COSMOS

Cultivate resilient smart Objects for Sustainable city application

Grant Agreement Nº 609043

D7.4.1 Smart heat and electricity management: Evaluation and recommendations

WP7 Use cases Adaptation, Integration and Experimentation

Version: 1.0
Due Date: 30 October 2014
Delivery Date: 13 March 2015
Nature: R
Dissemination Level: PU

Lead partner: Hildebrand Technology Limited
Authors: Joshua Cooper
Abie Cohen
Internal reviewers: Sergio Fernández Balaguer
Andres Recio Martin
Saima Iqbal

www.iot-cosmos.eu

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 609043

Version Control:

Version	Date	Author	Author's Organization	Changes
0.1	22 April 2014	Abie Cohen	HILD	Creation
0.2	11 March 2015	Panagiotis Bourelos, Achilleas Marinakis	ICCS/NTUA	Internal Review
0.3	12 March 2015	Abie Cohen	HILD	Review Changes & Updated Requirements, Recommendations and Appendix
1.0	13 March 2015	Joshua Cooper	HILD	Final review and updates

Annexes:

№	File Name	Title
1	Appendix A	Appendix A

Table of Contents

1 Introduction 4
2 Methodology 5
3 Criterion 6
4 Technology 8
5 Use Case Scenarios 9
5.1 London Borough of Camden Heat Network 9
5.1.1. Use Cases 9
5.1.2. Technologies 10
6 Requirements 16
7 Recommendations 17
7.1 Overall Recommendations 17
7.2 Unmet Requirements 19
8 Conclusion 21
9 Appendix 22
9.1 Requirements 22
9.2 Evaluation Percentages 24

1 Introduction

This deliverable focuses the evaluation of the adoption of the COSMOS technologies in real-world smart city cases. We will use the use case scenarios formulated in Year 1 as the basis for the assessment. It is important to evaluate the technologies thoroughly by looking at their consistency, correctness and completeness.

These measures can be expanded into a list of criteria with which we will evaluate each of the technologies in WP3-6. In using these measures, we will implement a benefit vs. cost assessment when applied to specific use cases in each of the two COSMOS scenarios.

Finally, it is important to use what we identify in this deliverable to make recommend what our next steps should be and where we should focus our efforts in terms of research activities and productivity.

In this Work Package deliverable we will

- provide the background material describing the current situation in each of the use case scenarios
- define a clear set of evaluation criteria for any given technology
- identify the technologies used in COSMOS
- assess each of the technologies in WP3-WP6 against the aforementioned criteria
- evaluate each of the technologies in WP3-WP6 alongside the overall requirements
- make recommendations based on the evaluations

The outcome of this deliverable is to evaluate the technologies developed in COSMOS in each of the use case scenarios, against a clear and complete set of criteria.

2 Methodology

In order to thoroughly evaluate the COSMOS technologies and assess the benefits they provide in different domains, we must first clearly define a complete set of evaluation criteria. This set of criteria must look at all aspects of a given technology and be able to apply to specific use cases.

We then collect and list all of the technologies used in COSMOS, which have been described in WP3-WP6. Consequently, we look at the London use case scenarios and evaluate how each of the technologies described will be used in them. This is done in a very structured mannered by using the criteria to test different aspects of the technologies.

Next, we consider the requirements in D2.2.1 and assess whether the use of the technologies developed in COSMOS will meet the necessary standards and solve the corresponding issues.

Finally, based on the results from the evaluation of the technologies in each of the Smart Heat and Electricity Management use case scenarios, we recommend next steps to take in COSMOS. Using the benefits and costs we found the technologies provide in different domains, we can help direct the research activities of the project.

3 Criterion

In order to properly assess the efficacy of the technologies used in COSMOS and their applications to the different use case scenarios, we need a set of criteria to test each of them against. This will ensure that each evaluation is fair and comparable. It is important that the criteria we choose to use capture the key measures of evaluating technology; namely consistency, correctness and completeness.

When looking at these measures, we realise that they can be broken down further to fundamentals and grouped into four main criteria blocks. We take consistency to mean how technically feasible, reliable and extendible the technology is. Correctness refers to whether it satisfies the problem at hand and how convincingly it does that. Finally, we take completeness to mean whether it is actually acceptable to implement such a technology and weigh its pros against the cons.

The following are a list of evaluation criteria which we will use a check list when assessing the efficacy of the COSMOS technologies:

1. **Functionality**
   a. *Satisfaction*.

      This concerns the extent to which the designed product satisfies the requirements.

      Does the technology solve the problem? Is it a direct or indirect solution? Does it completely solve the problem or only partly?

   b. *Ease of use*.

      This concerns the ease of use for the users. The users are e.g. operators and application engineers.

      Is it easy to design, implement and maintain? What programming languages are required, if any, and how well known are they? Are some libraries, if any, required and how accessible are they? Does it require specialised operators or application designers?

   c. *Reusability*.

      The extent to which the product can be used in other situations. Includes scalability and ability to use in (dis)similar contexts.

