotiation protocol from the IEEE802.11 Standard has been adopted. The mechanism is implemented in the data collection function of the M2M gateway and adopted for each sensor platform to enable integration of heterogeneous devices. Before a M2M device is able to exchange information it has to be registered in at least one M2M gateway. The M2M gateway will send a beacon signal every few second which the M2M devices can use to register themselves to the M2M gateway. First, after receiving a beacon signal the M2M device has to verify if it is allowed to connect to a specific M2M gateway. This is done by sending an authentication request to the M2M gateway. The request includes the ID of the M2M device and the encryption setting if available. The device ID is needed to identify if the M2M device is allowed to connect. Encryption setting can be also used to establish a secured connection between M2M device and M2M gateway. Messages send by a M2M device or by the M2M gateway must be acknowledged, if a request or response does not get acknowledged in a certain amount of time (depending on the distance between device and M2M gateway) the request or response will be send again. The response time can be also used to approximate the distance between M2M device and M2M gateway. The authentication response tells the M2M device if it is allowed to proceed in the negotiation process or not.

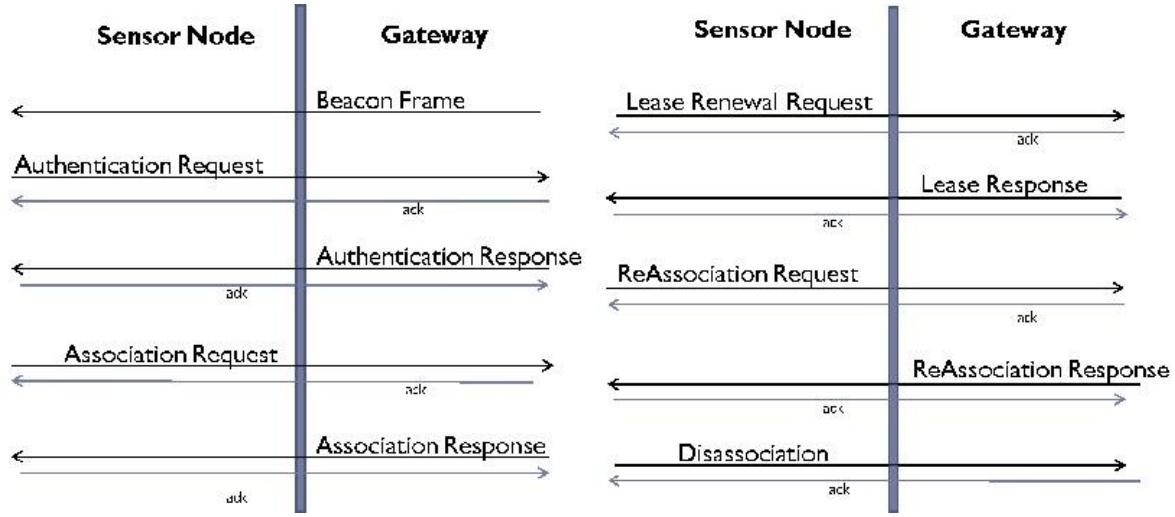


Figure 3-28: Association procedure

If it is allowed, the M2M device has to request association with an association request. The request includes: battery status, signal strength, hop distance (if possible) and the capabilities of the device and their current state. This information will be stored in the semantic database and used for the further algorithms of the M2M gateway. The M2M gateway will send a response, which includes the lease time of the association. During the lease time period, the device must renew its lease to indicate the M2M gateway that it is still available. The lease renewal process requires that the renewal request contains the same information as an association request. The response contains the new lease period. The lease duration time depends on the battery life and the distance to the M2M device.

3.3.2.2 Semantic Data Dissemination

In Gossip Protocols [25], N M2M devices talk in a peer-to-peer way to other M2M devices. The decision is made random or deterministic on which M2M device will communicate next. The underlying networking aspects are not part of the gossiping algorithm and it is assumed that M2M device to M2M device communication via multi-hop is supported by lower layers. The messaging mechanism is divided into virtual rounds where in each round a M2M device can send a message to another M2M device. Therefore in each round the number of M2M

devices which also start gossiping increases exponentially. This effect leads to fast and reliable data dissemination in the network. In a network of N M2M devices the worst-case scenario will lead to O(log(N))+O(ln(N))+O(1) rounds where in each round the distribution of M2M devices that do not send any messages follows the exponential distribution. To reduce the number of communicated messages in the gossip protocol while retaining the fast dissemination we propose a deterministic M2M device selection, where a network is virtually split into semantic similar groups based on the context information of the network. This leads to a new N_ which is defined as N=C where C is the amount of the introduced groups. This limits the messaging only to a certain group. The groups are defined by generating sub graphs also called overlay networks. The overlay networks are created according to the structure of high-level concepts (i.e. context definitions) in a model and form a logical network. The context information is stored in a semantic model based on the W3C Semantic Sensor Network Ontology where concepts are linked in a hierarchical structure. Virtual groups are formed based on their subClass relationship in the context model. A possible application scenario is: M2M devices distributed in different rooms in a building. The different rooms will therefore be the criteria to group the network. This approach is not limited to spatial information and can also be applied to other relationships defined in the context model.

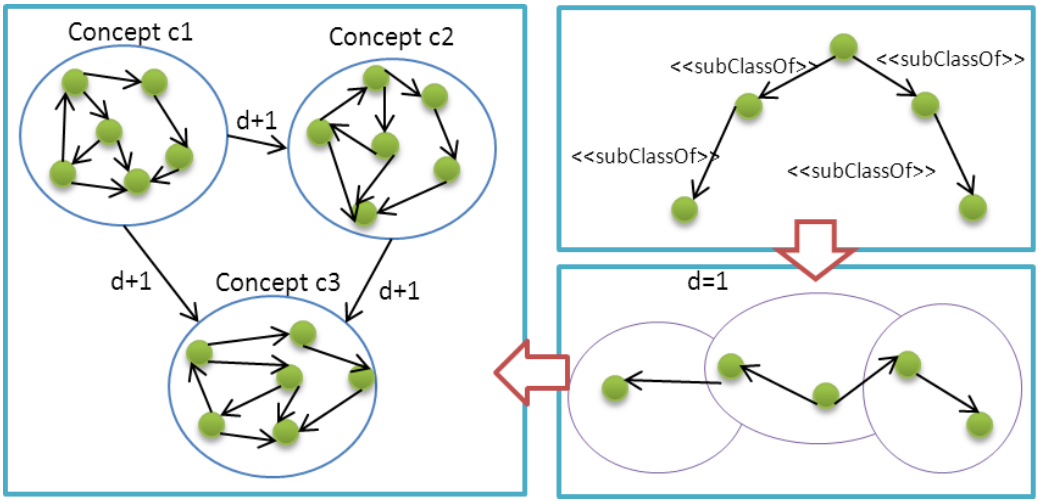


Figure 3-29: Context based grouping

As shown in Figure 3-29, groups are created based on some relationships designed in the context model. This leads to two different types of connections, intra-group connections where M2M devices of one group communicate with each other and inter-group connections where M2M devices of different groups communicate. The degree of linking and therefore the reliability of the overlay network and the gossip algorithm depend on the fan-out of the inter- and intra-connected M2M devices. The fan-out in turn depends on the underlying topology.

3.3.2.3 Semantic Data Aggregation

The data aggregation is based on the symbolic aggregate approximation [21] used to transform continuous time series data into discrete values. This leads to a significant compression of the data.

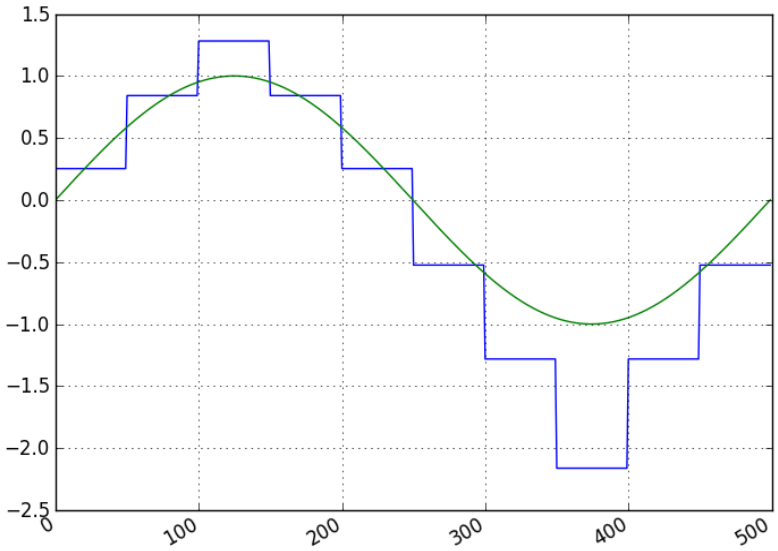


Figure 3-30: Symbolic Aggregate Approximation (SAX) transformation

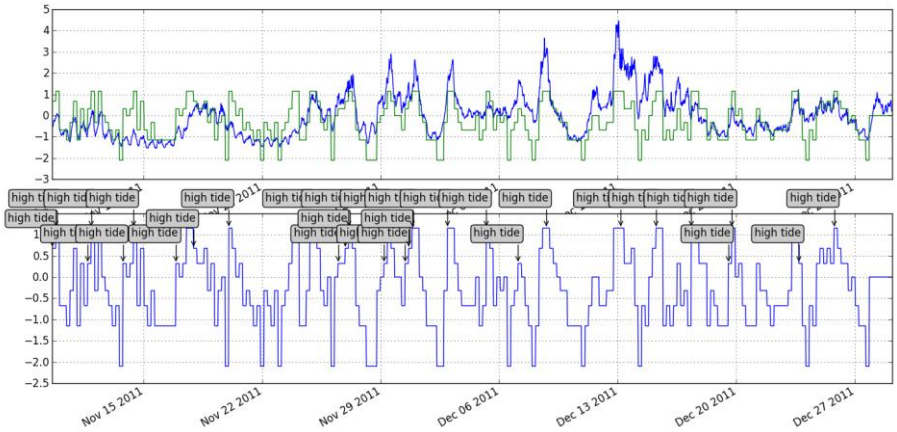


Figure 3-31: Linkage to semantic model

As shown in Figure 3-31, representative data sample over a time period is abstracted, the patterns created by the transformation are linked to the semantic context model, this leads to a reduction in the transferred data from sensor to gateway as only abstracted information is transmitted. In this scenario, the data output from the data aggregation is the discrete time samples that is abstracted from the real sensor data (continuous data), such as discrete sample temperature data or simply high tide events during the period. The data processing and aggregation is used to discretize the raw sensor data into a lower-dimensional representation. This component utilizes the Symbolic Aggregate Approximation (SAX) algorithm to convert continuous data (e.g. {1,2,3,4,5,4,3,2,1}) into a compressed discretized representation (e.g. fa,b,b,ag). We then use an abductive model that stores the mapping between discretized representation and abstractions (e.g. {a,b,c,d}) representing a change in the light sensor data in a room. The abductive reasoning and temporal component infers the current observations and determines which abstractions are the most plausible ones.

We use the Parsimonious Covering Theory (PCT) [26], an abductive logic framework to transform the sensor data to abstractions. The parsimonious covering theory is predominantly used in the medical domain to model and infer the disorder of a patient based on observations made by a doctor. It uses an abductive approach which is based on partial observations. Abductive reasoning infers the most likely explanation given a set of incomplete or partial observations. In contrary to deductive reasoning where a conclusion always can be inferred, abductive reasoning gives only an educated guess about the most likely explanation. The advantage of abductive reasoning is that for partial-observable concepts and incomplete observations a conclusion can be given, where deduction would need the complete observations to draw conclusions.

The created patterns using SAX are then linked to semantic descriptions that define thematic, spatial and temporal features, providing highly granular abstract representation of the raw sensor data. This helps to reduce the size of the data that needs to be communicated from the sensor nodes to the gateways or high-level processing components. The energy cost of transmission via wireless radio on sensor nodes is higher than doing an internal process. SAX can be optimally implemented within the sensor nodes. So this can also enhance the energy efficiency of wireless sensor nodes that are used for long term observations. We also discuss and evaluate a method that uses abstract patterns created by SAX method and occurrences of different observations in a knowledge-based model to create perceptions from sensor data.

We have evaluated our approach by collecting data from heterogeneous sensors and running the SAX method on a wireless sensor network. The evaluation results show that our proposed methods can effectively create low-level (i.e. SAX patterns) and high level (i.e. perceptions) abstractions from the sensor data. Sending SAX patterns that are represented as string “words” also reduces the size of information that is sent from the nodes. The patterns are created based on analyzing observations and measurement as time series data, so it does not require any prior threshold definition or heuristics to set the parameters for pattern creation.

3.3.3 Subtestbed 2.3: Connectivity for low-power devices

Apart from high level functionalities depicted in section 3.2.3 the main functionalities embedded on each element of the architecture can be summarized as follows:

- **Non-LTE-M devices**
    - They must send bootstrap messages containing petition towards its default M2M Server. The IP address of that M2M Server must be stored on the devices.
    - They must handle commands received from M2M Server so as to act over the LED connected to one of the I/O pins of the board.
- **M2M Gateway**
    - They are in charge of creating and maintaining socket connectivity between the end node and the M2M Server:
        - Once powered on, a new socket is initialized to establish the connection with the server and receive command destinated only to the gateway.
        - Every time a node sends a data request to the M2M Server, the gateway must check if it is already registered. In case it is not, a new socket must be created to tunnel information from that specific node. Otherwise, the data is redirected through the previously established connection from that node.
        - It must maintain a table mapping nodes with socket sessions active. In order to update the table, the gateway must poll the network so as to identify which are the active nodes, so as to be able to handle node disconnection due to

bad coverage or low battery power. In case a node previously registered is not reachable, the socket connection must be terminated.

  - It must maintain a black list identifying malicious nodes. The M2M Server is supposed to have the needed intelligence to detect nodes that are not working properly and may compromise the security of the network. In that case, the gateway must block them.

- **M2M server**
  - It must show in a dynamic way the nodes and gateways active by that moment.
  - It must store historical data about nodes registered and medicine tags approached to nodes, filtering by end node sending the command and medicine approached.
  - It must enable sending asynchronous commands to nodes regarding:
    - Remote acting over LEDs
    - Blocking malicious nodes

Having in mind these functionalities and the message formats described in sections 3.1.3 and 3.2.3 the possible messages transmitted over the testbed can be seen in Figure 3-32. They can be categorized into:

- Registration messages, derived from bootstrapping both end nodes and gateways.
- Messages derived from approaching NFC tags to Non-LTE-M devices, resulting into a request to the M2M Server for including that medicine in the stock table and receiving a command back indicating the success or failure of the operation.
- Messages derived from asynchronous commands sent by the M2M Server to force a specific behaviour on nodes.
- Request from M2M Server to include malicious nodes into the Gateway's black list.

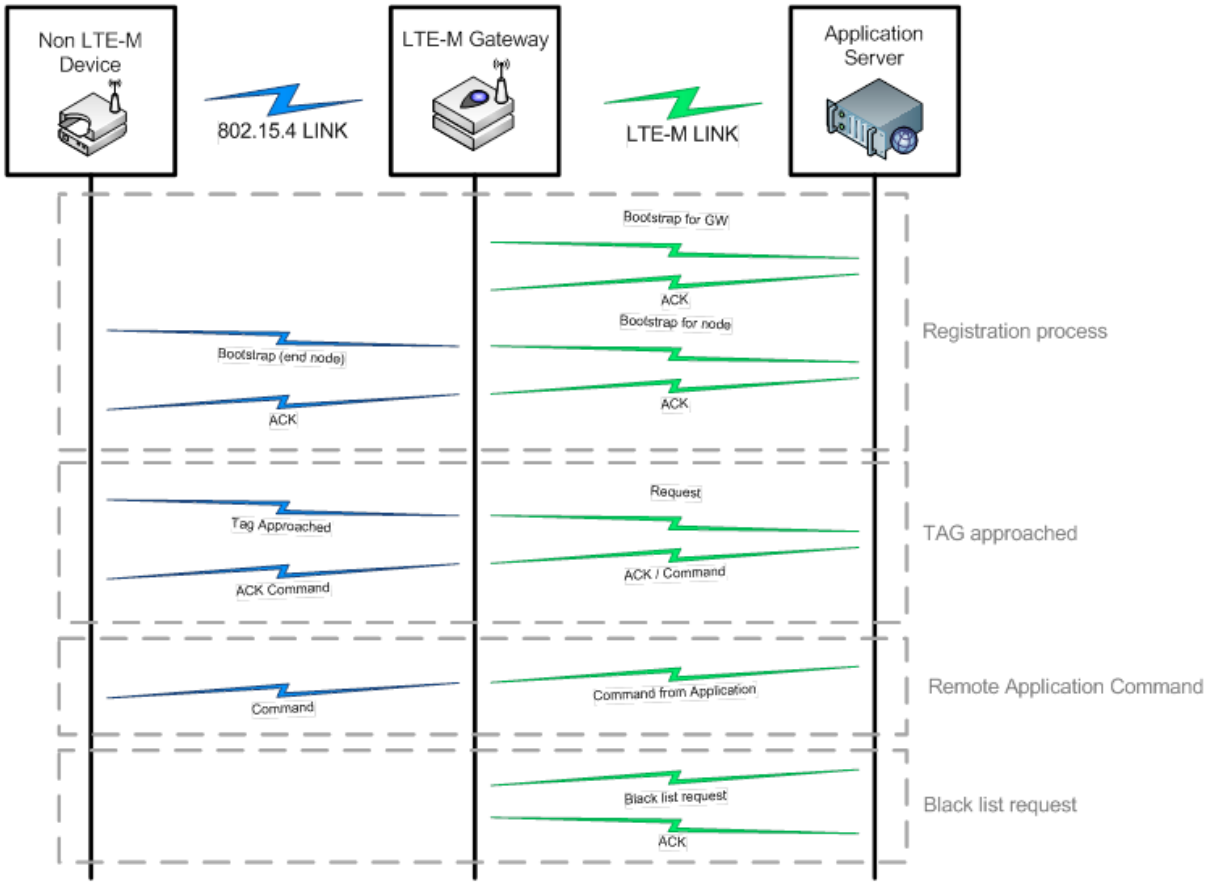


Figure 3-32: All different messages transmitted in the testbed

The flow chart and state diagram of each element in the testbed is described in detail in the following subsections.

3.3.3.1 Non LTE-M devices

The idea about Non-LTE-M devices is keeping them as simple as possible. In order to do that, it is needed to identify the critical tasks that they need to implement and avoid making them perform any other optional tasks. More elaborated functions in order to manage them will be embedded on gateways.

This way, each end node performs the following tasks:

- Once switched on, it must authenticate itself to the M2M Server. A specific message is sent to the Gateway in order to perform this task. The node must then wait for the confirmation, and, in case it is not received before a predefined timeout, it must retransmit the request until it gets confirmed.
- Every time a NFC tag is approached, it must read the proper data field (defined by the application) and transmit the date towards the gateway.
- In case a command is sent from the M2M Server (due to a request from the node or via an asynchronous request from application users), it must handle acting over its LED connected to the I/O ports.

The way the information, data and commands flow can be seen in Figure 3-33.

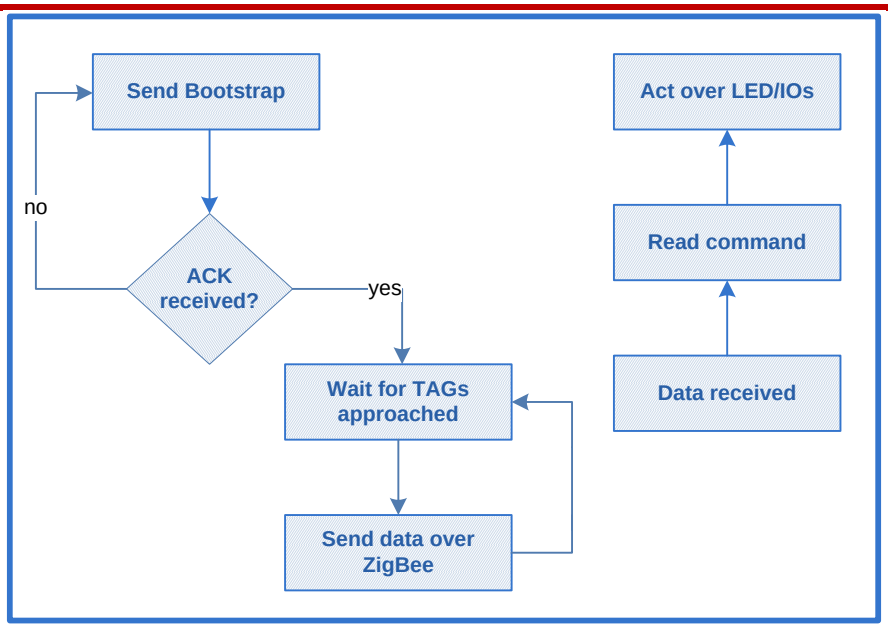


Figure 3-33: Flow diagram for Non-LTE-M devices

Devices are encapsulated and battery powered. The LED is visible outside the encapsulation and there is a switch able to power on/off the module (Figure 3-34).