      How extendible is this technology? What sort of scale can it be rolled out to? Can it be applied to any other components of COSMOS? How generalizable is it or is it extremely specific/custom?

2. **Construction**
   a. *Structuring*.

      This concerns the partitioning of the product in logical or physical components.

What architecture is used? How complex is the system? How do different components in the technology communicate with each other and how efficient is that?

b. *Convincingness*.

This concerns the evidence that the construction will work and has the defined functionality (empirical proof/statistical argument).

How well known is this technology? What sort of research has to be done before design and implementation? Has this been used in another component of COSMOS? What is the likelihood of the problem being solved by using this technology as a solution?

3. Realizability

a. *Technical realizability*

This concerns certainty that it is technically possible to produce the product.

What technical requirements are there? How difficult would it be to implement this technology? Do the technical components that make the system's architecture link well together?

b. *Economical realizability*

This concerns the business case for the product.

Is the application of this technology financially feasible? Can the cost be covered by scalability and if so what sorts of volumes are we looking at? Do the benefits outweigh the costs? Is the technology worth the justification? Or is there a more cost effective solution that satisfies the problem?

4. **Impact**

a. *Risks*

Risks of the product during development stage or use.

Does the technology introduce new problems? Are there any privacy or security issues inherent to this technology? Are there authorization restrictions between components? Are there any risks that could end up affecting the end users through the applications?

4 Technology

A technology is the realisation of a function in the Internet of Thing's Architectural Reference Model. This includes physical devices, platforms, services and analytics; all of which are used to solve certain problems or add particular functionality to an IoT system.

COSMOS aims to build a smart system that uses Things in the space of IoT to solve problems experienced in cities nowadays. The two use case scenarios we focus on are Heating Networks in London and the Bus System in Madrid. In order to solve the issues that arise in the two scenarios we develop certain technologies in WP3-WP6. These technologies, when combined, produce the overall COSMOS system ranging from the hardware to the software and from the servers to the sensors. Each of the technologies mentioned in this section fulfil a specific role and have a purpose in COSMOS.

This section provides us with a clear list of technologies used in COSMOS, described fully in WP3-WP6:

(WP3) D3.2.1: *End-to-end Security and Privacy*

- Hardware Security Board running Linux-based system

(WP4) D4.2.1: *Information and Data Lifecycle Management*

- Complex Event Processor (CEP) Management service via REST API
- Message brokering & storing using Rabbit MQ & OpenStack Swift, respectively
- Cloud-based Object Storage search & pre-processing: Softlayer (new search API) & storlets

(WP5) D5.1.1: *Decentralized & Autonomous Things Management*

- MAPE-K model including Social Monitoring & Analysis

(WP6) D6.1.1: *Reliable & Smart Network of Things*

- Machine Learning based methods to build Predictive Models for Interpolation/Extrapolation e.g. Kalman Filter & Artificial Neural Networks
- Enhancing CEP-based situational assessment processes with adaptive feedback loop at runtime to constantly evaluate and analyse situations (based on the relation between different events)
- Experience sharing through storage (semantic store API) and search (querying language: SPARQL)

5 Use Case Scenarios

5.1 London Borough of Camden Heat Network

5.1.1. Use Cases

5.1.1.1. Capital Planning

Use case : Capital Planning
ID: 1
Brief Description: The EnergyHive system in each building enables Capital Planning officers to perform a more rigorous cost/benefit analysis of suggested programs or technology installations. The system provides accurate information as to the carbon/monetary saving of an implementation.
Primary Actors: Capital Planning Officer
Secondary actors: Mechanical & Electrical Engineer, Sustainability Officer
Preconditions: EnergyHive system must be installed throughout each building in the estate
Main Flow: 1) Sustainability Officer identify an opportunity for environmental improvement of system 2) Engineer select appropriate technology for instalment 3) EnergyHive system provides detailed information as to the effect of the change in the system 4) Capital Planning officer uses EnergyHive information to assist in cost/benefit analysis
Postconditions: The Capital Planning officer decides whether to rollout the proposal

5.1.1.2. Minimising Carbon

Use case : Minimising Carbon
ID: 2
Brief Description: An effective way to minimise carbon is to give more weighting to processes with lower carbon production levels, whilst maintaining the demand. The interconnected IoT-based system using an energy platform will make possible effective management of the energy supply in order to minimise carbon production. With minimal input by the resident or site staff, the system will predict the estate's heat and electricity consumption in half hourly intervals and manage the CHP and boiler accordingly.
Primary Actors: Resident
Preconditions: Specialised Instalments 1) Gas Flow meter to CHP from boiler to regulate the Gas supply 2) Control system with temperature sensor on boiler 3) Flow meter/temperature sensor on Solar Thermal 4) Heat meter in each dwelling 5) Communication infrastructure between sensors and hub
Main Flow: 1) System predicts the estate's heat and electricity demand for a half hour period 2) System calculates required gas supply and distributes to CHP and boiler accordingly