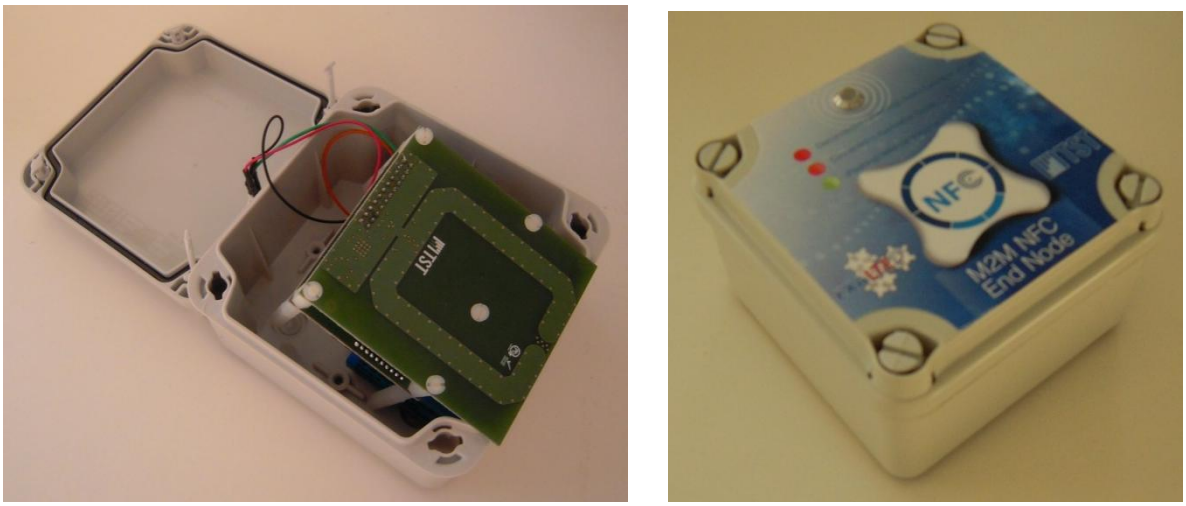


Figure 3-34: Non-LTE-M device on its encapsulation

3.3.3.2 M2M Gateway

The Gateway is the device in the capillary network equipped with extra resources in charge of handling more heavy and specific managing tasks regarding all the nodes behind it.

Its main function is redirecting the data received from Non-LTE-M nodes via the capillary interface towards the M2M Server in the core IP network through its cellular interface.

But it has as well many other functions associated. They can be classified into periodical tasks (performed at certain predefined moments in a cyclic way during the time the device is up), transmission tasks (activated each time data is received from an interface) and managing tasks (related to special features).

Each of these tasks is described on the following list:

- Periodical tasks:

  - The gateway must maintain an updated list of active nodes in the network. Each time a node tries to authenticate to the M2M Server, a new entry is created in the gateway's database, but this entry must be checked periodically to avoid having mapped on the gateways nodes that are not anymore present in the network.

- Transmission tasks:
  - As well as Non-LTE-M nodes, the gateway must authenticate with the M2M Server when powered on, and wait for the confirmation or retransmit the request.
  - Each time a message is received in the capillary interface, the address translation mechanism starts working. If the message is for authentication, the gateway must create a new socket connection towards the IP address specified on the message from the node, and map it in the internal database, associating the node address with the socket session created.
  - If it is a regular message, the gateway must check which session is associated to the node address and redirect the info using that socket. It may happen that the connection is broken by the time it tries to transmit, or that the node is identified as malicious by the server and blocked, so the proper security and error handling mechanisms are implemented in the gateway so as to be able to manage this situations, creating new connections for the node and blocking its data respectively.
  - The same way, if a command is received in the cellular interface, the address translation is done the other way around, selecting the capillary address that maps the socket session from which the packet is received and sending the command in the proper format.
- Managing tasks:
  - Gateways must maintain an updated black list regarding malicious nodes. The detection is done by the M2M Server, and it may request inclusions on the black list, so as to avoid these nodes to transmit outside the capillary network. In that case, when the command is received from the server, the gateway must include in the black list database the requested node and ignore from then the packets received.

All these functions are explained using the flow diagram in Figure 3-35:

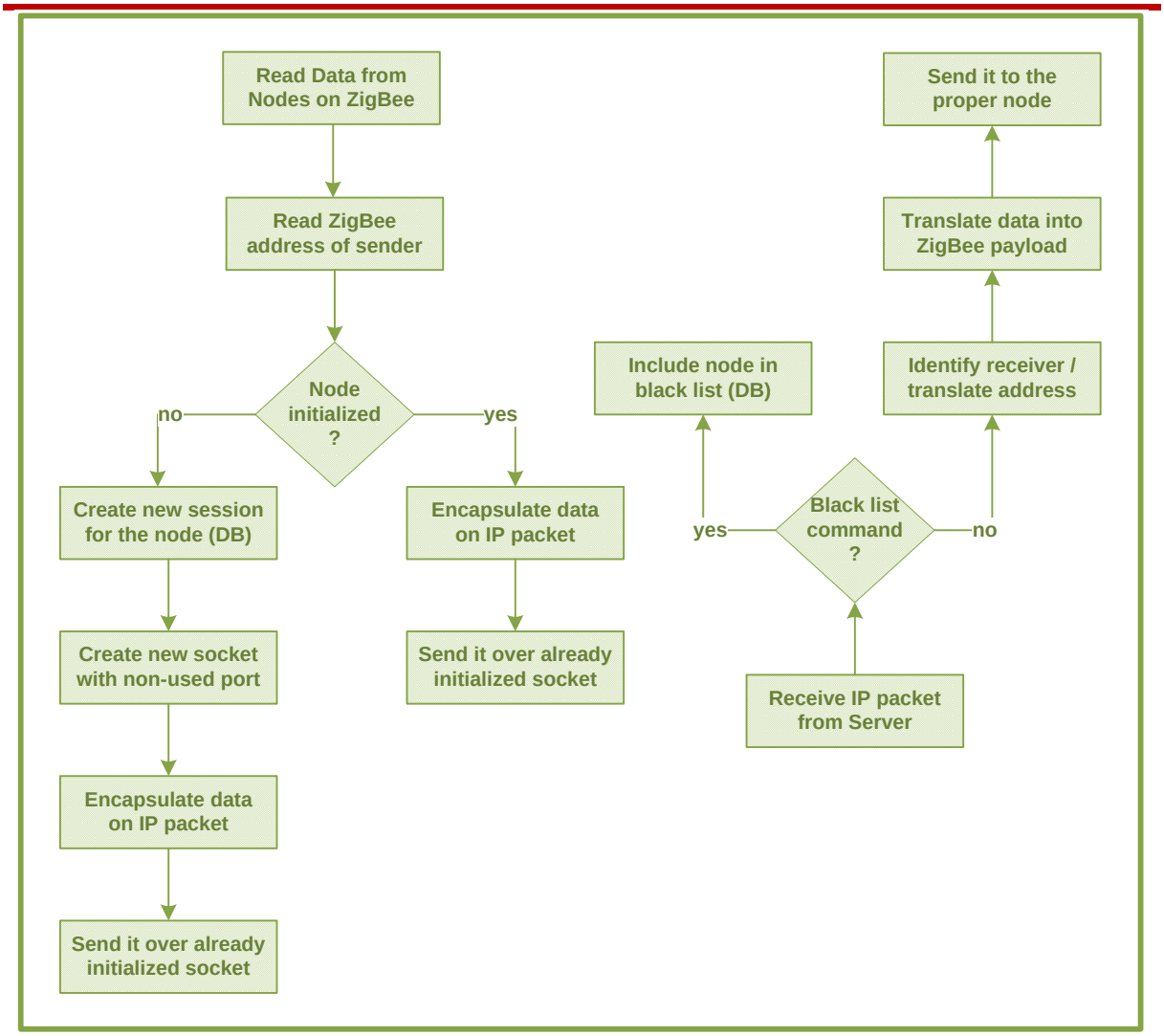


Figure 3-35: Flow diagram for M2M Gateways

Finally, as in Non-LTE-M nodes, an encapsulation is provided for gateways, with the same idea of having LED and power switch accessible from the outside. In addition, this time a power socket is also provided to power the node and the cellular antenna is also placed outside the encapsulation so as to assure coverage in the proof of concept area (Figure 3-36).

Figure 3-36: Gateway on its encapsulation

3.3.3.3 M2M Server

The idea of the M2M Server relies on a remote piece of hardware placed somewhere in the IP core network able to handle requests from devices, managing them and sending some commands to act over them.

In this particular subtestbed the application relates to hospital logistics, so the server is in charge of:

- Maintaining an updated list of active nodes and gateways authenticated. It is done by adding nodes at authentication process and updating the active nodes by testing socket sessions periodically.
- Storing the data related to the medicines approached to Non-LTE-M nodes in a database. The way this info is further used is out of the scope of this demo, but it can be used to place just-in-time orders so as to maintain the needed stock in hospital, as an example. In addition, the information must be suitable to accept queries and be filtered by medicine or sender node.
- Sending commands so as to act over the nodes, modifying its state and acting over the LED on the I/O interfaces.
- Managing the network and performing black list inclusions so as to avoid malicious nodes to send data through the network.

These are the tasks shown in Figure 3-37.

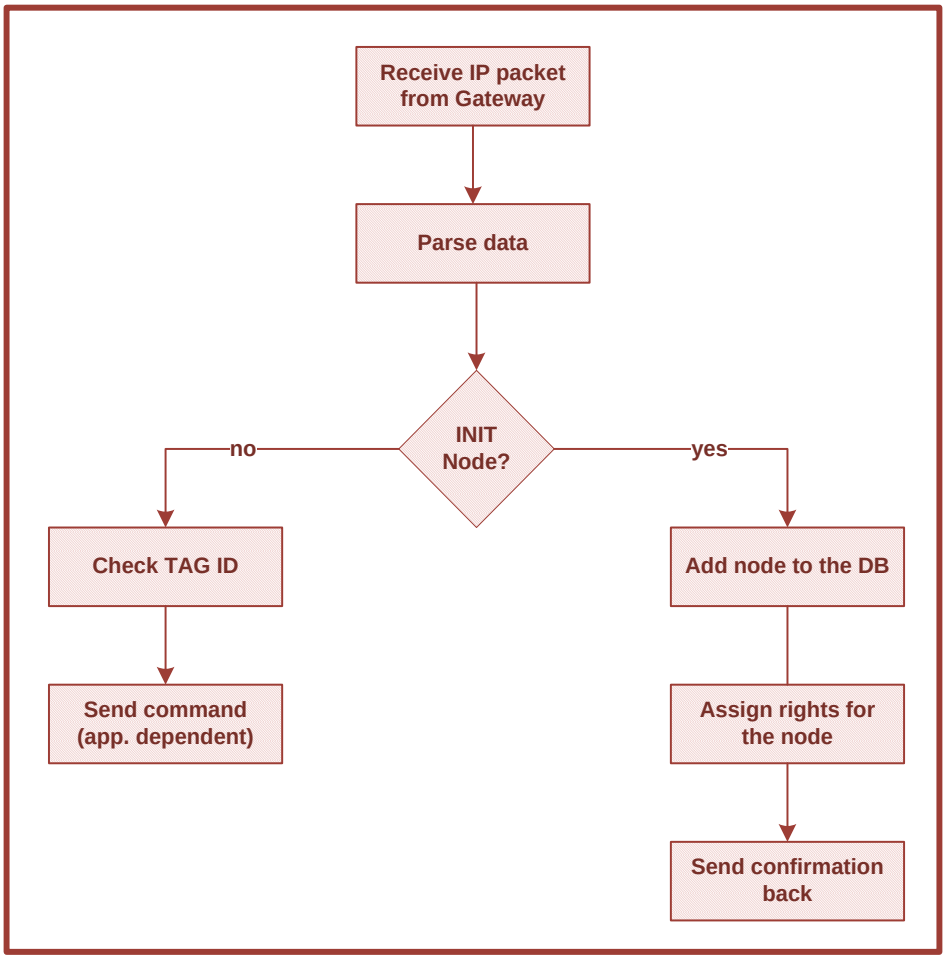


Figure 3-37: Flow diagram for the M2M Server

Finally in Figure 3-38, all these features can be seen:

- On the left hand side, the active gateways (in red) and nodes (in blue) are shown. Clicking each gateway allows showing the devices behind it, so the blue nodes present on the list are only the ones associated to the selected gateway.
- Each blue Non-LTE-M node has three coloured buttons (red, green and orange), used to send asynchronous commands to act over the LED colour on nodes.
- The central part of the interface has three tabs:
    - Medicine tab shows the database of medicine tags approached to the nodes. It shows the last occurrences and the info can be filtered by type of medicine or sender node.
    - Log tab shows all events happening in the network, concerning not only data transmission associated with tags, but also device authentication and special requests.
    - Finally, lists tab enables managing the black list, isolating malicious nodes from the network.

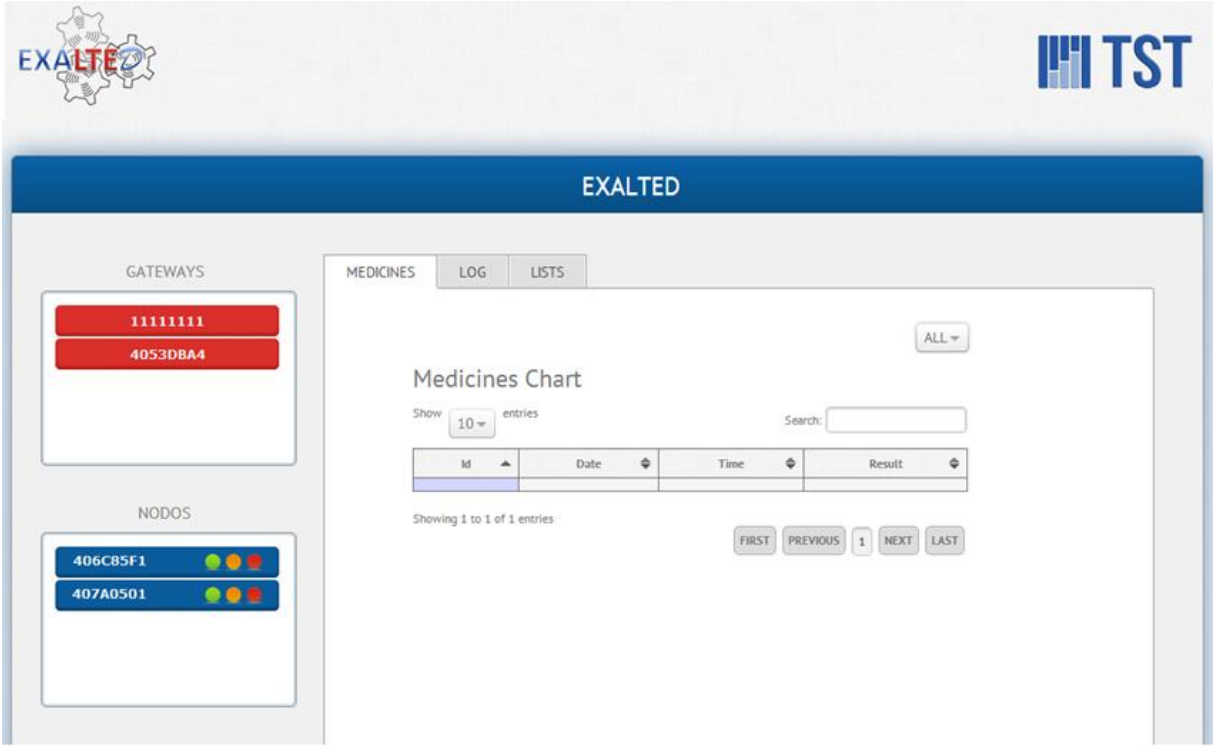


Figure 3-38: Overview of the M2M Server interface

3.3.4 Subtestbed 2.4: End-to-end security

3.3.4.1 Component features

The main feature provided with this subtestbed is the message protection against unsolicited modifications. The Crypto-Mac is prepared by the Secure Element on non LTE-M device while it is prepared by the SIM on LTE-M device. One end (typically the device) prepares the crypto-MAC that is sent with the message and the other end hashes the message on reception and encrypts the hash value with the secret key. If the result is different from the Crypto-Mac then the message is discarded.

- **Non-LTE-M device** interacts with the Secure Element and with the gateway either directly or indirectly. The Secure Element can be seen as a co-processor to secure the messaging with the M2M server. Features are listed below:
    - Maintains an event handler that periodically triggers the device to send the current meter index value to the gateway
    - Prepares the application payload containing the current meter index value
    - Processes messages sent by the gateway and reacts accordingly
    - Calls the EXALTED Secure Element to protect the payload
    - Sends/received messages to/from the gateway
- **EXALTED Secure Element (SE)**:
    - Maintain the cryptographic key able to sign application payload
    - Provides the integrity of an application payload
- **LTE-M device:** It combines features from both the SE and the non LTE-M device except that it does not communicate with the gateway but directly interacts with the M2M server through OTA. Another distinction is that cryptographic operations are performed by the SIM, there is no other secure element on this device than the SIM. Specific feature of this device is to process commands sent by the M2M server through SMS and to react accordingly.
- **LTE-M Gateway**:
    - Receives SMS from the server that have to be routed to an non LTE-M device
    - Received messages from non LTE-M devices to be routed to the server
    - Maintains a table containing the non LTE-M devices identifier, their MAC address, their ZigBee addresses
- **M2M server**:
    - Securely collects KW/h values generated by devices. These data values shall be certified by the server
    - The server can generate and send secure commands to end-devices. .These end-devices can be LTE-M or non LTE-M devices.
    - Process protected messages

3.3.4.2 LTE-M and Non-LTE-M devices

It is advantageous to have various types of device to showcase the end-to-end security, but the main motivation to use two types of devices is related to a more pragmatic project organization issue related to time. Because of the late availability of the EXALTED Secure Element a device currently design out of the project has been recycled in to act as a LTE-M device. There is no need to add an extra secure element to an LTE-M device because it already has a SIM to secure the Radio Access Network communication. Targeting this device at first enables to set the server infrastructure and test the security between the LTE-M device and the M2M server.

In a second step non LTE-M devices are integrated and the EXALTED Secure Element is used as the security mean.

3.3.4.3 Alternative design

It is possible to protect application data from malicious updates by using the security mechanisms already available in the capillary network or between the gateway and the

server. ZigBee proposes a security architecture that enables to protect data from malicious devices. SMS sent by the server to the gateway or the LTE-M devices could be protected with GSM 03.48 envelop [27] to prevent an attacker to send commands and there are many ways to secure IP communication between the gateway and the server. Compared to the end-to-end security proposed with this subtestbed these alternative designs have two major flaws:

- The service provider has to settle agreement with the MNO operator to use cryptographic keys
- There is room for an attack on the gateway: all data have to be switched from one protection system to another one that is very unsafe.

3.4 Tracing of technical requirements

The different subtestbeds composing these E2E communication demonstrators deal and address several EXALTED requirements. The project in WP2 has defined a set of functional, network, devices, service and non-functional requirements. As reference, the main activities developed in Testbed 2 are:

- Cooperation of IP and non-IP worlds based on M2M communication paradigm
- Integration of multiple heterogeneous devices into a single testbed
- Secure communications in capillary networks
- End-to-end IPv6 connectivity
- Vehicle-to-Vehicle-to Infrastructure communication
- Cooperation of IP and non-IP devices
- Support efficient provisioning of a set of M2M equipment
- Data aggregation and data dissemination at gateway level

Based on the aforementioned actions, Table 3-3 shows, which technical requirements are addressed, and in which subtestbed they are covered. It also indicates the status of implementation after the second year of the project. A complete list of all technical requirements can be found in Annex A-1.

A detailed numerical evaluation of the experiment will be disclosed in the final public deliverable.

Table 3-3: Technical requirements

ID	Title	Priority	Goal	Testbed	Fulfilment
FU.1	Support of large number of devices	Mandatory	Up to 60.000 devices	2.1,2.3	Done
FU.3	Support for diverse M2M services	Mandatory	Support for multiple applications	2.1, 2.3	Ongoing
NT.1	Heterogeneous networks	Mandatory	Multiple capillary interfaces.	2.1, 2.2, 2.3	Done
NT.4	Support of multi-hop communication	Medium	Mesh Networking	2.1, 2.3	Done
NT.5	Half duplex operation of terminals	Mandatory	Transmitting with just 1 antenna	2.1, 2-3	Done

NT.9	Reliable delivery of a message	High	Connection oriented protocol at Transport Level for IP segment (M2M server<->Gateway) and ACK reception of unicast messages at capillary level (Gateway<->M2M devices)	2.1, 2.3	Done
NT.11	Traffic Aggregation	Medium		2.2	Done
NT.16	Support for dual stack IPv4/IPv6	Mandatory	Enable LTE-M devices to support both stacks	2.1, 2.3	Pending
NF.1	Scalability	Mandatory	Support for clustering features	2.1, 2.3	Evaluating
DV.1	Self organized M2M equipments	Mandatory	Autonomous network discovery	2.1, 2.3	Done
DV.4	Location Information	High	Semantic Database	2.2	Ongoing
DV.6	M2M Gateway detection and registration	Mandatory	Self configuration of capillary networks	2.2	Done
NF.3	Extensibility and adaptability	Medium	Procedures exportable regardless underlying technologies	2.2	Done
SV.3	Efficient provisioning of a set of M2M equipment	Mandatory	Procedures exportable regardless underlying technologies	2.2	Ongoing
SV.6	Security	Mandatory	Data Protection - Integrity	2.4	Ongoing

4. Testbed 3: Device Management

4.1 Overview and objectives

This testbed is composed of three subtestbeds in order to showcase novelties related to device management aspects:

- **Subtestbed 3.1** demonstrates a novel Lightweight Device Management message encoding which aims to optimize both the message payload size and the processing performance.
- **Subtestbed 3.2** implements a novel self-diagnostic which warrants device reliability and performance while enabling system scalability.
- **Subtestbed 3.3** is focusing on security aspects covering device pairing, authentication and provisioning in self organized capillary network.

The device management, self-diagnostic and security solutions above mentioned are generic, they can thus be applied to all EXALTED major use cases: Smart Metering and Monitoring, eHealth and Intelligent Transportation System. Scenarios coverage of this testbed is limited to Smart Metering and Monitoring use case.

Figure 4-1 provides a high level view of these subtestbeds making use of necessary hardware along with proposed algorithms.

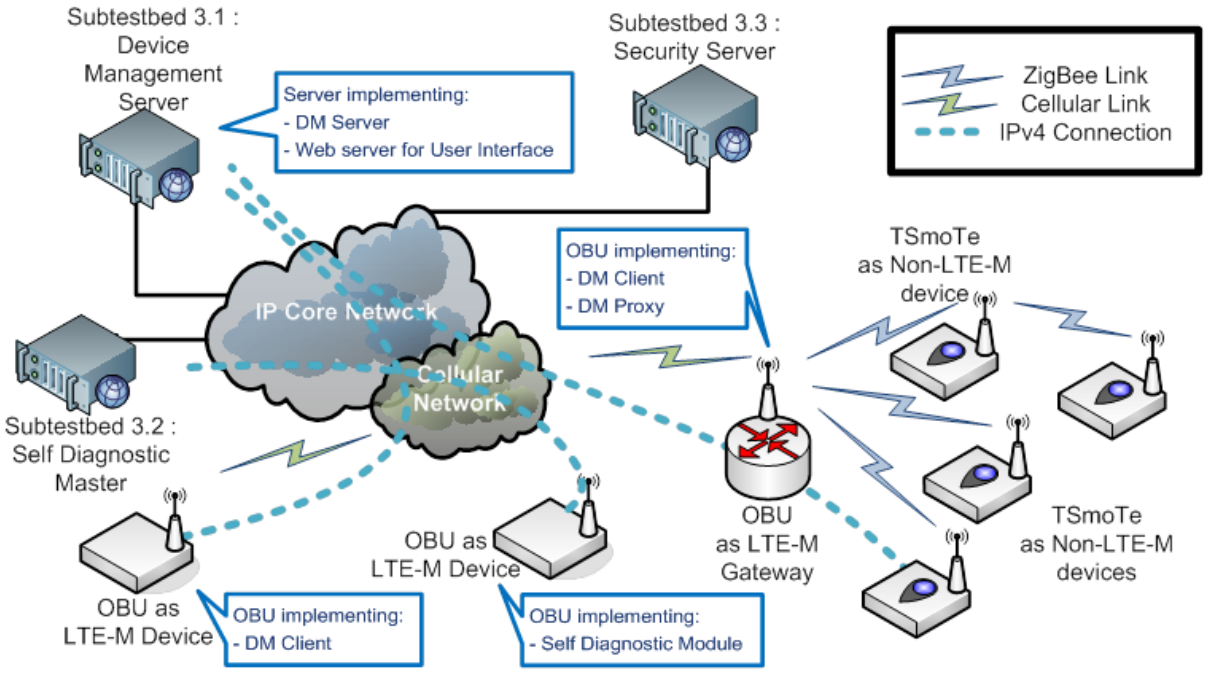


Figure 4-1: Overview of Device Management subtestbeds

Key objectives of subtestbeds are listed in Table 4-1 along with applicable EXALTED use cases. Further descriptions of objectives, components, interfaces, algorithms and supported features are detailed in separated sections pertaining to each subtestbed.

Table 4-1: Summary of Device Management oriented subtestbeds

Subtestbed	Objectives	Use Cases
3.1	• Lightweight Device Management • Device-to-device messaging	SMM

	• Data collect • Device configuration • Scalability	
3.2	• Root failure detection • Assisted healing • Spectrum efficiency	SMM
3.3	• Device pairing in self organize capillary network • Device bootstrapping • Credential management	SMM

4.1.1 Subtestbed 3.1: Lightweight device management

Device Management (DM) is essential in M2M ecosystem in order to remotely manage the life cycle of M2M gateways and devices. Open Mobile Alliance Device Management (OMA-DM) specification [28] is well established and is being used to manage over 1.4 billion mobile devices [29]. However, the current OMA-DM protocol (v1.x) is too verbose and requires the device to support XML which is memory consuming. Therefore OMA-DM v1.x is not suitable to manage constrained M2M devices which are mostly low cost devices; therefore they have low processing, power and memory capabilities. For this reason, OMA has created a work item, named OMA Lightweight M2M, to address this need. This new specification is not available yet, it is expected to be release in 2013. Furthermore it will not be backward compatible with the current OMA-DM v1.x.

4.1.1.1 Solution overview

This subtestbed implements a novel method to compress OMA-DM messages while maintaining full compatibility and full interoperability with existing OMA-DM v1.x servers. This novel solution, EXALTED Lightweight Device Management, is depicted in Figure 4-2 and will be implemented in the DM Subtestbed 3.1 for the following reasons:

- Stakeholder possible needs

  Service Providers wish to continue managing existing mobile devices and to launch new service supporting new M2M devices while minimizing the investment.

- Values added

  Stakeholder needs can be satisfied by using the proposed solution in which a proxy adapter is used to compress OMA-DM messages. As the existing mobile devices and OMA-DM server remains unchanged, the investment is therefore limited to the proxy adapter and device management client in M2M devices. Stakeholder does not have to develop and operate new servers (implementing the future OMA-DM Lightweight specification) to manage new M2M devices.

- Use Cases

  All device management functionalities derived from ITS, eHealth and SMM use cases can be fulfilled by this OMA-DM compliant solution, for instance, device configuration, software update, device monitoring, device self diagnostic, etc.

- Assessment of KPIs

  This subtestbed will also be used to assess the predefined KPIs: Actual payload size, transmission payload size, message encoding efficiency.

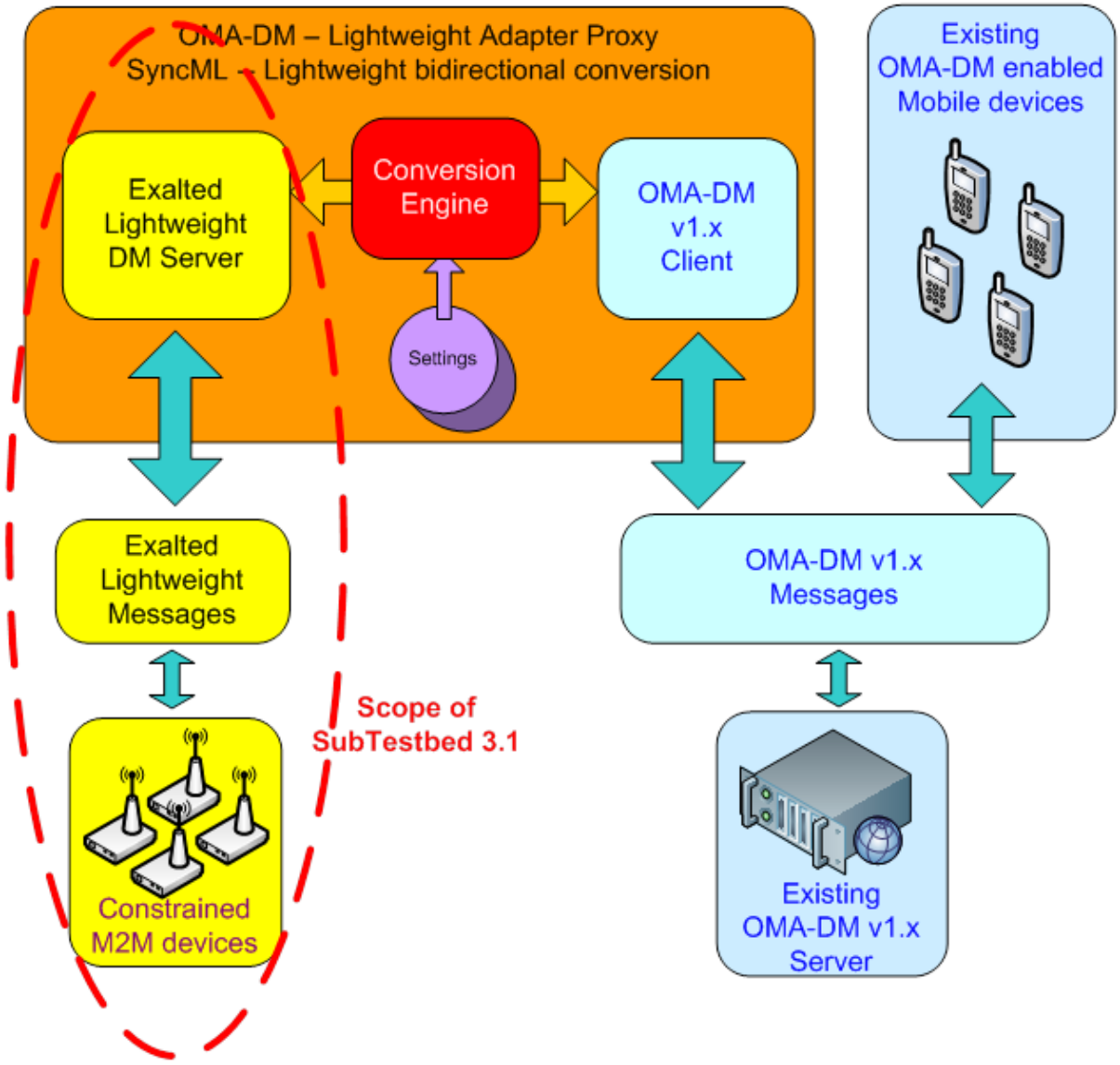


Figure 4-2: Overview of Device Management solution

4.1.1.2 Scope

The focus of Subtestbed 3.1 is restricted to EXALTED Lightweight DM aspects, as shown in yellow blocks in Figure 4-2:

- EXALTED Lightweight DM Server
- EXALTED Lightweight DM Client implemented in M2M devices

OMA-DM v1.x server, OMA-DM v1.x client, OMA-DM v1.x enabled mobile devices and Conversion engine are not part of this subtestbed, as most of these entities, but the conversion engine, are known art and widely deployed.

In order to showcase scenarios within an end-to-end perspective, an M2M application server will be developed. This latter will expose a user interface. EXALTED DM Client will be implemented on Android smartphone and OBU. Figure 4-3 depicts the entities used in this subtestbed.

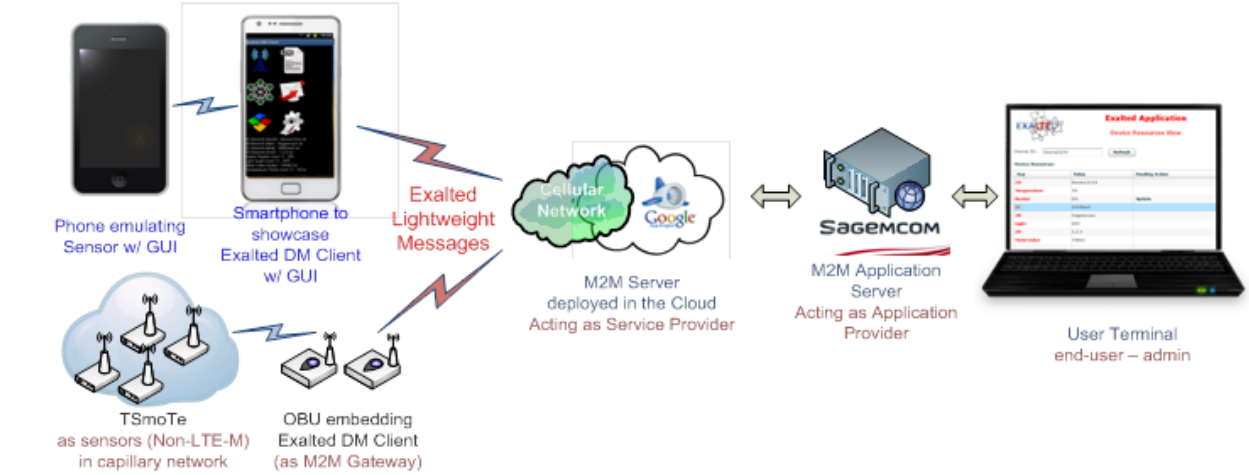


Figure 4-3: Subtestbed 3.1 entities

4.1.1.3 Objectives

Payload reduction

The implemented payload encoding aims to reduce the footprint: in Figure 4-2, the EXALTED Lightweight message size should be significantly smaller than OMA-DM v1.x message size to achieve a Reduction factor over 80%.

Use Case and Functions

The OMA-DM compliant DM solution can support all EXALTED major use cases. As stated in the EXALTED project report D7.1 [30], the Smart Metering use case will be developed and showcased. The following functions will be implemented, so that some smart metering scenarios could be emulated:

- Device attach/detach (sensor/actuator attach/detach to/from gateway)
- Device configuration e.g. data collect frequency
- Data collect. e.g. emulate the scenario of metering indexes collect
- Device control to emulate the power-cut scenario. Application to send power cut command to device. Turn off heater in case of power consumption peak
- Device-to-Device messaging to emulate the temperature auto-regulation scenario: sensor posting temperature information to another device

User Interface and Evidences

While the targeted scenarios will be emulated and functions will be shown, this subtestbed shall have the ability to show following evidences to user:

- Manipulation of device attributes, user shall be able to create, read, update or delete attributes.
- Device Management payload message, in order to assess the actual size and to view the attributes that have been changed by the user.
- A simulator will be used to emulate virtual devices/gateways connecting to the server. The intent of this simulator is to show the scalability.

4.1.2 Subtestbed 3.2: Self diagnostic

Self diagnostic in the device aims at providing answers to several EXALTED objectives: first and foremost the device reliability itself. Traffic reduction over the communication link is also

a very important objective due to the high number of devices considered in EXALTED. These objectives are addressed as follows:

- Improved device reliability through diagnostic production and resulting healing mechanisms enhances the device lifecycle while their automatic nature lowers maintenance cost.
- The “self” part in the device self diagnostic is the result of a service offload from the server side to the device side as EXALTED suggests performing diagnostic locally. No resource needed to perform remote diagnostic has to transit over the communication channel any more, as the diagnostic is now built as a local service. Only service requests are needed which limits the amount of data exchanged by nature and achieves maintenance traffic reduction over the main communication link.

The above points are actually addressed by three key concepts:

- Device root failure detection mechanism is driven by rules. New failure detection rules can be created by device manufacturers and remotely posted to devices by the M2M service provider. As a result, this dynamic and flexible mechanism enables stakeholder to optimize CapEx. Rules also include healing algorithms which are automatically applied by the embedded self diagnostic components to attempt to fix the detected failure. The remote M2M application can query diagnostic results for monitoring or dash board purposes.
- Assisted healing is an important value-added feature. In the event a device failure has been detected and cannot be fixed by the initial detection-healing rules, the M2M application or the service provider are notified and thus can provide further assistance or counter measure to fix the failure. Updated rules along with enhanced healing algorithm are then posted to devices.
- Distributed self diagnostic approach aims to offload the monitoring and diagnostic tasks on the M2M application or on the service provider. To achieve this, the root failure detection and assisted healing managers are deployed onto a tree hierarchy of nodes. Each node fan out the diagnostic and healing rules, received from upper level node, to child nodes or leaf nodes. On the application level perspective, the rule is only sent once to the top parent node, which propagates it to lower level nodes. The traffic over the main link, namely between the application and the top parent node, is therefore significantly reduced. The diagnostic and healing management is deployed and delegated to the subsequent parent nodes. Each of these latter nodes collects and aggregates diagnostic results and alarms from lower level nodes, and forwards them to the adjacent parent node. The aggregated data is ultimately collected at the top level parent node and sent to the application over the main link. The two ways data traffic and signalling reduction can thus be achieved.

4.1.2.1 Scope

This subtestbed has limited scope, namely, only the device root failure detection and assisted healing concepts will be implemented and showcased.

The distributed model, delegation and aggregation aspects on the third concept are not implemented. This subtestbed focuses the effort on the former point.

4.1.2.2 Objectives

Diagnostic and healing manager will be developed and hosted in an application. It exposes a user interface to showcase diagnostic and healing functionalities. In the real world, the core engine of this manager will be deployed on the M2M application server, on the service provider server and on nodes. This entity is referred as Remote Application in the remaining of this report.

The client side of diagnostic and healing algorithms are implemented on the OBU. In the real world application, these client algorithms are hosted in all devices. This entity, namely OBU, is referred as Device in the remaining of this report.

Device root failure detection and assisted healing scenarios will be showcased using the remote application and OBU entities.

4.1.3 Subtestbed 3.3: Secure element device management

The purpose of this subtestbed is to provide a mean to set keys in the Secure Element to secure the data messages exchanged between the end-device and the M2M server. The main algorithm is the group key setting that enables to set a common key for a group of devices. This notion of group key is a generalization of a shared key that is just known by two peers, typically the device and the server, to secure the communication between both ends. By sharing the same group key among various devices there are less keys to administrate on the server side which address a scalability objective and it enables a possible communication between a device's community if required by the application.

Another mechanism enables the M2M server to replace a key when it is necessary. This algorithm can be generalized to a new key setting which means that on request the M2M server may create new groups among devices managed by the server. It is possible to create a group key for devices part of different capillary network for example.

The group key setting is more secure that the so-called pairing mechanism. Pairing establishes shared keys between both ends that do not share any pre-arranged secure key material. Indeed in all this section it is assumed that some data have to be prepared both in the secure element before deployment and on the server to enable the group key setting. The advantage of this pre-personalization is that it prevents the "Man in the Middle" attack. This secure approach comes with the cost of the device pre-personalization and the server provisioning job that must be done in background before deployment.

The group key setting is directly managed by the server and there is no group key leader notion except the server itself.

A Device Management Agent is required on the device to run management scripts. The design has been made to avoid specific business operation implemented in this Agent to keep it as generic as possible.

With this subtestbed new devices can be inserted in such a way that secure communication is possible with a peer. There is no distinction between a Non-LTE device and an LTE device regarding the Secure Element management because the gateway is only there to route messages to the right end without interpretation of the semantic of the message.

4.2 Components and interfaces

4.2.1 Subtestbed 3.1: Lightweight device management

This section describes the components required in entities depicted in Figure 4-4.

Figure 4-4 shows how components (yellow blocks) are deployed in each physical entity (blue box). Components are linked together through interface (i1 – i6). Note that there are internal interfaces (black colour). The end-user or administrator interacts with the application via the user interface i1.

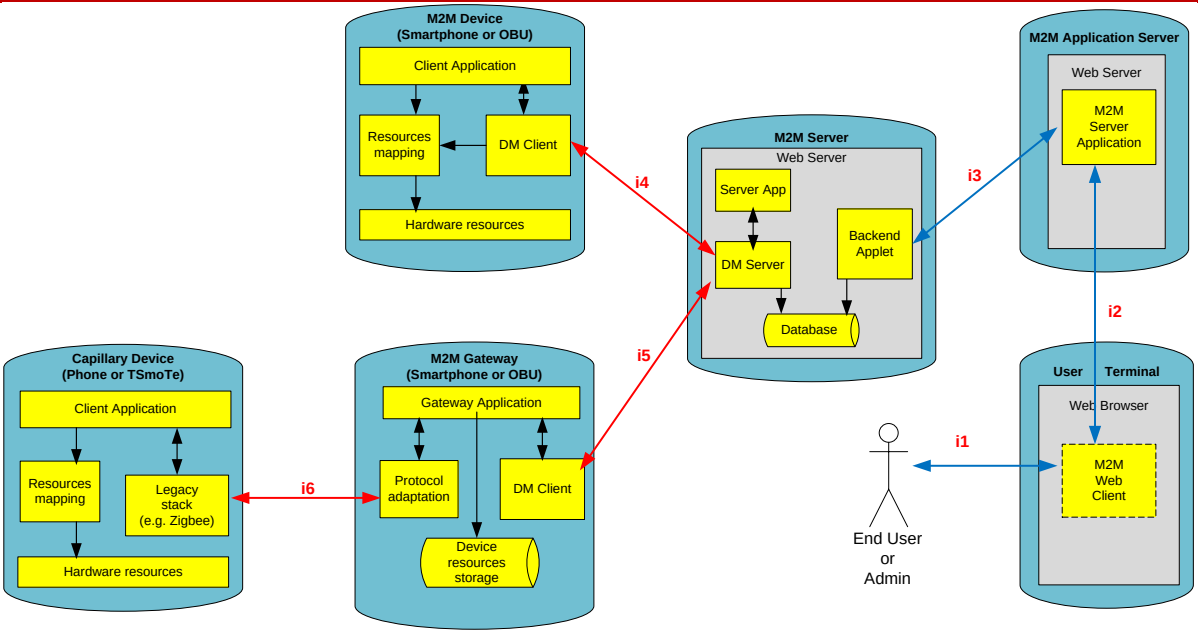


Figure 4-4: Components and interfaces

There are two main communication flows:

- Communication between Devices or Gateway and the Device Management server (red interfaces i4-i6)
- Communication between Device Management server and the Application server (blue interface i3)

Descriptions in A, B and C provide a high level view of the communication flows:

A: Device to Gateway to Device Management Server, upstream communication

1. Capillary device (Non-LTE-M device) posts requests or data over interface i6 to the M2M Gateway:
    - Requests could be pairing request or any control flows
    - Data could be device sensor data (e.g. temperature)

   Interface i6 is usually associated to short range communication protocol type e.g. ZigBee
2. Upon receiving data from the capillary device, the M2M Gateway performs a protocol translation in order to retrieve the data, the sender and the recipient. Decoded data is then stored on gateway local structure storage (device resources storage) and marked as a pending upstream task.
3. On the next connection to the DM server (e.g. next device wake up or scheduled connection time), the gateway will retrieve all pending upstream tasks from the device resources storage, and push them to the device management client. Capillary device data can therefore be aggregated and posted to the remote server over interface i5 which implements the Lightweight device management protocol.

   Likewise, M2M device does not have to cope with steps 1 and 2. M2M device data is posted to the remote server over the interface i4 using the lightweight device management protocol
4. The device management server authenticates the M2M gateway and stores the capillary device data on database.

5. The device management server retrieves the pending downstream tasks which are set by the application per end-user instructions (refer to C.). Downstream data is intended to the on-line gateway or to devices being managed by the connecting gateway is sent back as a response over interface i6.

B: Device Management Server to Gateway to Device, downstream communication

Downstream communication reflects the reverse flows and operations, similar to those in the upstream case.

C: End-user to Device Management Server

1. The end-user uses a web browser to connect to the application server over the user interface i1. The user can query for devices.
2. The query request is sent to DM server over interface i2 then i3. The backend servlet handles the request and searches for the device attributes (e.g. sensor data, actuator status, and configuration) in the database. Device data is returned as XML format and displayed on the user interface.
3. The user has the ability to alter editable attributes, create new attributes or delete attributes.
4. Requests submitted by the user on step 3 are sent to the application server over interface i2. For simplicity, the application server forwards the requests to the DM server over interface i3, which are processed by the backend servlet of DM server. Device or gateway attributes are formatted as XML and are exchanged on interface i3 over http. Device Management protocol is not needed.
5. Pending operations created in step 4 by the user are persisted in DM server database. These operations are relayed to the targeted device on step A-5.

Note that M2M application server does not need to implement Device Management protocol in order to manage remote devices. Device Management protocol is handled by the M2M server as a value added exposed by the service provider.

The next section gives more insight on how algorithms are put together to implement device management features.

4.2.2 Subtestbed 3.2: Self diagnostic

Self diagnostic in the device aims at improving the reliability of the devices providing them means to evaluate their own status and to initiate self preserving mechanisms that allow the whole system to maintain service continuity. Such an approach also helps reducing the maintenance traffic offloading to the device tasks such as diagnostic and remote code execution that were otherwise performed on the server side. Self diagnostic aims at automating these tasks introducing device embedded software that makes device diagnostic a target hosted service. The self diagnostic specifications in EXALTED D6.3 [31] went a bit further defining this framework as a basic component in distributed M2M network of devices diagnostic.

Self diagnostic is based on the Self Diagnostic Manager (SDM) module hosted in the device. This module provides diagnostic service to a variety of local (e.g. also embedded in the device) or remote applications and interfaces with the device's operating system as well as the device context formed by any available Hardware Abstraction Layer (HAL) or Application Programming Interface (API). The device context provides the base for status evaluation, the result of such diagnostic triggers a reaction ranging from a lack of reaction to self preservation or healing initiatives.

Figure 4-5 represents the basic components interacting in a system that is self diagnostic capable.

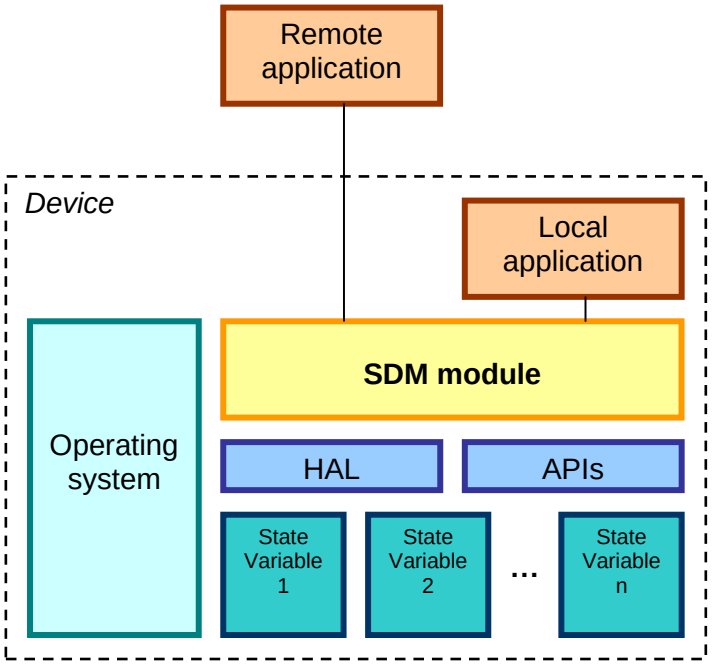


Figure 4-5: Architecture of a device hosting a self diagnostic manager module

In the architecture detailed in, the SDM module offers diagnostic service to local or remote applications. The diagnostic is built as a logical combination of state variables (e.g. battery level, communication error rate, etc) exposed through HAL or APIs that forms rules of diagnostic. These rules do not form a static set; they can be updated to better match the reliability requirements for the device.

As a response to the diagnostic request, the SDM returns aggregated diagnostic results to the application. This latter traffic over the main link is reduced to the minimum as the diagnostic is performed internally. There is no need to transmit over the main link the set of resources needed to perform the usual server side device diagnostic.

4.2.3 Subtestbed 3.3: Secure element device management

This subtestbed is based on the same components as the Subtestbed 2.4. Please refer to section 3.2.4 for the description of the components.

4.3 Algorithms and features

4.3.1 Subtestbed 3.1: Lightweight device management

The following features are envisaged in order to emulate some smart metering scenarios. They could be grouped in three different categories:

1. Device Management
    - Device attach/detach (sensor/actuator attach/detach to/from gateway)
    - Device configuration, e.g. data collect frequency
    - Device control to emulate the power-cut scenario. Application to send power cut command to device. Turn off heater in case of power consumption peak
2. Data transfer
    - Data collect, e.g. emulate the scenario of metering indexes collect

3. Messaging

- Device-to-Device messaging to emulate the temperature auto-regulation scenario: sensor posting temperature information to another device

Usually, in order to implement the above features, the device or the gateway would have to embed three different protocols. For instance, OMA-DM for device management functions, FTP for data transfer (e.g. data collect, firmware download) and Message Queue Telemetry Transport (MQTT) [32] for messaging (publish/subscribe). Constrained devices are not likely to embed different protocol clients. In addition to this high complexity and high resource demanding, multiple security schemes may be required to secure each protocol channel.

In Subtestbed 3.1, only the lightweight device management protocol is implemented to support all above features. Data transfer and messaging features can be enabled with the proposed lightweight device management protocol.

4.3.1.1 Data transfer

Data transfer feature is enabled in this subtestbed using custom data objects. Smart metering data objects (e.g. smart metering indexes, room temperature) can thus be created and collected. They are sent over the device management protocol similarly as for OMA Management Objects (MO). OMA has defined enabler to support Firmware Update Over The Air (FOTA), namely Firmware Update Management Objects (FUMO). Firmware can be transmitted to devices within the MOs.

Figure 4-6 below depicts how the data transfer transaction flows:

- Capillary devices post device data to the gateway over a legacy protocol. The temperature data is collected in a data object tree (similar to Management Objects); attributes are accessible through a defined URI based on the path in the data object tree. Device data is aggregated in the data objects tree.
- Based on a predefined scheduler, the gateway makes a connection to the M2M server using the OMA-DM device management protocol. Aggregated device data is then sent to the server in the EXALTED Lightweight message.
- The server stores and updates the data objects tree. This latter contains therefore a copy of all manage gateway data.
- On the user end point, upon user action, the M2M application server relays the request to the EXALTED DM server (the backend server).
- The backend server makes a query on the data objects tree, which is persisted in database, and returns the requested device data along the reverse chain back to the user interface.

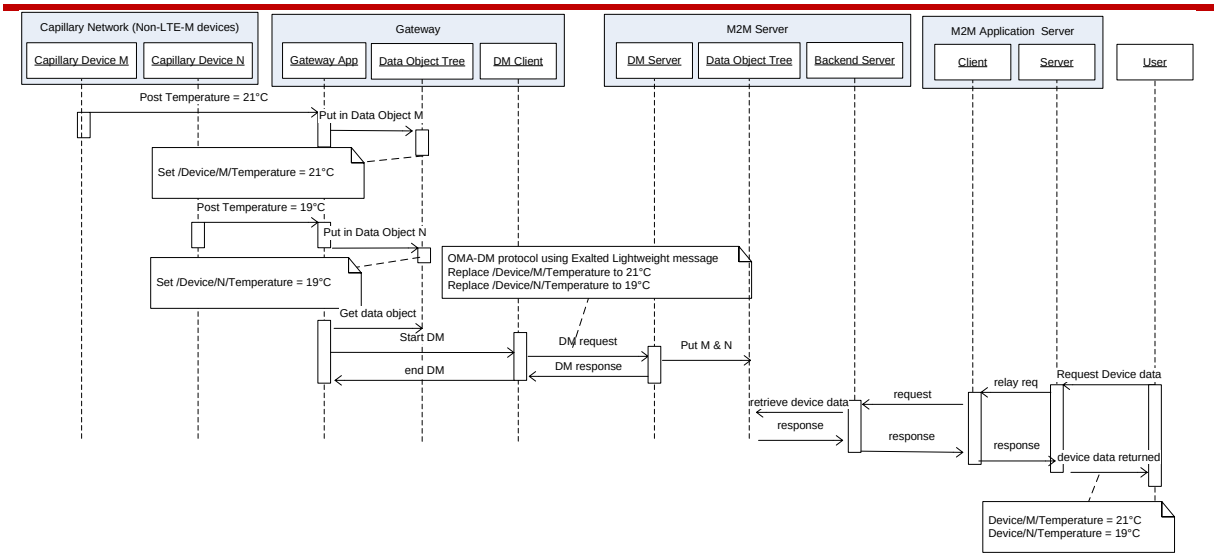


Figure 4-6: Posting and collecting device data - end-to-end sequence

4.3.1.2 Managing devices behind the gateway

Gateway keeps an inventory of devices attached to it. The inventory is reported to the DM server over the DM protocol. The inventory is updated whenever new devices are attached to the gateway or whenever devices are detached from it. This mechanism enables the DM server to locate a given capillary device based on the managing gateway. This is the foundation to build the device-to-device messaging enabler in subsection 4.3.1.3. The application is also aware of devices behind a particular gateway. Devices, attached to a gateway, are visible on the user interface of this subtestbed. This demonstrates the end-to-end information flow along the device “plug-n-play” feature.

OMA’s GateWay Management Objects (GwMO) is used to manage capillary devices.

4.3.1.3 Messaging

A simple device-to-device messaging mechanism can be enabled by defining a messaging data object for each device. The transmission mechanism of this messaging data object between end-points (device and DM server) is the same as for data transfer (subsection 4.3.1.1) using the device management protocol.

The messaging enabler is composed of:

- Messaging Data Object (DO)
- Messaging client and server

For this subtestbed, the Messaging DO is intentionally limited to the attributes following in Table 4-2:

Table 4-2: Messaging Data Object

Attribute	Mandatory	Comment
Sender	Yes	DeviceID of the sender
Label	Optional	Describe the sender
Recipient	Yes	ID of recipient (deviceID, serverID, groupID)
Message	Yes	The message itself

Further attributes may be defined to support advanced control and functions, such as Time To Live (TTL), topic name to support publish and subscribe messaging approach, etc. For this subtestbed, the messaging DO is limited to the above attributes set. The intent is to

showcase that device-to-device messaging can be supported over the lightweight device management protocol, for constrained devices.

Per OMA-DM enabler approach, sending a message requires the DM client to set one Replace command for each attribute through a URI in the request. In this simple case, four Replace commands are necessary in an OMA-DM message. For this proof of concept, to reduce the complexity and the size of payload, these four attributes are concatenated into one attribute using Comma-Separated Values (CSV) formatting. Thus, only one Replace command is needed in the message.

A messaging data object having one message is represented as follow:

MessageDO = {Sender;Label?;Recipient;Message}

Furthermore, a messaging data object containing more than one message is represented as:

MessageDO = {Sender;Label?;Recipient;Message}{Sender;Label?;Recipient;Message}{……..}{….} …{Sender;Label?;Recipient;Message}

Several messages can be serialized into one single data object. The transmission of this data object only requires one Replace command in the message. This approach leads to a significant payload reduction. Using binary format to serialize messages can further reduce the payload size.

The workflow of this messaging enabler is depicted in Figure 4-7.

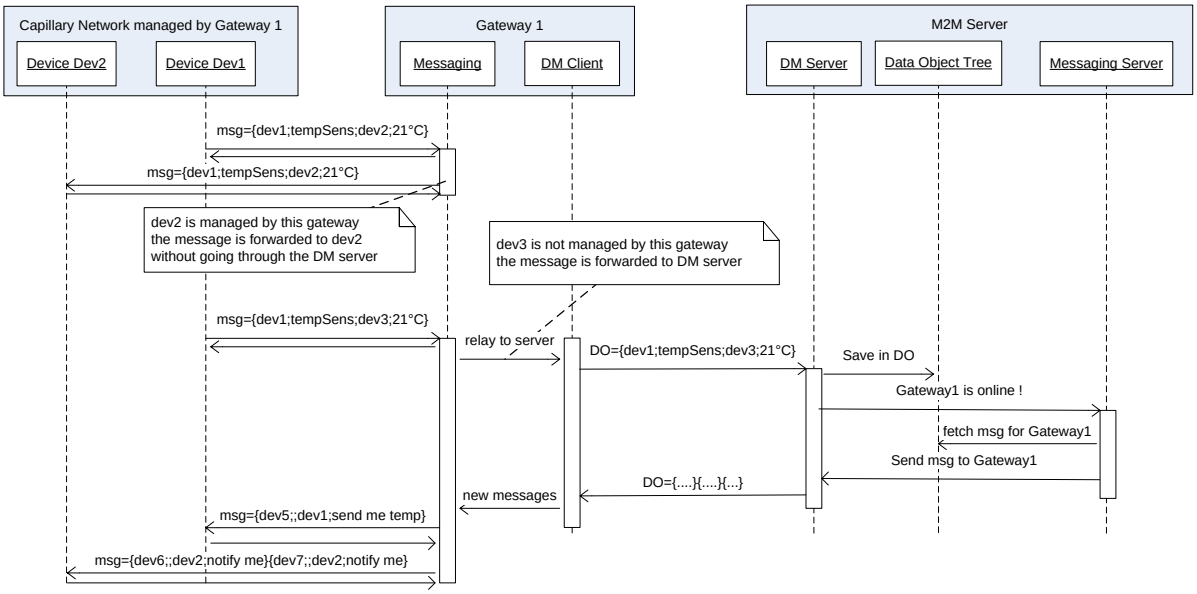


Figure 4-7: Device-to-device messaging

Device dev1 sends a message to device dev2 via the gateway. The latter forwards the message directly to device dev2, as it is within the same capillary network. This message does not go through the DM server, refer to Figure 4-7.

The workflow for Device dev1 to send a message to device dev3 via the gateway is as follows:

- Gateway1 is not aware of dev3, the message is forwarded to the DM server, over the DM protocol
- Upon receiving the message data object sent by Gateway1, DM Server saves it into the data object tree (same database as for the Data transfer case)
- DM server then informs the messaging server that Gateway1 is online. The messaging server search for pending messages, in the data object tree, that are intended to devices

managed by Gateway1. All pending messages are sent back to gateway as a response over the DM protocol

- Gateway1 dispatches the messages to proper recipients.

Note that the message sent by dev1 and intended to dev3, is stored in the data object tree. It will be fetched by the messaging server and delivered to the gateway that manages dev3.

4.3.1.4 OMA-DM v1.x compliant lightweight device management

All previously mentioned features are supported by this lightweight device management solution.

In order to achieve this compliance, the novel solution reuses the same protocol, security scheme and enablers as OMA-DM, for instance DM notification, setup phase, management phase, mutual authentication, and authentication challenge. This solution set the focus on payload reduction. This method will be detailed in [34] and named as ELFOMA (EXALTED Lightweight For OMA). An 85% average payload reduction can be achieved. The key compression concept is summarized here after. For further detail, please refer to the aforementioned report [34].

OMA-DM v1.x messages, as shown in Figure 4-2 must have a counterpart lightweight version, also known as aka EXALTED Lightweight messages, without losing information. Likewise, Lightweight messages originated by constrained devices must be translated to OMA-DM v1.x. This is the role of the Conversion engine, which is not developed in this subtestbed. This approach set the definition of the EXALTED Lightweight as follow:

- OMA-DM message is SyncML-base defined by SyncML Representation protocol [33]. Each SyncML element has a Content Model defining a list of mandatory or optional child elements along with multiplicity option. Each SyncML and its child elements are then serialized to OMA-DM messages based on a Document Type Definition (DTD).
- EXALTED Lightweight message is serialized based on a custom DTD. To maintain compatibility, SyncML elements are reused, that is, Content Models are reused. However child elements of a SyncML element may be ordered differently. Least frequently use child elements can be placed at the end of the model.
- ELFOMA defines a custom DTD so that the EXALTED Lightweight messages are formatted CSV (Comma Separated Values).

Examples below provide an overview of the payload reduction concept.

OMA-DM Content Model for Replace element:

Replace: (CmdID, NoResp?, Cred?, Meta?, Item+)

ELFOMA Content Model for Replace element:

Replace: (CmdID, Item+,Meta?,Cred?,NoResp?)

Note that child elements have been reordered. E.g. Cred and NoResp child elements are moved to the end.
Child elements marked with “?” are optional. E.g. Meta, Cred and NoResp are optional.
Child element “Item”, marked with “+”, can specify a list or an array of elements of type “Item”, without at least one element.

OMA-DM Content Model for Item element:

Item: (Target?, Source?, SourceParent?, TargetParent?, Meta?, Data?)

ELFOMA Content Model for Item element:

Item: (Target?,Source?,Data?,Meta?, SourceParent?, TargetParent?)

Note that SourceParent, TargetParent optional child elements have been reordered.

Figure 4-8 shows an excerpt of OMA-DM payload highlighting the Replace command element (1170 bytes):

```
<Replace>
  <CmdID>3</CmdID>
  <Item>
    <Source>
      <LocURI>./DevInfo/DevId</LocURI>
    </Source>
    <Meta>
    <Format xmlns='syncml:metinf'>Chr</Format>
    <Type xmlns='syncml:metinf'> text/plain</Type>
    </Meta>
    <Data>IMEI:493005100592800</Data>
  </Item>
  <Item>
    <Source>
      <LocURI>./DevInfo/Man</LocURI>
    </Source>
    <Meta>
    <Format xmlns='syncml:metinf'>chr</Format>
    <Type xmlns='syncml:metinf'>text/plain</Type>
    </Meta>
    <Data>Device Factory, Inc.</Data>
  </Item>
  <Item>
    <Source>
      <LocURI>./DevInfo/Mod</LocURI>
    </Source>
    <Meta>
    <Format xmlns='syncml:metinf'>chr</Format>
    <Type xmlns='syncml:metinf'>text/plain</Type>
    </Meta>
    <Data>SmartPhone2000</Data>
  </Item>
  </Item>
  <Item>
    <Source>
      <LocURI>./DevInfo/DmV</LocURI>
    </Source>
    <Meta>
    <Format xmlns='syncml:metinf'>chr</Format>
    <Type xmlns='syncml:metinf'>text/plain</Type>
    </Meta>
    <Data>1.0.0.1</Data>
  </Item>
  <Item>
    <Source>
      <LocURI>./DevInfo/Lang</LocURI>
    </Source>
    <Meta>
    <Format xmlns='syncml:metinf'>chr</Format>
    <Type xmlns='syncml:metinf'>text/plain</Type>
    </Meta>
    <Data>en-US</Data>
  </Item>
</Replace>
```

Figure 4-8: OMA DM message example with Replace command

The above OMA-DM payload is encoded as shown in Figure 4-9 using ELFOMA scheme (111 bytes):

```
20=3;{;2A;"IMEI:493005100592800"}
{;2B;Device Factory, Inc.}
{;2C;SmartPhone2000}
{;2D;1.0.0.1}
{;2E;en-US}
```

Figure 4-9: EXALTED Lightweight DM message example with Replace command

From Figure 4-8 to Figure 4-9 the payload size has been reduced by 90%:

- The replace command has been replaced by the corresponding WBXML token (token value is 20) [33] (WBXML Token definitions).
- The value of first child element “CmdId” is 3, in CSV format only the value is presented. The name of field (ChildItem) is discarded to reduce the size of the payload. ELFOMA identifies the “CmdId” element through its position in the content model.
- Meta, Cred and NoResp child elements of the Replace command element are optional and are not presented in the OMA-DM payload. In ELFOMA, they are placed at the end of the content model; they are mentioned in compressed version.
- A list of “Item” elements is presented using curly brackets.
- Inside each curly bracket, values of child elements of the “Item” element are listed. Missing values are left blank (e.g. Target?); default values can be left blank (e.g. Meta).
- Resource tokens 2A, 2B, 2C and 2D are used as shortcut to reference respectively /DevInfo/DevID, /DevInfo/Man, /DevInfo/Mod, /DevInfo/DmV. The Conversion engine in the proxy adapter performs this mapping.
- The proxy adapter will automatically assign pre-defined default values to child element being left blank. For instance, in the above example, the “Meta” field being left blank,

therefore the proxy adapter will assign the Format to “chr” and the Type to “text” along with the associated namespaces.

Another example of OMA-DM payload presenting the SyncHdr protocol element (541 bytes) is shown in Figure 4-10:

```
<SyncHdr>
   <VerDTD>1.2</VerDTD>
   <VerProto>DM/1.2</VerProto>
   <SessionID>1</SessionID>
   <MsgID>2</MsgID>
   <Target>
      <LocURI>
         http://www.syncml.org/mgmt-server
      </LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
   <Cred>
      <Meta>
         <Type xmlns='syncml:metinf'>
            syncml:auth-basic
         </Type>
         <Format xmlns='syncml:metinf'>
            b64
         </Format>
      </Meta>
      <Data>QnJ1Y2UyOk9oQmVoYXZl</Data>
   </Cred>
   <Meta>
      <MaxMsgSize xmlns='syncml:metinf'>
         5000
      </MaxMsgSize>
   </Meta>
</SyncHdr>
```

Figure 4-10: OMA DM message example with SyncHdr element

The Lightweight payload of the above SyncHdr sample (95 bytes):

```
2C=1.2;DM/1.2;1;2;mgmt-server;"IMEI:493005100592
800";{{1;1};QnJ1Y2UyOk9oQmVoYXZl};{;;1D=5000}
```

Figure 4-11: EXALTED Lightweight DM message example with SyncHdr element

The compression rate exceeds 82% for this example.

WBXML token for SyncHdr element is 2C:

- An alias “mgmt-server” is used to refer to the URI as specified in OMA-DM payload. The proxy adapter use this predefined setting to perform the mapping alias to URI.
- The Meta element is presented by {Type?;Format?;Data?} content model. Type is set to 1 (basic authentication), and Format is set to 1 (base64 encoding). These settings are defined in the proxy adapter which provides the mapping to the full OMA-DM Meta representation.
- The custom MaxMsgSize element is mapped to a token 1D, defined in proxy adapter.

For further detail, please refer to the EXALTED project report D4.3 [34].

Figure 4-12 unveils the compactness the ELFOMA encoding scheme. It achieves better results than others compression methods such as EXI, JSON, WBXML, GZIP. Furthermore, it has a very low encoding/decoding complexity as CSV parsing is straightforward while decoding EXI, WBXML, and GZIP requires much more device resources (memory, processing capabilities).

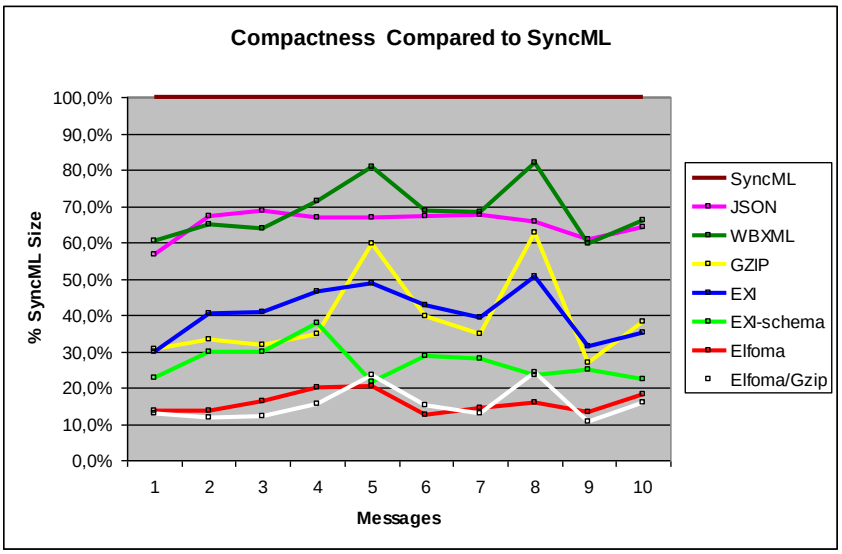


Figure 4-12: Compactness of EXALTED Lightweight messages

4.3.1.5 Screen captures and Simulator

Figure 4-13 below is a screen capture of the EXALTED DM client implemented on a smartphone, and of the M2M Application user interface.

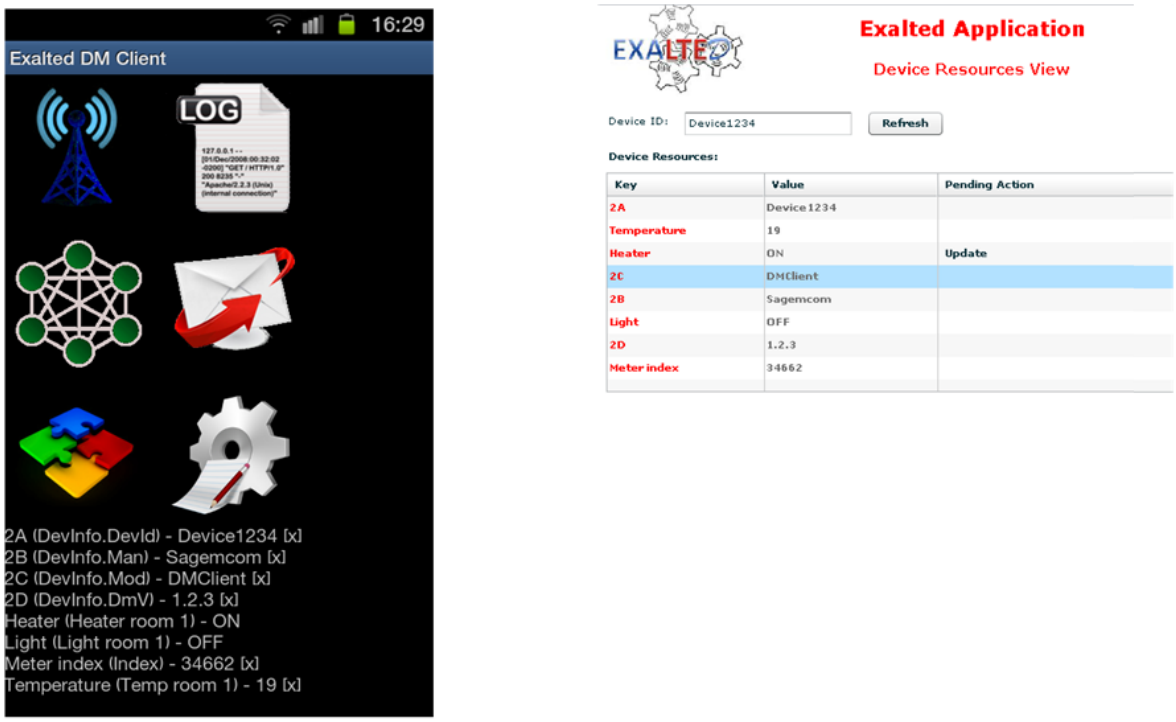


Figure 4-13: Screen capture of the DM client and M2M application UI

A special version of the EXALTED DM client running on a smartphone or on a laptop (console) will provide the feature to create a number of virtually devices connecting to the M2M server. The number of virtual devices is defined in the user interface.

4.3.2 Subtestbed 3.2: Self diagnostic

Figure 4-14 details the main blocks of a device hosted implementation of the SDM manager. The SDM agents hold the diagnostic rules that are applied by the Prolog block to the state variables that define the context of the device. The Prolog block is built with TU Prolog, a widely deployable Java based and light-weight Prolog. Access to the state variables exposed by the system or APIs is made through the Java Native Interface (JNI).

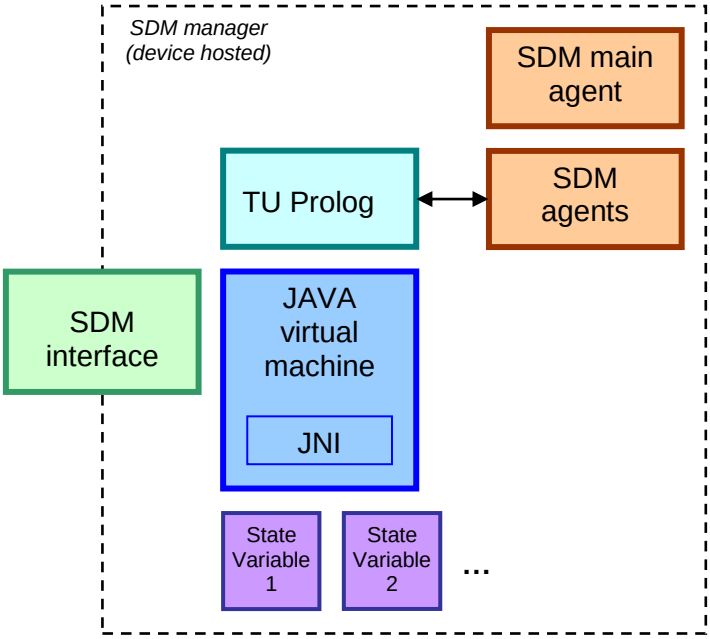


Figure 4-14: Main blocks of the self diagnostic implementation

Two diagnostic sequences will be particularly addressed here:

- Root failure cause detection: among a set of possible failure cause, the device is able to pick-up the root one.
- Assisted healing: after a report of inconclusive diagnostic, the set of diagnostic rules is remotely updated in order to strengthen the diagnostic range.

A remote application communicating with the SDM module through its SDM interface is also developed in order to demonstrate the benefits of the solution in terms of traffic over the main radio link. UDP is used as a transport protocol to connect the remote application to the SDM interface.

4.3.2.1 Root failure cause detection

This is the basic failure detection feature, the SDM as to be able to process the diagnostic rules and to detect the root failure cause when secondary failure causes are also reported in a hierarchical set of diagnostic rules.

As depicted in the project report D6.3 [31], the EXALTED approach for diagnostic rule design is a hierarchical one from top level rules that evaluate broad functional status to the lowest level rules that apply to the device's exposed state variables. As the rules "connect" in a recursive way and apply their logic to the results of lower level rules, the resulting rule set is similar to a hierarchical tree that guides the rule processing when performing diagnostic, starting from the lowest level rules and injecting their results to the immediately upper level rules and so on until the node corresponding to the diagnostic request is reached. In a failure scenario, the highest level node to report a failure in a branch provides the root cause.

The test scenario consists of triggering such errors in the testbed that error reporting in the rule processing procedure spread into the hierarchical tree of diagnostic rules and the device is able to identify the highest level failure reporting rule. It's a standard test as it covers the most basic function of self diagnostic but also one the most critical.

Figure 4-15 represents the exchange diagram between an application (local or remote) actively requesting a device diagnostic service and the SDM manager hosted in the device. Messaging between the SDM interfaces is not changed by the local or remote nature of the service requesting application, only the transport mechanism for SDM messaging changes to fit the system requirements.

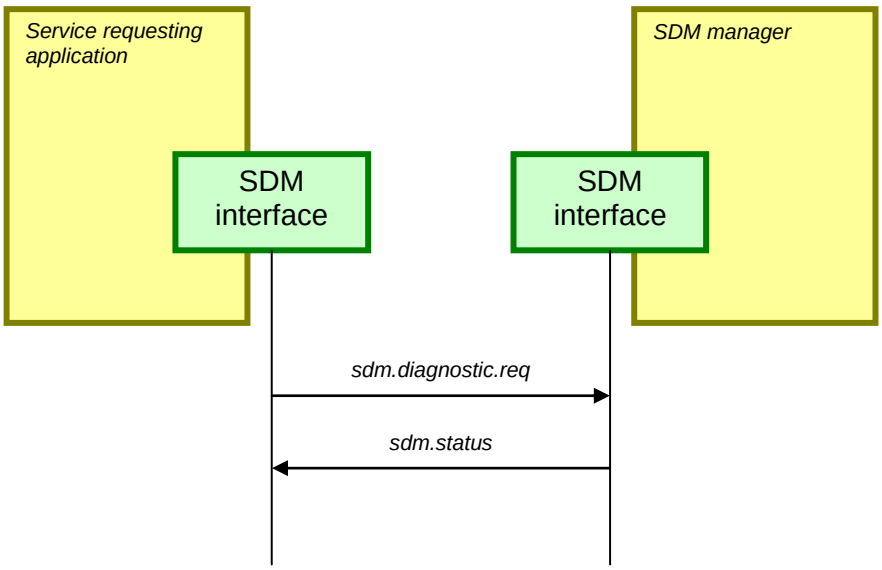


Figure 4-15: Exchange diagram for diagnostic service request

Note: The EXALTED project report D6.3 [31] provides more details regarding the SDM API.

4.3.2.2 Assisted healing

The rule approach for self diagnostic enables rule set management such as addition, removal, activation or deactivation of diagnostic rules during the whole active life of the equipment. The scenario addressed in the testbed is built upon the lifecycle of diagnostic rules. A device fails to identify the cause of a failure due to inadequate rule coverage and reports inconclusive result to a managing remote entity due. A remote application will assist the device updating the rule set so a relevant diagnostic can be performed.

In EXALTED's SDM architecture, diagnostic rules are handled by diagnostic agents. Actually, requesting a specific diagnostic service in the device is equivalent to running the associated diagnostic agent in the SDM. Updating rules in the device is equivalent to installing and activating one or several new diagnostic agents. In practice, diagnostic agents are files describing their associated rule as well as the eventual reactions resulting from the diagnostic.

The assisted healing scenario has a similar start as the root failure cause detection one. An application requires diagnostic service from the local SDM and gets a diagnostic result but the result is inconclusive as the installed set of rules can't identify the failure. The application decides to update the device rule set with a more complete one and transfers new diagnostic agent descriptors (DAD) files to the local DAD repository. Prior to execution, a diagnostic agent has to be activated. In order to save traffic, the activation process can be streamlined and the activation request attached to the installation procedure (e.g. an implicit activation request item can be found in the DAD file) to form a sequence of requests.

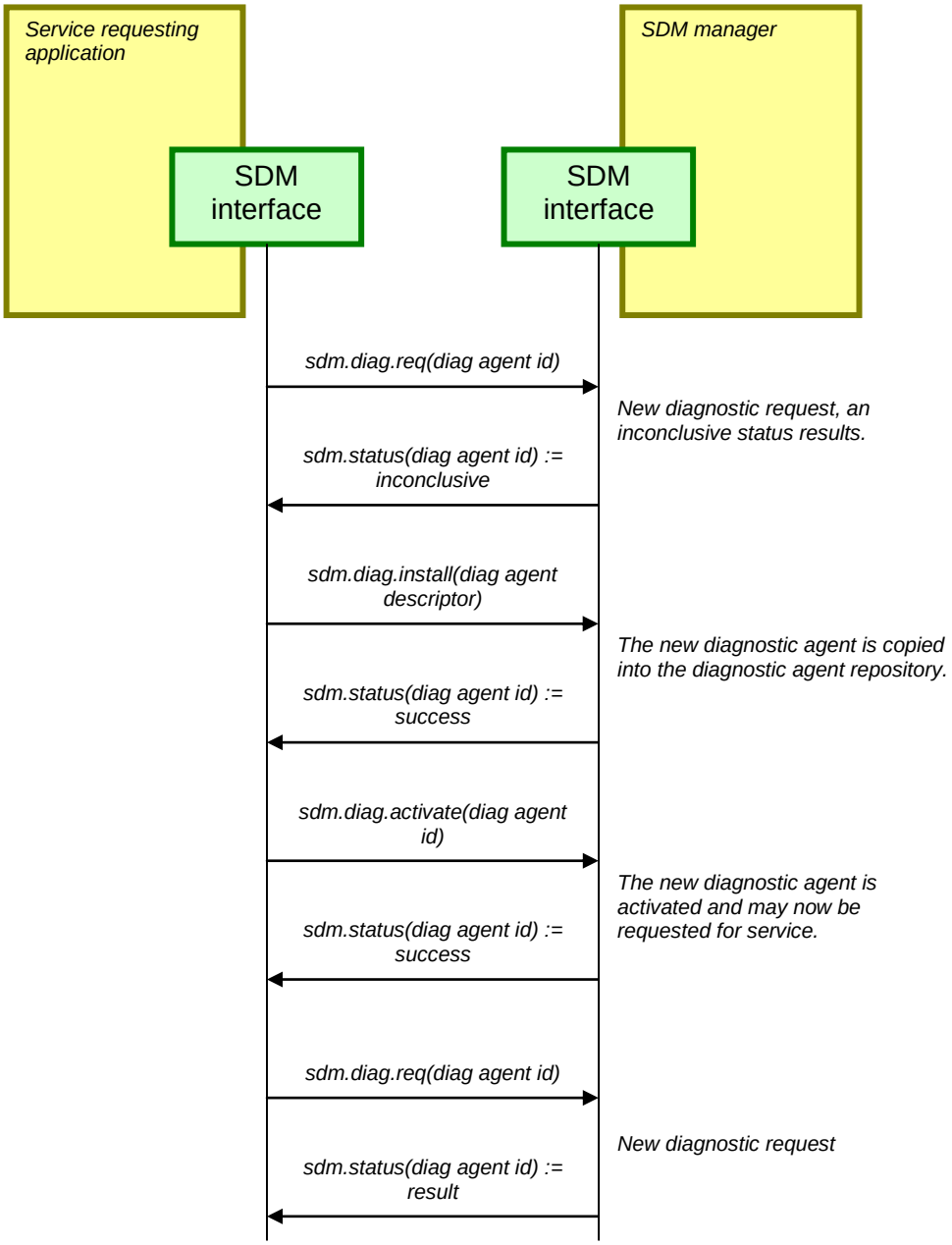


Figure 4-16: Exchange diagram for the assisted healing scenario

NB: Figure 4-16 displays several requests/answers transactions. Implementation wise, this may not ultimately translate into as many transactions on the communication link as the SDM supports transaction compression. A sequence of several requests can be defined and sent in a single transaction by the requesting application, the sequence will be processed sequentially by the target SDM as long as each request performs successfully and the SDM does not require additional input. In the end, the requesting application receives a global sequence success notification or an indication of the failed request if the sequence failed at some point.

4.3.3 Subtestbed 3.3: Secure element device management

With this subtestbed it is assumed that all devices do have a Secure Element to secure the communication and those cryptographic keys are stored in this Secure Element.

The gateway is supposed to be permanently connected with the M2M server with an IP link.

As a design principle driver it is stand that the gateway is not involved in the security messaging in the way it has not to be aware that messages are secured or not. It just sticks to its role to route message to the right end.

Secret keys are preferred to a security based on private keys and certificates also known as Public-Key Infrastructure (PKI). Reason of this choice is because secret keys are less demanding for the Secure Element both on computing power and memory size and much easier to administrate.

4.3.3.1 Group key setting

In the following both the EXALTED Secure Element or the SIM card of the LTE-M device are called the secure element.

A secure element has always been pre-personalized with a Factory key that is the same for a set of devices. Setting a shared secret between the secure element and the M2M server is less difficult than establishing a secret between two entities that do not share a secret in advance. It is possible to establish a common secret between two entities from no predefined secret: the Diffie-Hellman key agreement protocol [35] addresses this problem but it is sensitive to the “Man in the Middle” (MITM) attack.

Thanks to the Factory key that is prepared before the deployment of the device the group key setting algorithm is protected against the MITM attack. The strategy of the protocol is first to set a shared session key and then to send the group key to the device protected by this session key.

In the following Crypto-MAC refers to Message Authentication Code, not to the MAC address or it is explicitly named so. A Crypto-MAC is similar to a signature in a way that a message is hashed and then this hash is encrypted with a key. With the protocol below a symmetric key is used to encrypt the hash and with this regard is different form a digital signature where it must be the private key to be used to sign the hash.

A nice aspect of symmetric cryptography is that whatever sequence of bits can make a key. The handshake protocol describes in this section details how to generate keys from nonce exchanged between the secure element and the server.

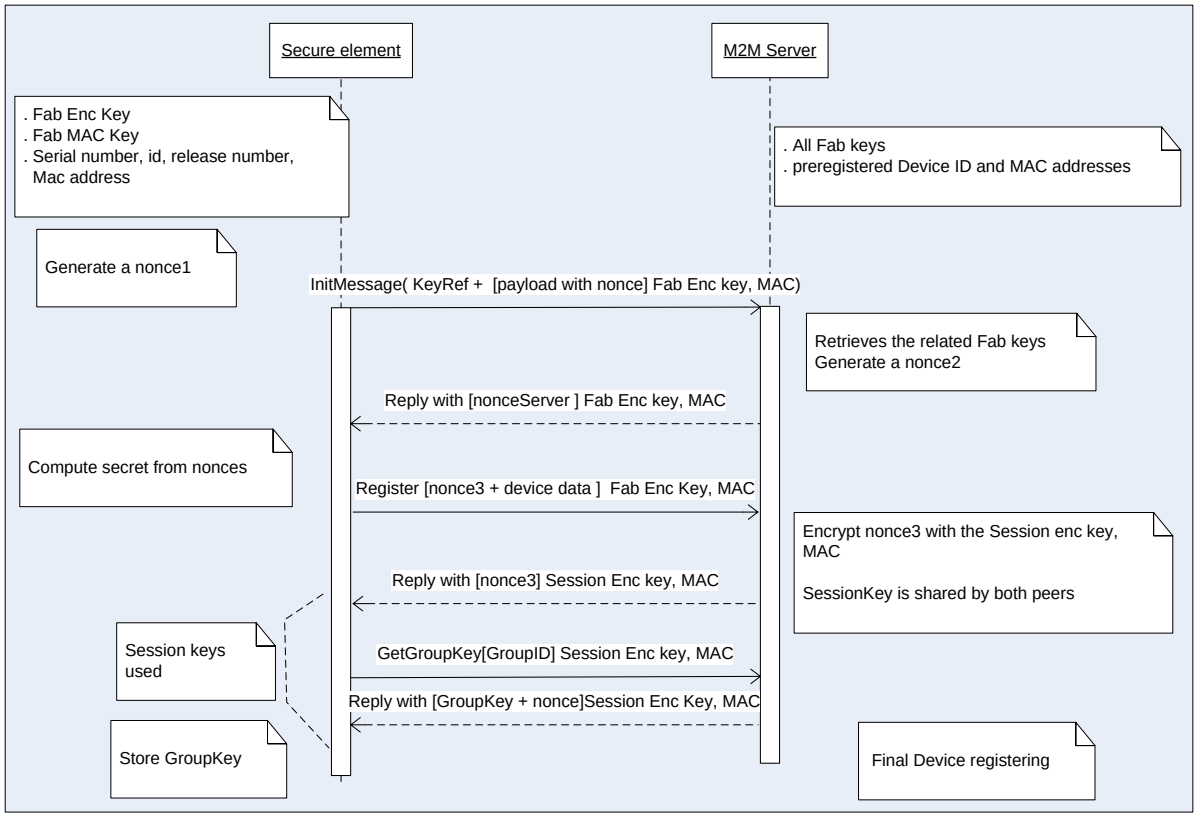


Figure 4-17: Group key setting in the Secure Element

First a nonce is sent to the M2M server protected with the Fab Keys. The device knows the server address either because it directly received this address with an SMS or because the gateway booted the handshake and provided the IP address of the M2M server.

The server answers to this InitMessage by providing its nonce. These two nonces enable to obtain both Session Encryption and Crypto-MAC key. The device sends a Register message with another nonce - called nonce3 in the diagram- and all devices data like the device ID, maybe the release number of the software, the MAC address.

Reply to this register message is the nonce3 protected with the session keys generated by both peers. Both the secure element and the M2M server share the same secret. Next message is the GetGroupKey that is given in the answer by the server. Both the message and the answer are protected with the session keys. The Group key is in the Secure Element; all messages exchanged from now on can be Crypto-MACed using this group key.

4.3.3.2 Secure element Management

There is no way for the M2M Server to initiate an IP communication because either LTE-M devices or LTE-M gateways have private IP address within the MNO domain. Best timing to process some management commands like key updates, software updates or whatever is at the end of an application messages exchange initiated by the LTE-M peer.

On the ending acknowledge message sent by the server, he must be able to specify that in addition to the application session ending with the current message there are some management commands to be run by the device. To keep this generic and to avoid designing specific business oriented messages the best choice is to send an URL of a script to be run by the end-device. Because this administration script might contain sensitive data it has to be protected. Again a group key is required to encrypt the script to be downloaded and run locally.

On the device side an Administration agent is required to download the script, send it to the secure element to be deciphered and then processed by the device. There is no real Device Management logic implemented in the Device Management Agent since its purpose is only to process the various commands of the script. Commands of the script identified so far are the listed hereafter:

- Send a message to a target
- Wait for an answer from a target
- On reception on an answer, send the answer to a target
- Ask the Secure Element to process a payload
- Send part of its binary code to the Secure Element.
- Perform commands received from the Secure Element (write, read, flash)

A target can be either the Secure Element or a remote peer, typically the MEM server

4.3.3.3 Key replacement

A Key encryption Key (KeK) is also send by the server to the Secure Element to be stored for a future usage during the group key setting. This KeK is used to encrypt a new key value that is sent to the Secure Element. This key replacement process is initiated by the M2M server to like any Device Management action as described in the previous section: a Device Management notification action is sent to the device together with an URL address on a final acknowledge message at the end of an application message exchange

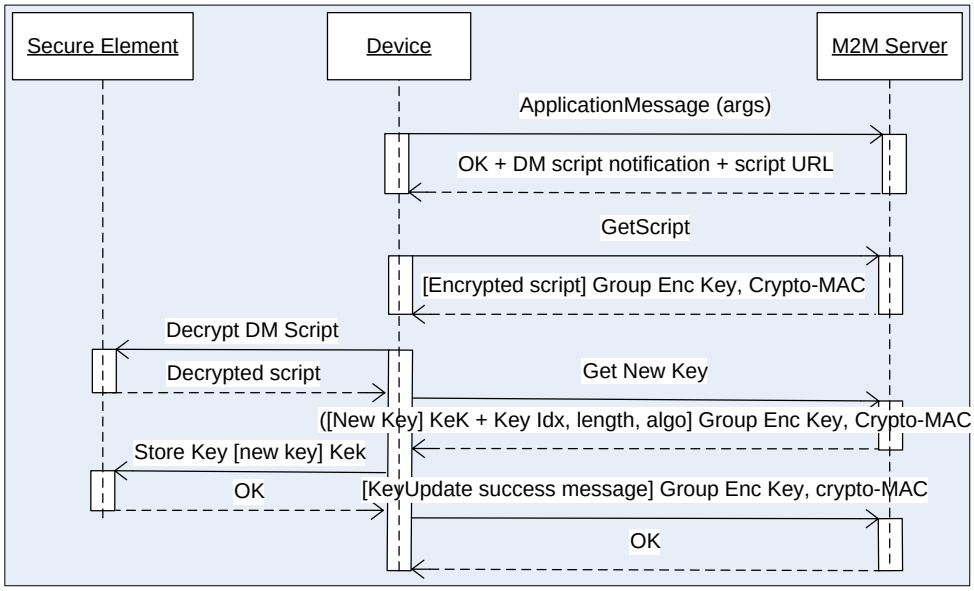


Figure 4-18: Key replacement sequence diagram

The Device Management Agent of the device downloads the Device Management script thanks to the URL passed. The first command of the management script is to get on the M2M server the new key that is encrypted with the KeK. The second command is to send this encrypted set of bytes to the Secure Element in a special command that tells the Secure Element that a key has to be replaced together with the key index, key length and the encrypted key algorithm. On successful replacement a message is sent back to the Device Management Agent that is also sent by the Device Management Agent to the M2M Server.

Note that the KeK is just one key among others in the Secure Element and that it can also be replaced if the server considers that it might be compromised and that it is more secure to replace it with a new key value.

This key replacement mechanism can be easily generalized in a new key setting mechanism by using a new key index instead of an existing one. This enables to set dynamically keys for new group or new service providers. This feature might also be considered as a possible security flaw and be configured only if the actor deploying the device as explicitly asked for it.

4.4 Tracing of technical requirements

Technical requirements pertaining to these three subtestbeds are listed in the below along the goal to be achieved and the status of fulfilment. A complete list of all technical requirements can be found in Annex A-1.

Table 4-3: Technical requirements

ID	Title	Priority	Goal	Sub-testbed	Fulfillment
FU.5	Local and remote device management	High	Lightweight protocol Spectrum efficient	3.1	Ongoing
FU.6	Unique identity for devices	High	Use Unique IDs to address devices and gateways	3.1	Done
FU.7	Security and provisioning	Mandatory		3.3	Ongoing
SV.3	Efficient provisioning of a set of M2M equipments	Mandatory		3.3	Ongoing
SV.6	Security	Mandatory	Secure Element Keys setting Secure Element management	3.3	Ongoing
NT.6	End to end device to device communication	Mandatory	Device-to-device, device-to-server, server-to-device messaging over the lightweight DM protocol	3.1	Done
NT.8	Mobility Management	Mandatory	Support mobility management at the application level	3.1	Done
NT.12	Self-diagnostic and self-healing operation	Medium	Failure detection, ability to inject new rule for self-healing	3.2	Ongoing
NT.13	Multicast and broadcast communication	Mandatory	Multicast message to devices behind a gateway	3.1	Ongoing
NF.1	Scalability	Mandatory	Increase scalability base on spectrum efficiency. Increase the number of managed devices under a given bandwidth	3.1	Evaluating
DV.2	Reliable M2M equipment	High	Failure detection mechanism	3.2	Ongoing
DV.7	Protocol translation at the gateway	Mandatory	Translate Lightweight DM commands onto device legacy commands	3.1	Ongoing
DV.10	Remote configuration	Mandatory	Device configuration over the lightweight DM protocol	3.1	Ongoing

5. Conclusion

Recapitulating the description of the three EXALTED testbeds in this report, it is stated that their specification is finalized. For each testbed a list of components was provided, and all interfaces between these components were defined. Furthermore, the realization of each testbed was justified by highlighting the different objectives and the added value that can be achieved. This was complemented by the presentation of the respective algorithms, protocols and features that are being implemented.

Based on these facts, the role of the testbeds towards the achievement of the overall project objectives is assessed in the following:

Apart from the mentioned benefits of Testbed 1 for the evaluation of the algorithms GFDM and CDMA, it clearly provides the proof of concepts that LTE-M is a system that coexists with LTE in the same spectrum using the same eNodeB hardware. It can be shown that the simultaneous transmission of LTE-M and LTE signals in the same frequency band, superimposed on the air, can be successfully separated at the receiver (LTE-M enabled eNodeB) and decoded independently from each other. Further it can be shown that the impact of LTE-M on the performance of LTE is only marginal.

The contributions of Testbed 2 to the project objectives are manifold. Firstly, a proof of concept for IP networking in M2M communications is given for various use cases. The connectivity and interacting of different types of devices within capillary networks is shown. Particular challenges are the mobility in the ITS use case, the envisaged scalability as well as efficient and fast routing. In one of the subtestbeds seamless connectivity is demonstrated explicitly for constrained low power devices, which is considered as one of the key challenges in EXALTED. The proposed solution is a lightweight address translation at the M2M gateway, which connects the IP world with the low power devices. With secure and reliable communications another overall EXALTED objective is addressed in Testbed 2. It can be shown that application payloads are successfully protected against attacks. Two different secure element form factors are used for this experiment.

With the help of Testbed 3, it is shown that the payload size of device management messages can be significantly reduced. This is a contribution to the project objectives energy efficiency, spectrum efficiency, and support of a big number of devices, given the fact that in the real system these messages have to be transmitted over the LTE-M air interface. The proposed self-diagnostic aims primarily at the objective to support fully autonomous devices equipped with self organizing and self healing capabilities for various use cases. Again, Testbed 3 contributes to the security framework, here in the context of self organizing capillary networks.

As a summary, it is claimed that all three testbeds together provide a significant contribution towards the achievement of the project objectives. Testbed experiments and measurements will complement the simulative and analytical evaluation of the proposed algorithms, protocols and features in the technical work packages. A detailed numerical evaluation and comparison with the technical requirements will be disclosed in the final deliverable entitled “Final proof of concept validation, results and analysis”

Annex

A1. Technical Requirements

After the key applications addressed in EXALTED were selected, the respective technical requirements were derived. These requirements are the background for design, specification and performance assessment of algorithms, protocols, and system concepts in WP3. The following Table A1-1 lists all technical requirements. An identification number is allocated to each requirement. The meaning of the acronyms is as follows:

FU Functional requirements
SV Service requirements
NT Network requirements
NF Non-functional requirements
DV Device requirements

Table A1-1: Full list of technical requirements

ID	Title	Priority	Dependencies
FU.1	Support of large number of devices	Mandatory	NT.10
FU.2	Efficient spectrum management	Mandatory	
FU.3	Support for diverse M2M services	Mandatory	
FU.4	Network initiated packet-data communication	Mandatory	
FU.5	Local and remote device management	High	
FU.6	Unique identity for devices	High	NT.16
FU.7	Security and provisioning	Mandatory	
SV.1	Overall QoS	Mandatory	NT.14, NF.5
SV.2	Allow multiple service providers on M2M devices	Low	SV.3, NF.3
SV.3	Efficient provisioning of a set of M2M equipments	Mandatory	NF.1, NT.13, DV.1, DV.10
SV.4	Change of subscription	Mandatory	SV.3
SV.5	Delegation and distribution of functionality	Mandatory	NT.4
SV.6	Security	Mandatory	
NT.1	Heterogeneous networks	Mandatory	
NT.2	LTE-M backward compatibility	Mandatory	
NT.3	Minimum number of modifications in network infrastructure	Mandatory	NT.2
NT.4	Support of multi-hop communication	Medium	
NT.5	Half duplex operation of terminals	Mandatory	
NT.6	End to end device to device communication	Mandatory	NT.4
NT.7	Flexible addressing scheme	Mandatory	NT.3
NT.8	Mobility management	Mandatory	
NT.9	Reliable delivery of a message	High	
NT.10	High node density	Medium	NF.6
NT.11	Traffic aggregation	Medium	
NT.12	Self-diagnostic and self-healing operation	Medium	DV.1
NT.13	Multicast and broadcast communication	Mandatory	
NT.14	End-to-end QoS system	Mandatory	NT.6, NT.8
NT.15	End-to-end session continuity	Mandatory	NT.6, NT.8, NT.9
NT.16	Support for dual stack IPv4/IPv6	Mandatory	NT.7
NT.17	Reduced signalling	Mandatory	DV.3
NF.1	Scalability	Mandatory	NF.6

NF.2	Energy efficiency	Mandatory	DV.3, DV.9
NF.3	Extensibility and adaptability	Medium	
NF.4	Real time performance	Medium	
NF.5	Congestion control mechanism	Low	
NF.6	Address space scalability	High	
NF.7	Control signalling integrity protection and encryption	Mandatory	
NF.8	Service provisioning for MNO/SP customers	Mandatory	SV.3
NF.9	Roaming support	Mandatory	NT.8
DV.1	Self organized M2M equipments	Mandatory	
DV.2	Reliable M2M equipments	High	
DV.3	Energy efficient duty cycles	Mandatory	NF.2
DV.4	Location information	High	
DV.5	Location locked M2M equipments	High	
DV.6	Gateway detection and registration	Mandatory	NT.13
DV.7	Protocol translation at the gateway	Mandatory	NT.1
DV.8	Information routing at the gateway	Mandatory	NT.1
DV.9	M2M equipment wake-up	Mandatory	
DV.10	Remote configuration	Mandatory	NT.12
DV.11	Software update over the air	Mandatory	

A2. Key Performance Indicators (KPIs)

Stepping towards a common evaluation methodology in the EXALTED project, a first list of key performance indicators (KPIs) was identified (Table A2-1)[2], in order to capture the wide range of the project's objectives, to serve as the basis for the assessment of the candidate technologies, techniques and system concept, and to quantify their impact on the overall performance of the EXALTED system. For LTE-M a subset of these KPIs will be utilized. This shortlist is still under discussion.

Table A2-1: Definitions of EXALTED KPIs

KPIs		EXALTED objective
Generally valid	**(K1) BER**: Bit error rate at the output of the decoder. **(K2) Packet Error Rate (PER)**: A packet represents the information block protected by CRC at the MAC layer. **(K3) Packet Loss Rate (PLR)**: As PER, but it only counts erroneous packets due to excessive latency. **(K4) Frame Error Rate (FER)**: A frame represents the information block protected by CRC at the RLC layer. **(K5) Outage probability**: probability of being excluded from the network either for battery or route reconfiguration.	
Evaluation of TX signal processing	**(K6) Peak-to-Average Power Ratio (PAPR):** Ratio of peak power and average power of the transmitted signal in the time domain. **(K7) Out-of-band radiation (OOB)**:	• Complexity reduction • improved resource management • energy efficiency

[2] The KPIs are the result of an inter work package collaboration, within the framework of the EXALTED evaluation methodology.

Evaluation of retransmissions schemes (ARQ, HARQ)	**(K8) Average number of retransmissions**: In case of erroneous transmissions, ARQ and HARQ mechanisms are used to retransmit packets until they are successfully received. **(K9) Reliability**: Average Number of Retransmissions.	• energy efficiency • signalling overhead reduction
Evaluation of transmission schemes with feedback	**(K10) Feedback bandwidth**: Required feedback data rate in bit/s.	
Evaluation of broadcast/multicast services	**(K11) Redundancy overhead** spent per user for reliable multicast message reception.	• energy efficiency
Evaluation of achievable data rates and spectral efficiency	**(K12) Throughput**: number of successfully received bits or messages per time unit in bit/s or messages/s. **(K13) Average packet call throughput** defined as $R_{pktcall} = \frac{\sum_k good\ bits\ in\ packet\ call\ k\ of\ user\ i}{\sum_k t_{end\_k} - t_{arrival\_k}}$ where k = denotes the k^{th} packet call from a group of K packet calls where the K packet calls can be for a given user i , $t_{arrival\_k}$ = first packet of packet call k arrives in queue, and t_{end_k} = last packet of packet call k is received by the UE. **(K14) Spectral efficiency (sum-rate)**: Number of successfully transmitted bits per time unit per frequency unit per cell in bit/s/Hz/cell.	• energy efficiency • signalling overhead reduction • improved resource management
Evaluation of achievable delays	**(K15) Average packet delay per sector**: The averaged packet delay per sector is defined as the ratio of the accumulated delay for all packets for all devices received by the sector and the total number of packets. The delay for an individual packet is defined as the time between when the packet enters the queue at transmitter and the time when the packet is received successively by the device. **(K16) E2E Delay/jitter**: Round trip time. **(K17) Access delay**: Needed time in order to join the network. **(K18) Bandwidth delay product**: Total available bandwidth * round trip time. Used for estimating the minimum buffer length needed in order to assure non-lossy transmission (it defines the maximum amount of data to be transmitted before receiving ACK or NACK confirmations). **(K19) Addressing translation delay**: Delay introduced by the needed processing time in order to map from IP to IEEE addresses. **(K20) Number of addresses mapped**: Maximum number of addresses supported when mapping IP addresses to non-IP ones on the M2M Gateway. **(K21) Handover delay**: Amount of time needed to leave a	

	network and join another one. **(K22) Percentage of satisfied users**: The percentage of users whose packets arrive at the destination within their maximum delay tolerance time interval	
Evaluation of number of supported users	**(K23) User per cell capacity**: Maximal number of simultaneously active users per cell. **(K24) CDF of number of served multicast users**.	• signalling overhead reduction • improved resource management
Evaluation of coverage and range	**(K25) Range**: Maximal possible distance between a M2M device and base station to enable communication with a given QoS, either directly or via a gateway or relay. **(K26) Coverage**: Percentage of area, where M2M devices can connect to a base station, either directly or via a gateway or relay.	• Het. network access management • mobility management
Particular evaluation of signalling overhead	**(K27) PHY Control channel and pilot overhead**: Percentage of radio resources utilized for signalling, control channels and pilots on PHY layer. **(K28) Paging efficiency**: Percentage of specific control channel information for paging procedures in bit/user. **(K29) Mobility management efficiency**: Percentage of specific control channel information for mobility procedures in bit/user. **(K30) Transmission Payload Size**: Size of the message exchange between 2 peers (e.g. device, cluster head, gateway, device management server). The size depends on the data encoding scheme, compression. **(K31) Payload Encoding**: Specify how device attributes, data are encoded and presented in the payload. **(K32) Actual Payload Size**: Size of the received message after decoding or decompression.	• mobility management • signalling overhead reduction • complexity, cost reduction
Particular evaluation of energy efficiency	**(K33) Mean power per signalling bit per user:** watt/bit. **(K34) Ratio between transmitted power and achieved throughput (energy efficiency)**: watt/(bit/s)=joules/bit **(K35) Consumed energy per message**: Sum of energy spent for signal processing and transmitted energy required for one message. **(K36) Standard Deviation of node energy levels**: This is an indicator of the variety of residual energy levels of nodes. We monitor this indicator to see how much energy equalization is achieved over time. **(K37) Average node energy levels**: This is an indicator that is used to monitor the network's overall energy consumption over time. **(K38) Coefficient of variation**: the ratio of (Standard Deviation of node energy levels/ Average node energy levels). **(K39) Network lifetime**: The time period until the first node depletes its battery energy.	• signalling overhead reduction • complexity reduction
Particular evaluation of complexity	**(K40) Complexity of encoding and decoding**, i.e. number of required multiplications. **(K41) Distortion**: Distortion is the performance metric used to measure how close an estimate is to its actual value. Typically, the distortion is measured by the	

	Mean Squared Error (MSE). **(K42) Number of CSI estimation:** the number of CSI estimation per decoded data bit. **(K43) Number of active antennas**: number of activated antennas compared to the available ones	
Evaluation of Radio resource management	**(K44) Radio resource consumption**: auto-diagnostic aims at reducing the amount of data exchanged between the remote device management server and the device moving one source of data transaction from the server side to the device side. **(K45) System resource consumption**: moving diagnostic to the device side introduces a new set of work in the device which means the system will run longer in order to perform the auto-diagnostic task. Energy wise, this is a cost and it should be minimized.	• Benefits of auto-diagnostic compared to a standard diagmon
Evaluation of Security	**(K46) Computational energy consumption:** This is the computational energy required to insure privacy, confidentiality and integrity of the data transmitted. It is related to the computational complexity of the algorithms involved **(K47) Radio energy consumption:** the radio energy required to insure privacy, confidentiality and integrity of the data transmitted. It is related to the data overhead required by the security process. **(K48) Infrastructure energy consumption:** This is the energy consumption required by the overall security layers adding up. It can be reduced by collapsing when possible different security layers into one**.** **(K49) Flexibility of the security enrolment process for capillary devices:** this indicator takes several parameters into account to reflect the overall flexibility of the security enrolment process for capillary devices **(K50) Total cost per user of the security solution:** This indicator(s) take(s) in to account the overall cost of the deployment of a security solution and(per client+infrastructure costs) and reflects it per user.	
Network Monitoring	**(K51) Query size:** Size of a query message exchange between two M2M elements (e.g. M2M device, cluster head, gateway, eNodeB, and network monitoring server). **Objective:** Shall be as small as possible. **Benefits**: Requires minimal memory, processing time and reduce transmission time (impacts: lower device cost, less energy consumption). **(K52) Passive monitoring:** Human manager submit the queries and perform analysis and management tasks). **Objective:** Monitoring should not be limited to passive monitoring and not depends on human intervention. **Benefits**: Passive monitoring introduces less overhead, minimal impact of memory and network traffic. **(K53) Centralised / hierarchical monitoring:** Centralised-processing approach requires continues polling of network health data from managed each sensor node in the network to the sink. **Objective:** Hierarchical monitoring (tasks are distributed among network managers, each manager reports to a higher level manager). **Benefits**: Centralised monitoring increases high data overhead, and this limits its scalability. Since individual node information is	Device / node monitoring mechanism to ensure that a response-to-demand datum is authentic reliable and secure

	important, aggregation solutions may not be applicable. In addition, in case of network partitioning, the nodes that are unable to reach the central sink are left without any management functionality. Local management tasks can be done at a lower level that reduces communication costs. Meanwhile global view of the network can still be available by reporting lower-level managers to higher level sink which can enable sink to make network-wide management control decisions. **(K54) Frequency of queries:** How often the queries need to be disseminated. **Objective:** For better energy-efficiency should be less frequent as possible. For more accuracy should be more frequent.	

A3. Mapping of requirements, use cases and testbeds

The main envisioned scenarios in EXALTED are specified in the report D2.1 [36] and they can be split into three main use cases:

- Intelligent Transport System (ITS)
    a. Remote Monitoring of Vehicle Data
    b. In-Vehicle M2M Diagnosis
    c. Railway Remote Monitoring and Failure Detection
    d. Parking Time Check
    e. Vehicle Collision Management
    f. Gateway vehicle for car-to-car communications
- Smart Metering and Monitoring (SMM)
    a. Energy Smart Metering – Building Management
    b. Industrial Monitoring
    c. Environmental monitoring
    d. Security – Surveillance
- E-Health Scenario

A first abstraction of use case into general requirements was carried out in the report D2.1 [36], where these were organized into five groups of technical requirements: Functional Requirements (FU), Service Requirements (SV), Network Requirements (NT), Non-functional requirements (NF), and Device Requirements (DV), see Annex A-1.

In this report, three different testbeds were described, each with differentiated goals aimed at providing the main functionalities of the EXALTED system concept. In this annex we analyse each use case individually, and provide a mapping to both requirements set forth in D2.1 [36] and to the functionalities that will be implemented on each of the testbeds. We begin with a summary overview of the use cases (presented in detail in D2.1 [36]) and then provide a mapping of the most relevant requirements for each use case.

A3.1 Use Case Implementation with EXALTED Architecture

A3.1.1 Intelligent Transport System

In the ITS scenario, a number of applications are defined, classified as *vehicle-to-application* and *vehicle-to-server.*

Vehicle-to-application:

A first use case consists in *remote monitoring of vehicle data*. In this case, sensors within a vehicle communicate with the vehicle gateway, which in turn communicates with the LTE-M eNodeB to communicate with a server connected in the infrastructure. Referring to the EXALTED architecture, this corresponds to an intra-vehicle capillary extension, connected to the LTE-M system via a gateway.

The second use case, *in-vehicle M2M diagnosis*, extends the first use case by adding functionality to the intra-vehicle capillary network. In this case, connection to an internal/external application server may also be carried out via the capillary area interface.

The third use case consists in *railway remote monitoring and failure detection.* This use case presents differentiated challenges with respect to the previous ones. Since its main goal is to detect safety-critical problems, latency requirements are critical. The latency requirement justifies an architecture where devices are directly connected to the eNodeB via the LTE-M interface. On the other hand, other non-critical applications may be connected via an onboard gateway in an architecture similar to the previous use cases.

Vehicle-to-vehicle:

The first vehicle-to-vehicle application is *parking time check,* where vehicles intercommunicate in a peer-to-peer fashion with a law enforcement vehicle to enforce parking regulation. This type of communication is performed through the capillary network and does not involve communication with the LTE-M core network.

The second application, *vehicle collision management*, involves communication between the car collision avoidance system and several sensors both within the vehicle and in nearby vehicles. It also involves communication with an application server which exchanges data with the collision avoidance system.

This application poses several communication challenges. First, latency and error requirements are high since it is a safety application. Second, assuming a capillary configuration of the vehicle sensors and collision avoidance system (gateway), it involves very low latency communication between different capillary networks corresponding to different nearby vehicles. In this particular application, using reserved LTE-M spectrum may provide additional guarantees on the latency and jitter, as long as the system is kept under sufficiently low load.

The third application, *gateway vehicle for V2V communications*, proposes a multi-vehicle capillary network, where a gateway vehicle concentrates traffic from several vehicles. While this configuration solves some of the communications problems raised in the previous one, it is very demanding in other requirements such as mobility support and security of the transmission.

A3.1.2 Smart Metering and Monitoring

In this use case, a first application, defined as *Energy Smart Metering – Building Management* must ensure secure transmission of data for the optimization of energy usage or other natural resources. The communication pattern is from devices (meters, sensors, actuators) to an application server, through both LTE-M and capillary networks. This use case may be delay tolerant; however a high degree of scalability is required for massive deployments. This use case does not typically require mobility support.

In *industrial* monitoring, remote control of devices and industrial automation are possible application fields. Both use cases can be implemented using the combination of LTE-M and capillary communication. In this application field low delay is typically required, as well as secure communications.

In *environmental* monitoring, scalability and traffic aggregation are dominant issues given the large amounts of data that can potentially be collected. Real time performance is less of a requirement.

Finally, *security and surveillance* require presents the most challenging set of requirements in the SMM field, including security and confidentiality of communications, real-time performance, scalability, etc.

A3.1.3 eHealth

The eHealth scenario consists of a heterogeneous set of applications related to the well-being of citizens. Typically, it consists of a capillary network and a gateway providing *connectivity* to application servers. It is also a very challenging use case in terms of requirements: security and confidentiality, real-time performance by some applications, mobility, location information, and scalability.

A3.2 Mapping of Requirements to Use Cases

The following Table A3-1 provides a mapping between the requirements set forth in D2.1 [36] and the use cases envisioned for EXALTED. The table highlights the most important requirements for each use case but does not imply that non-highlighted use case are not beneficial for a given use case.

Table A3-1: Mapping of requirements and use cases

REQUIREMENT			USE CASE										
			ITS						SMM				e-health
ID	Title	Priority	a	b	c	d	e	f	a	b	c	d	
FU.1	Support of large number of devices	Mandatory	x			x	x	x	x	x	x	x	x
FU.2	Efficient spectrum management	Mandatory	x				x	x	x	x	x	x	x
FU.3	Support for diverse M2M services	Mandatory	x	x	x	x	x	x	x	x	x	x	x
FU.4	Network initiated packet-data communication	Mandatory	x		x				x	x	x	x	x
FU.5	Local and remote device management	High	x	x	x	x	x	x	x	x	x	x	x
FU.6	Unique identity for devices	High	x	x	x	x	x	x	x	x	x	x	x
FU.7	Security and provisioning	Mandatory	x	x	x	x	x	x	x	x	x	x	x
SV.1	Overall QoS	Mandatory	x	x	x	x	x	x	x	x	x	x	x
SV.2	Allow multiple service providers on M2M devices	Low							x	x	x	x	
SV.3	Efficient provisioning of a set of M2M equipments	Mandatory	x		x				x	x	x	x	x
SV.4	Change of subscription	Mandatory							x	x	x	x	x
SV.5	Delegation and distribution of functionality	Mandatory	x		x				x	x	x	x	x
SV.6	Security	Mandatory	x	x	x	x	x	x	x	x	x	x	x
NT.1	Heterogeneous networks	Mandatory	x		x			x	x	x	x	x	x
NT.2	LTE-M backward compatibility	Mandatory	x		x				x	x	x	x	x
NT.3	NT.3 – Minimum number of modifications in network infrastructure	Mandatory	x		x				x	x	x	x	x
NT.4	Support of multi-hop communication	Medium	x		x		x	x	x	x	x	x	x
NT.5	Half duplex operation of terminals	Mandatory							x	x	x	x	x
NT.6	End to end device to device communication	Mandatory		x			x	x	x	x	x	x	x
NT.7	Flexible addressing scheme	Mandatory		x				x	x	x	x	x	x
NT.8	Mobility management	Mandatory	x		x	x	x	x		x		x	x
NT.9	Reliable delivery of a message	High	x	x	x	x	x	x	x	x	x	x	x
NT.10	High node density	Medium	x	x			x	x	x	x	x	x	x
NT.11	Traffic aggregation	Medium	x		x				x		x		
NT.12	Self-diagnostic and self-healing operation	Medium	x	x	x	x	x	x	x	x	x	x	x

NT.13	Multicast and broadcast communication	Mandatory				x	x		x	x	x	x	
NT.14	End-to-end QoS system	Mandatory	x	x	x	x	x	x	x	x	x	x	x
NT.15	End-to-end session continuity	Mandatory	x		x	x	x	x					x
NT.16	Support for dual stack IPv4/IPv6	Mandatory	x	x	x	x	x	x	x	x	x	x	x
NT.17	Reduced signaling	Mandatory		x	x				x	x	x	x	x
NF.1	Scalability	Mandatory	x				x	x	x	x	x	x	x
NF.2	Energy efficiency	Mandatory							x	x	x	x	x
NF.3	Extensibility and adaptability	Medium			x		x	x	x	x	x	x	x
NF.4	Real time performance	Medium		x	x	x	x	x		x		x	x
NF.5	Congestion control mechanism	Low	x		x		x	x	x	x	x	x	x
NF.6	Address space scalability	High	x		x	x	x	x	x	x	x	x	x
NF.7	Control signaling integrity protection and encryption	Mandatory	x	x	x	x	x	x	x	x	x	x	x
NF.8	Service provisioning for MNO/SP customers	Mandatory											
NF.9	Roaming support	Mandatory	x		x	x							x
DV.1	Self organized M2M equipments	Mandatory		x			x	x	x	x	x	x	x
DV.2	Reliable M2M equipments	High	x	x	x	x	x	x	x	x	x	x	x
DV.3	Energy efficient duty cycles	Mandatory							x	x	x	x	x
DV.4	Location information	High	x		x	x	x		x	x	x	x	x
DV.5	Location locked M2M equipments	High							x	x	x	x	
DV.6	Gateway detection and registration	Mandatory	x		x				x	x	x	x	x
DV.7	Protocol translation at the gateway	Mandatory	x		x				x	x	x	x	x
DV.8	Information routing at the gateway	Mandatory	x		x				x	x	x	x	x
DV.9	M2M equipment wake-up	Mandatory	x		x	x			x	x	x	x	x
DV.10	Remote configuration	Mandatory	x		x	x			x	x	x	x	x
DV.11	Software update over the air	Mandatory	x		x	x	x	x	x	x	x	x	x

A3.3 Implementation of Use Case Functionalities in Testbeds

The three testbeds developed within EXALTED aim at demonstrating the feasibility of using the techniques and algorithms designed within EXALTED to meet these requirements, tightly bound to the application scenarios considered in the project. Therefore, the approach taken in the implementation of use cases is an indirect one: the implementations satisfy the set of requirements, which in turn demonstrates the feasibility of the applications proposed.

The purpose of Testbed 1 is to demonstrate the generally valid functionality of LTE-M on PHY layer only. Therefore, it is not intended to validate the suitability of the algorithms with respect to one specific use case or application. Instead, the algorithms tested with LTE-M apply to all use cases and scenarios, as they apply to the technical improvements proposed at the PHY layer.

Testbed 2 is comprised of four sub-testbeds, whose goals and use cases are summarized in Table A3-2.

Table A3-2: Goals and use cases of Testbed 2

Subtestbed	Goals	Use Cases
2.1	• Scalability (support for billions of M2M devices) • Efficient routing for complex use cases • Mobility management • Reliability of data transferred • Real time operations	ITS and eHealth
2.2	• Heterogeneity • Energy efficiency • Implementation of complex gateway routines • Interoperability	SMM

2.3	• Scalability • Energy efficiency • Payload compression • Efficiency on capillary transmissions • Leverage of LTE-M data transferred	SMM
2.4	• End-to-end security	SMM

Testbed 3 is comprised of three sub-testbeds, whose goals and use cases are summarized in Table A3-3.

Table A3-3: Goals and use cases of Testbed 3

Subtestbed	Objectives	Use Cases
3.1	• Lightweight Device Management • Device-to-device messaging • Data collect • Device configuration • Scalability	SMM
3.2	• Root failure detection • Assisted healing • Spectrum efficiency	SMM
3.3	• Device pairing in self organize capillary network • Device bootstrapping • Credential management	SMM

Table A3-4 shows the matching between the technical requirements considered in the EXALTED and their relationship with the three testbeds being developed within the project (indicated as T1, T2, and T3).

Table A3-4: Matching between EXALTED technical requirements and testbeds

ID	Title	Priority	T1	T2	T3
FU.1	Support of large number of devices	Mandatory	X	X	
FU.2	Efficient spectrum management	Mandatory	X		
FU.3	Support for diverse M2M services	Mandatory		X	
FU.4	Network initiated packet-data communication	Mandatory			
FU.5	Local and remote device management	High			X
FU.6	Unique identity for devices	High			X
FU.7	Security and provisioning	Mandatory			X
SV.1	Overall QoS	Mandatory	X		
SV.2	Allow multiple service providers on M2M devices	Low			
SV.3	Efficient provisioning of a set of M2M equipments	Mandatory			X
SV.4	Change of subscription	Mandatory			
SV.5	Delegation and distribution of functionality	Mandatory			
SV.6	Security	Mandatory			X
NT.1	Heterogeneous networks	Mandatory		X	
NT.2	LTE-M backward compatibility	Mandatory	X		
NT.3	Minimum number of modifications in network infrastructure	Mandatory	X		
NT.4	Support of multi-hop communication	Medium		X	
NT.5	Half duplex operation of terminals	Mandatory		X	
NT.6	End to end device to device communication	Mandatory			X
NT.7	Flexible addressing scheme	Mandatory			
NT.8	Mobility management	Mandatory			X
NT.9	Reliable delivery of a message	High		X	
NT.10	High node density	Medium			
NT.11	Traffic aggregation	Medium			

NT.12	Self-diagnostic and self-healing operation	Medium			X
NT.13	Multicast and broadcast communication	Mandatory			X
NT.14	End-to-end QoS system	Mandatory			
NT.15	End-to-end session continuity	Mandatory			
NT.16	Support for dual stack IPv4/IPv6	Mandatory		X	
NT.17	Reduced signaling	Mandatory			
NF.1	Scalability	Mandatory	X	X	X
NF.2	Energy efficiency	Mandatory	X		
NF.3	Extensibility and adaptability	Medium			
NF.4	Real time performance	Medium			
NF.5	Congestion control mechanism	Low			
NF.6	Address space scalability	High			
NF.7	Control signaling integrity protection and encryption	Mandatory			
NF.8	Service provisioning for MNO/SP customers	Mandatory			
NF.9	Roaming support	Mandatory			
DV.1	Self organized M2M equipments	Mandatory		X	
DV.2	Reliable M2M equipments	High			X
DV.3	Energy efficient duty cycles	Mandatory			
DV.4	Location information	High			
DV.5	Location locked M2M equipments	High			
DV.6	Gateway detection and registration	Mandatory			
DV.7	Protocol translation at the gateway	Mandatory			X
DV.8	Information routing at the gateway	Mandatory			
DV.9	M2M equipment wake-up	Mandatory			
DV.10	Remote configuration	Mandatory			X
DV.11	Software update over the air	Mandatory			

A4. Examples for communication protocol sequences

A4.1 Smart Metering and Monitoring

Different Deliverables in the EXALTED have already described in detail the different set of requirements imposed to this use case. The scope of this annex is to provide a general view of the different messages, procedures and entities involved in the E2E communication for the applications based on these principles.

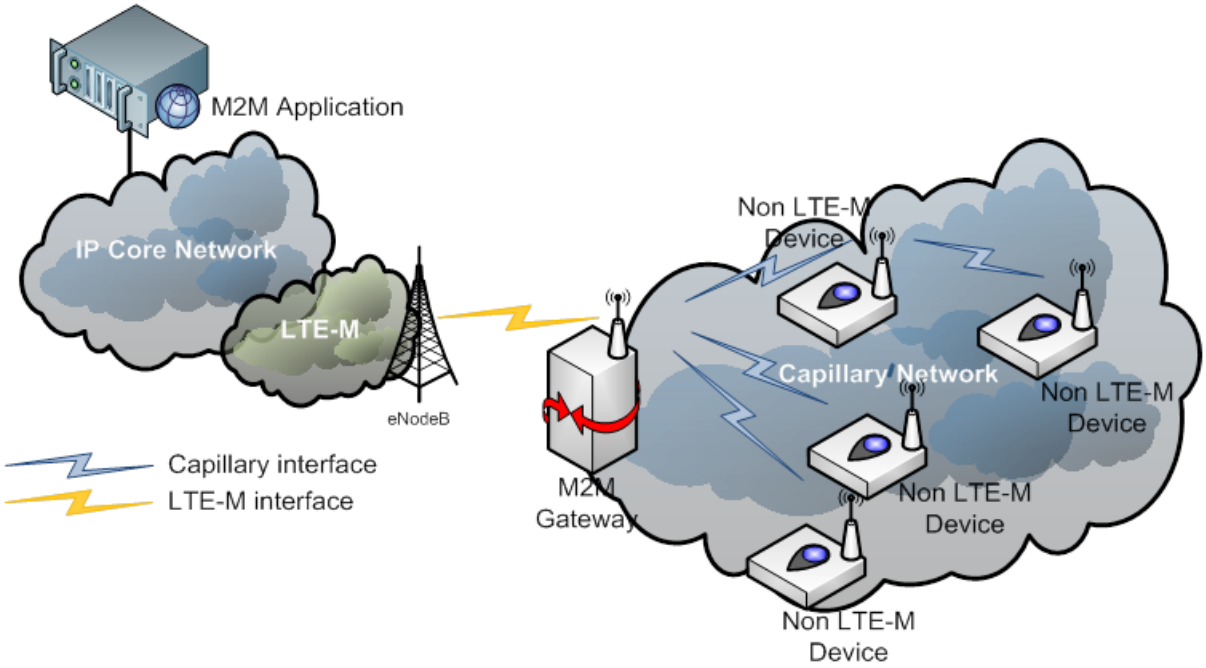


Figure A4-1: SMM use case

Figure A4-1 above depicts the different elements taking part in this use case. It is considered a big capillary network using a low-power communication interface, a gateway connecting public IP world with capillary network, the eNodeB to which the gateway is attached, the core network and the M2M application running on a server contactable by its IP. Since all of them have been presented in WP2 deliverables, this annex focuses on describing step-by-step the whole M2M scenario target of the EXALTED.

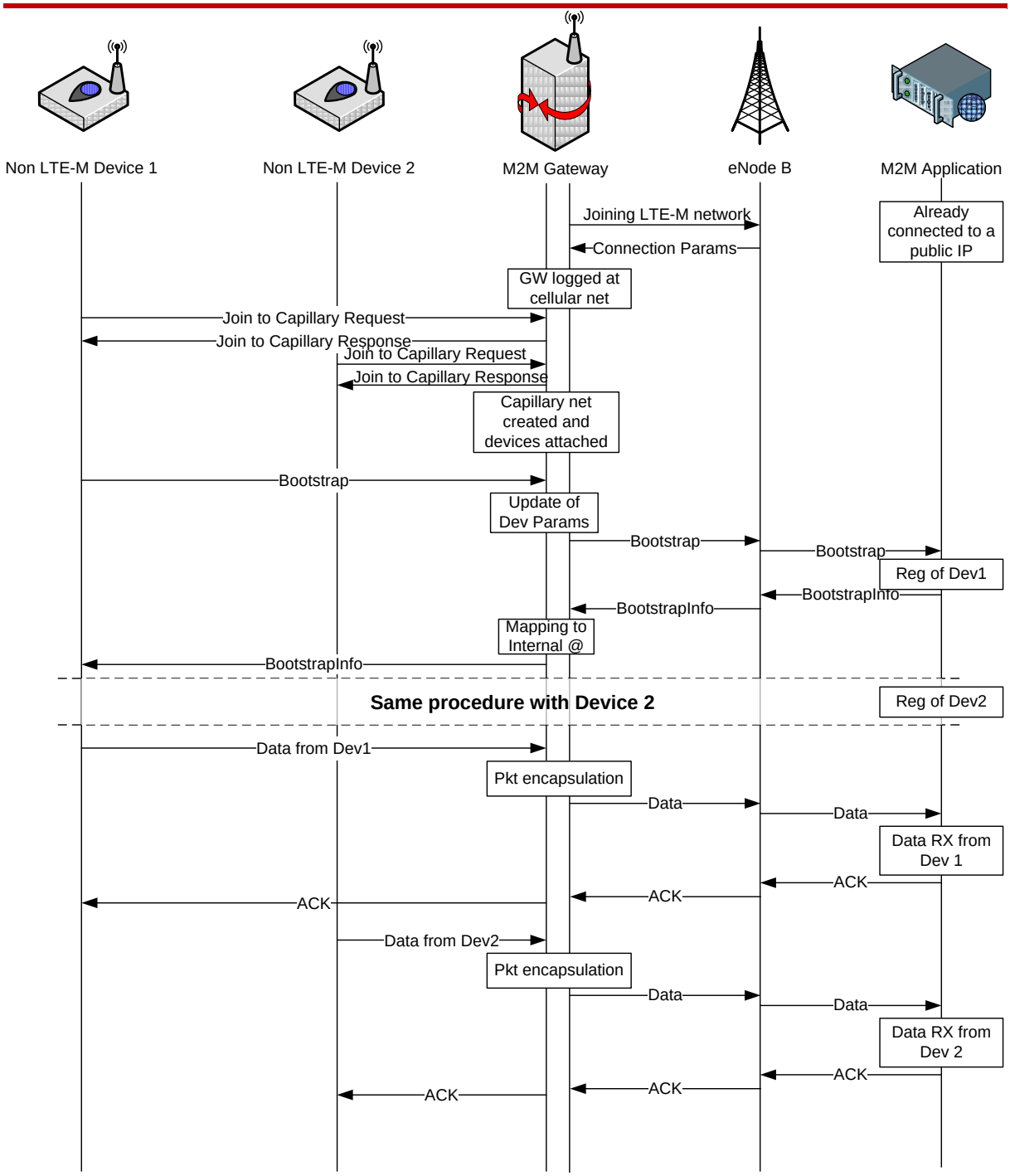


Figure A4-2: SMM message sequence

The execution of a SMM application demands three main steps prior to the exchange of information.

- **Enabling physical connections**. As presented in the figure the first phase is establishing the mandatory relationships between devices and gateway and between gateway and eNodeB. EXALTED is setting the foundation for an optimized version of LTE, LTE-M, but due to technical limitations it is impossible to exploit this technology in the demonstrators. At the same time, once the devices composing the capillary network have been deployed, they have to set the network itself, which means routing configuration and detection of the gateway. In this phase there is not any device management action as this is mandatory priori to enable DM functionalities.

- **Device configuration and registration**. Once the connectivity is possible using the wireless channels created. It is time for establishing the proper links between devices and M2M applications. This procedure is always initiated by devices in order to register themselves into the proper server thus enabling direct connections from devices to application and also the other way around. As presented in the figure, it is composed by a first step of bootstrapping informing gateway about devices destination address, and it forwards these packets properly. Then the registration is confirmed by servers and the bidirectional link open for communication.
- **Bidirectional data exchange**. Once the previous steps have been done, it is possible to perform data exchange between devices and servers. At this point, E2E M2M communication is available for its use.

The particular cases implemented in the different testbeds are compatible with this general view. Due to existing limitations some steps can be simplified or substituted for the ones that current technologies demand.

A4.2 ITS and eHealth use case

Subtestbed 2.1 is about combining vehicular networking and eHealth to record and transmit a patient's vital signs as a special telemedicine application that helps hospital resident health professionals to optimally prepare the patient's admittance. From the automotive perspective, this is a typical Vehicle-to-Infrastructure (V2I) communication scenario that can be very well supported over an LTE-M link under certain conditions. From the eHealth perspective, the path towards the Application Server should be as transparent as feasible.

As depicted in the overall picture of the integrated subtestbed, the proposed platform provides an IPv6 vehicular platform which integrates eHealth devices and allows sending captured health-related data to a Personal Health Record (PHR) application server in the IPv6 Internet. The collected data is viewed remotely by a doctor and supports a diagnostic decision.

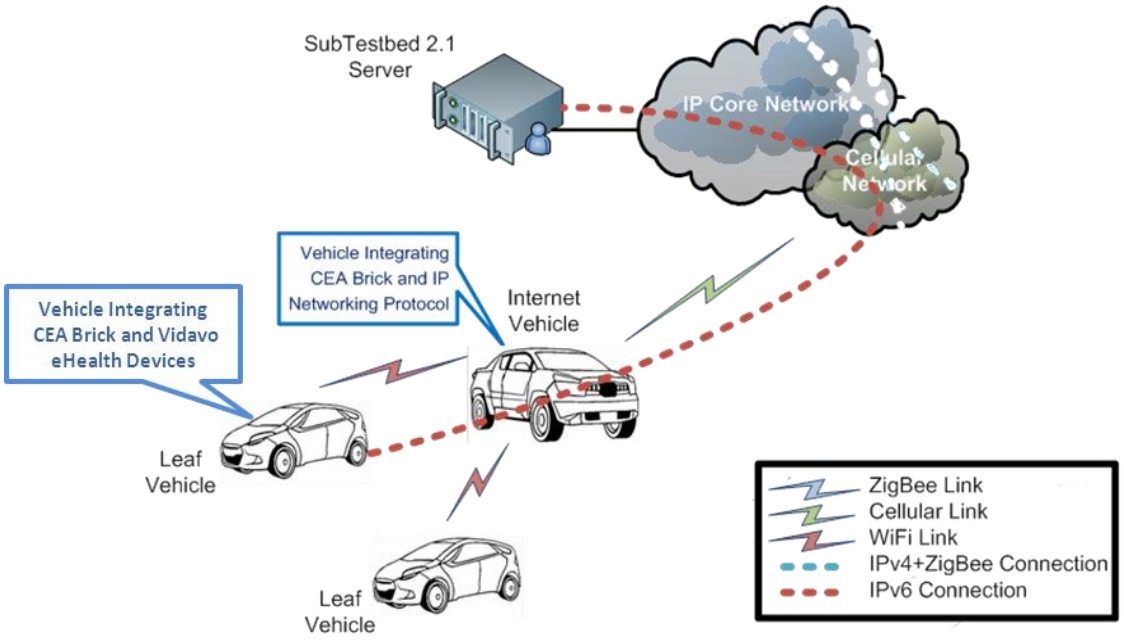


Figure A4-3: ITS and eHealth use case

The integration of vehicular and eHealth testbeds, described in the previous sections, requires the use of lightweight auto-configuration methods (we chose to enhance DHCPv6)

to provide IPv6 connectivity for resource constrained devices: the M2M Gateway and eHealth devices.

Basically, the above figure shows an IPv6-capable capillary network (eHealth) connected to an M2M Gateway (part of the ITS setup) inside the vehicle, and reaching the LTE-M infrastructure through it. Thus, communications from end-to-end involving the eHealth devices at one end, and the Application Server at the other are possible over an established IPv6 link.

In Figure A4-4 we show the message exchange diagram involving the M2M Devices, M2M GW, eNodeB and the Application Server in order to establish the connectivity and reliably deliver the packets.

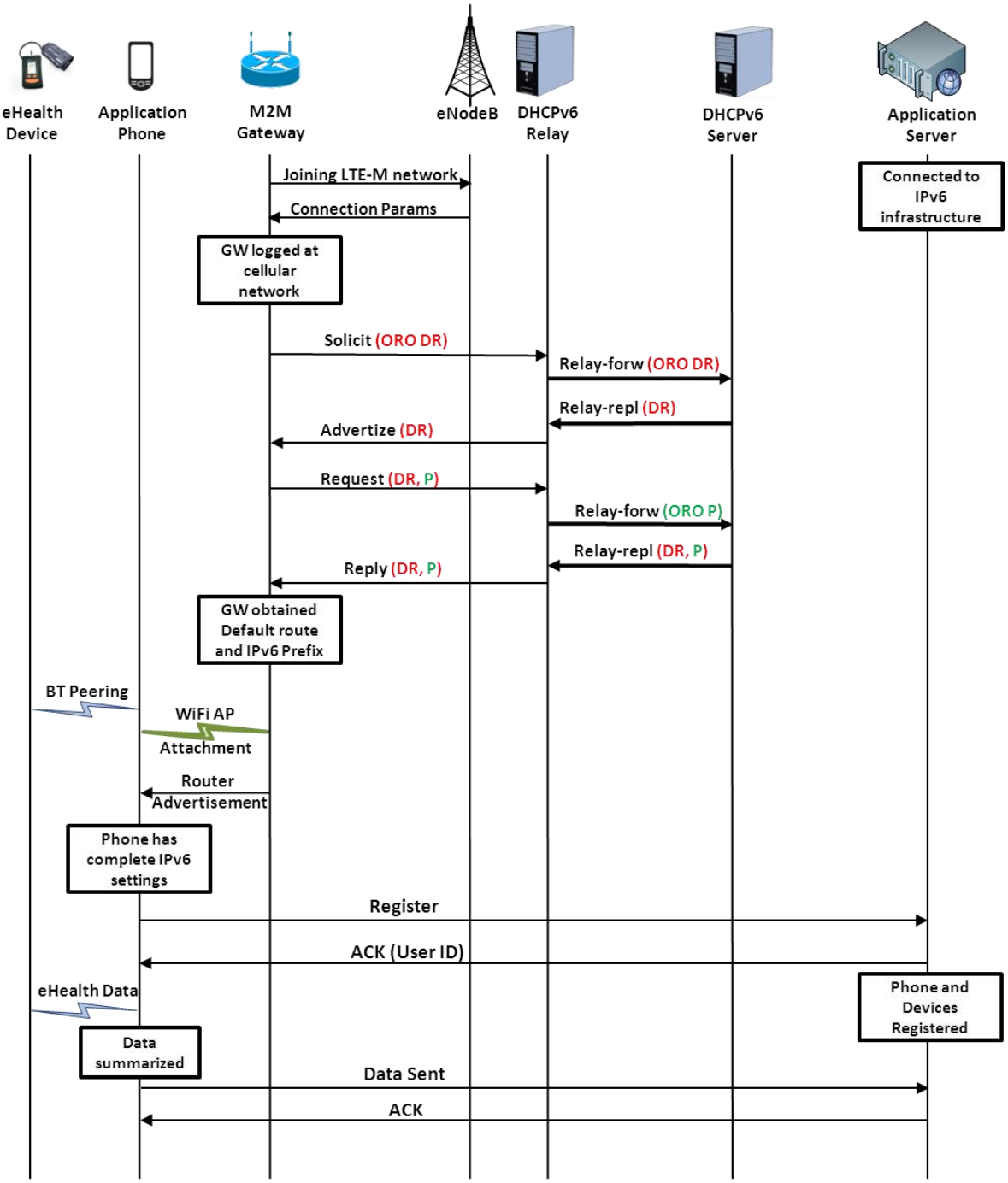


Figure A4-4: ITS and eHealth message sequence

The above message sequence can be separated into 5 main steps, each consisting on a certain number of protocol messages.

- **Enabling physical connections**. The first two messages of the above sequence aim at registering the gateway at the Infrastructure (eNodeB) and allocating the necessary resources. In order to accommodate with low-power and energy-restrained devices, EXALTED is setting the foundation for an optimized version of LTE, LTE-M. It is now impossible to exploit this technology in the demonstrators as it is currently specified. Once the gateway is registered to the network and the necessary resources allocated, the capillary network can be autonomous.
- **IPv6 parameters setting**. This phase involves the M2M Gateway, the DHCPv6 Server and the DHCPv6 Relays (if any). This procedure provides the M2M GW with a global prefix to redistribute in the capillary network, and a default route for reaching every node in the Infrastructure. This phase is initiated by the gateway, and involves protocol messages that detailed in section 3. Once the Gateway has these parameters, it can advertise them in the capillary network for the eHealth devices (more specifically the Application Phone).
- **Capillary network setting**. This phase involves the internal network of the vehicle only. Indeed, after the peering of the eHealth device with the phone and the attachment of the phone to the WiFi network (advertised by the M2M GW), the gateway starts advertising the prefix it obtained in phase 2. The phone is now able to auto-configure itself with the necessary parameters. The capillary network is now set up properly with the eHealth device attached to the phone via Bluetooth, the phone attached to the gateway via WiFi.
- **Server registration**. In this phase the phone registers itself and the peered eHealth device to the application server over the established IPv6 path and through the M2M Gateway. The server authenticates the message and acknowledged it with a user ID. The eHealth session is now established.
- **Data exchange**. The eHealth device captures health-related data and sends it to the phone. The phone summarize the data and allows for adding user comments before pushing it forward to the server over an encrypted IPv6 data path (TCP/SSL). The server acknowledges the received data and allows for remote monitoring of the physician.

As this integration networking-focused testbed is still under construction, we do not yet address security features relevant to the EXALTED. Privacy concerns for eHealth related topics and messages encryption are usually the primary security concerns in our area of work. A more in-depth threats analysis is necessary to cover all the relevant topics

List of Acronyms

Acronym	Meaning
API	Application Programming Interface
ARQ	Automatic Repeat Request
ARM	Advanced RISC Machine
BB	BaseBand
BER	Bit Error Rate
CapEx	Capital Expenditure
CDMA	Code Division Multiple Access
CoA	Care-of-Address
COPD	Chronic Obstructive Pulmonary Disease
CP	Cyclic Prefix
CQI	Channel Quality Indicator
Crypto-MAC	Cryptographic Message Authentication Code
CSV	Comma-Separated Value
DAD	Diagnostic Agent Descriptor
DHCP	Dynamic Host Configuration Protocol
DM	Device Management
DO	Data Object
DR	Default Route
DSP	Digital Signal Processors
DTD	Document Type Definition
DV	DeVice requirement
ECG	Electro-Cardio-Graph
EHR	Electronic Health Record
ELFOMA	EXALTED Lightweight for OMA
EXI	Efficient XML Interchange
FER	Frame Error Rate
FOTA	Firmware Over The Air
FPGA	Field Programmable Gate Arrays
FQDN	Fully Qualified Domain Name
FU	FUnctional requirement
FUMO	Firmware Update Management Object
FVC	Forced Vital Capacity
GFDM	Generalized Frequency Division Multiplexing
GW	Gateway
GwMO	Gateway Management Object
GZIP	GNU Zip
HAL	Hardware Abstraction Layer
HaLo	Hardware-in-the-Loop
HARQ	Hybrid Automatic Repeat Request
IoT	Internet of Things
IP	Internet Protocol
ITS	Intelligent Transportation System
IV	Internet Vehicle
JNI	Java Native Interface
JSON	JavaScript Object Notation
KeK	Key Encryption Key
KPI	Key Performance Indicator
LOS	Line-of-Sight
LTE	Long Term Evolution

LTE-M	LTE for Machine Type Communications
LV	Leaf Vehicle
M2M	Machine-to-Machine
MAC	Media Access Control
MIM	Machine Identity Module
MITM	Man In The Middle
MNO	Mobile Network Operator
MO	Management Object
MQTT	Message Queue Telemetry Transport
MR	Mobile Router
MSE	Mean Squared Error
ND	Network Domain
NDP	Neighbor Discovery Protocol
NEMO	NEtwork MObility
NF	Non-Functional requirement
NFC	Near-Field Communications
NLOS	Non Line-of-Sight
NT	NeTwork requirement
OBU	On-Board Unit
OFDM	Orthogonal Frequency Division Multiplex
OMA	Open Mobile Alliance
OOB	Out-Of-Band
ORO	Option Request Option
OTA	Over The Air
PAPR	Peak-to-Average Power Ratio
PCT	Parsimonious Covering Theory
PDN-GW	Packet Data Network Gateway
PER	Packet Error Rate
PHR	Personal Health Record
PKI	Public Key Infrastructure
PLR	Packet Loss Rate
RA	Router Advertisement
RESTful	Representational State Transfer, “full of”, “fully”.
RFID	Radio Frequency Identification
RPC	Remote Procedure Call
RS	Router Solicitation
RTT	Round Trip Time
Rx	Receiver
SAX	Symbolic Aggregate Approximation
SC-FDMA	Single Carrier Frequency Division Multiple Access
SCME	Spatial Channel Model Extended
SDM	Self Diagnostic Manager
SE	Secure Element
SMM	Smart Metering & Monitoring
SNR	Signal-to-Noise power Ratio
SoC	System on Chip
SPI	Serial Peripheral Interface
SSL	Secure Socket Layer
SV	SerVice requirement
TCP	Transmission Control Protocol
TTL	Time To Live
Tx	Transmitter
UART	Universal Asynchronous Receiver Transmitter
UI	User Interface

V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2V2I	Vehicle-to-Vehicle-to-Infrastructure
VULA	Vehicular Unicast Local Addresses
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
WS-*	Web-Services of many kinds
XML	Extensible Markup Language

References

[1] D. S. Baum, J. Salo, G. Del Galdo, M. Milojevic, P. Kyösti, and J. Hansen, “An interim channel model for beyond-3G systems,” in Proc. IEEE VTC’05, Stockholm, Sweden, May 2005.

[2] FP7 EXALTED consortium, “D3.3 – Final report on LTE-M algorithms and procedures”, project report, July 2012.

[3] FP7 EXALTED consortium, “D3.1 – First report on LTE-M algorithms and procedures”, project report, version 2.0, January 2012.

[4] James Roberts. The clean-slate approach to future Internet design: a survey of research initiatives. Annals of Telecommunications (Annales Des Télécommunications), Springer, 64(5-6):271–276, 2009.

[5] ETSI Intelligent Transport Systems (ITS), Online on 07 July 2012 at: http://etsi.org/WebSite/Technologies/IntelligentTransportSystems.aspxhttp://etsi.org/WebSite/Technologies/IntelligentTransportSystems.aspx.

[6] Y. Toor, P. Muhlethaler, and A. Laouiti. Vehicle Ad Hoc networks: applications and related technical issues. Communications Surveys & Tutorials, IEEE, 10(3):74–88, 2008.

[7] D.D. Stancil, F. Bai and L. Cheng. Vehicular Networking, Automotive applications and beyond, chapter 3, ”Communication systems for Car-2-X Networks”, pp 45–81, Wiley, 2010.

[8] Geographic addressing and routing for vehicular communications (GeoNet), FP7 ICT, Online on 08 Mai 2012 at: http://www.geonet-project.eu/?page id=9.

[9] K. Gosse et al. Vehicular Networking, Automotive applications and beyond, chapter 8, ”Standardization of vehicle-to-Infrastructure Communication”, pp 171–201, Wiley, 2010.

[10] Geoff Mulligan, “The 6LoWPAN architecture,” In Proceeding of EmNets ’07, Cork, Ireland, 2007.

[11] O. Troan and R. Droms, “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP),” version 6, IETF, 2003.

[12] L. Yeh, T.Tsou, M. Boucadair, J. Schoenwaelder and J. Hu. Prefix Pool Option for DHCPv6 Relay Agents on Provider Edge Routers. IETF (Internet Draft), draft-yeh-dhc-dhcpv6-prefix-pool-opt-05, 2011.

[13] R. Droms, P. Thubert, F. Dupont, W. Haddad and C. Request for Comments: 6276, DHCPv6 Prefix Delegation for Network Mobility (NEMO). IETF, 2011.

[14] D. Slamanig and C. Stingl. Privacy aspects of ehealth. Third International Conference on Availability, Reliability and Security (ARES),0:1226–1233, 2008.

[15] ZigBee Alliance, “Understanding ZigBee Gateway”, September 2010.

[16] FP7 EXALTED, WP4, “D4.2 –“– IP Networking System for M2M communications for EXALTED use cases” project deliverable, June 2012.

[17] Automated Control of Hospital Medicines and Supplies – Omnicell - http://www.avantec.uk.com/downloads/brochures/Avantec%20Brochure.pdf.

[18] L. Maruster and R.J. Jorna, “From data to knowledge: a method for modeling hospital logistic processes”.

[19] G. Privat, “From Smart Devices to Ambient Communications”, From RFID to Internet of Things Workshop, March 2006”.

[20] ETSI TS 102 689 V1.1.1 (2010-08) Technical Specification, Machine-to-Machine communications (M2M); M2M service requirements.

[21] F. Ganz, P. Barnaghi, F. Carrez and K. Moessner, “Context-Aware Management of Sensor Networks”, The Fifth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE11), 2011.

[22] Jessica Lin, Eamonn Keogh Li Wei and Stefano Lonardi, “Experiencing SAX: a Novel Symbolic Representation of Time Series,” DMKD Journal.

[23] EXALTED consortium, “D2.3 – The EXALTED system architecture (Final),” project report, August 2012.

[24] Card Guard Products & Technologies, Online on 15 Mai 2012 at: http://www.cardguard.com/cardguard.

[25] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu, “Gossip-based aggregation in large dynamic networks,” ACM Transactions on Computer Systems, 23(3):219–252, August 2005.

[26] A. Reggia and Y. Peng, “Modeling diagnostic reasoning: a summary of parsimonious covering theory,” Computer Methods and Programs in Biomedicine, vol. 25, no. 2, pp. 125 – 134, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169260787900484.

[27] 3GPP TS 03.48, Security Mechanisms for SIM application toolkit, R99, 06/14/2005NFC Forum."Technical Specifications". Retrieved December 2011.

[28] OMA Device Management v1.3. http://www.openmobilealliance.org/technical/release_program/dm_v1_2.aspx.

[29] OMA Device Management achieves 1.4 billion deployments. http://www.openmobilealliance.org/comms/documents/OMA_DM_1.4Billion_PR_Final.pdf.

[30] FP7 EXALTED WP7, “D7.1 – Business Models, Use cases & Technical Requirements Selection of scenarios for proof of concept testbeds and specifications for key building blocks functionalities and interfaces”, v2.0, project report, January 2012.

[31] FP7 EXALTED WP6, “D6.3 – Final specifications of the reliable device implementation”, project report, February 2012.

[32] Message Queue Telemetry Transport. http://mqtt.org/.

[33] SyncML Representation protocol v1.1. SyncML [RePro], http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/syncml/syn cml_represent_v11_20020215.pdf.

[34] FP7 EXALTED WP4, “D4.3 – Device management,” project report, to appear in October 2012.

[35] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976.

[36] EXALTED, consortium, “D2.1 - Business Models, Use cases & Technical Requirements”, project report, v2.0, January 2012.