3) Carbon produced is measured 4) Individual resident heat consumption is monitored
Postconditions: 1) The resident is charged for their personal heat consumption 2) Prediction errors are logged to improve system on later iterations

5.1.1.3. Minimising Demand

Use case : Minimising Demand
ID: 3
Brief Description: Another method of reducing carbon production is to minimise the demand for Heat energy production. This is possible through the current IoT platform, namely EnergyHive (designed by Hildebrand). The EnergyHive system will use smart meters to report real-time energy consumption information automatically and remotely. The system assists the user in setting a heating schedule with accordance to their budget.
Primary Actors: Resident
Preconditions: 1) EnergyHive system implemented in each dwelling 2) Valve up/ down control system to the radiator
Main Flow: 1) Resident accesses their customer account to view balance 2) Resident can set a heating schedule 3) Resident is given tariff and projected balance for a given schedule
Postconditions: 1) User can optimise their schedule to minimise their consumption

5.1.2. Technologies

5.1.2.1. End-to-end Security and Privacy

The technology used in this deliverable is a Hardware Security Board, which has been loaded with the Linux operating system. It clearly satisfies the aim of end-to-end security as it provides the necessary link between the physical world and the actual COSMOS platform. Application developers are unlikely to struggle with this technology as the software components are written in C/C++, which is a very well-known language. The main point of discussion is whether we can make this technology scalable, in the sense that we can generalise it enough to get it manufactured in a large volume of goods.

As the security board is implemented in hardware we benefit from high speeds and reliability, which we wouldn't get in a software-based solution. Also, the solution is simple enough and depends on very well-known programming languages and easily attainable libraries. Therefore we can classify this technology as one that is very likely to not have any issues in implementation or maintenance.

On a technical level, the requirements to program the hardware are not difficult to attain whatsoever giving the technology a lot of credibility. Furthermore, it does ensure end-to-end encryption and an extremely secure route for data to pass through. However, the monetary cost of this technology in this use case can be argued as unnecessary. It seems difficult to justify the cost of developing and manufacturing the security boards to then add them to energy sensors, which don't carry particularly sensitive data. Moreover, we must remember to consider how many sensors we can develop this technology for and if the number is enough to cover the development costs. On the other hand, as COSMOS deals primarily with the IoT space, in these use cases we consider how dangerous it is for a hacker or malicious piece of software to be able to control devices in a home. For example, without end-to-end encryption it would be less difficult for a hacker to, say, stop a building from drawing power from the grid.

Finally, we assess the level of risk this technology presents during its use in the heat network. It is clear that as this technology aims to provide full data security throughout the entire system; then not only does it not introduce new risks but in fact it reduces the chances of data being stolen or the system being infiltrated.

5.1.2.2. Information and Data Lifecycle Management

We will not assess the efficacy of the Complex Event Processor (CEP) as a technology in Camden's Heating Network here as it is more useful to evaluate it in WP6. The CEP relies on the Smart Network of Things in order to be autonomous, dynamic and share experiences between VEs. Hence, assessing its functionality, construction and impact will be more useful when done in conjunction with Machine Learning and Situational Awareness.

The first technology we assess in WP4 is therefore the use of a message broker and storage system in the Heating Network. These two tools easily satisfy the needs of the system, which are communication between devices and a layer that stores all of the data we are measuring. We choose to use Rabbit MQ to broker messages and OpenStack Swift to store the data as they are easy to implement and use and have numerous adapters and tools for specialized tasks such as integration with other existing platforms.

These technologies have been well researched and prove to be easy to implement and extremely reliable. Hence, we have no concerns in terms of whether the theory is in fact practical and how to implement them into the current COSMOS structure. Finally, we note that there don't seem to be any inherent problems with the brokering and storage systems and therefore it is financially viable.

We now assess the use of an Object Store as a place to store the data we receive from all of the multi-sensors, hubs and actuators via the message brokers. This involves the use of Storlets to help manipulate and analyse that data in an efficient way by running inside the object store system. One of the difficulties with using Object Stores over, say databases, is that an operator or application developer will have to predefine a data chunk size (for example 5Mb) when searching through the store. This is because the data is saved as a large file and has no columns or headers with which once could sequentially run through entries. This therefore leads to scalability issues, as the problem gets worse as the system expands in size. This is easily seen in the context of energy data building up over the course of days and weeks. Storing this data as a time series in a database is arguably a better method for data that follows this structure, instead of cutting up the file into arbitrary pieces when running searches and analyses.

The objects are stored in OpenStack Swift cloud storage and we augment this with Storlets. In terms of construction, structuring the object store is not an issue as all the data is saved in a large file. This leads back to the issue mentioned before of how to construct the querying API that we use to search through the objects. From a financial point of view, this method of storage is good value due to the way the data is entered and inexpensive object store services available from Amazon S3 and IBM Softlayer. However, we will need to find a way of dealing with reading large objects into memory for analytics, otherwise the system's performance will suffer. This is a risk we really need to think about as in this particular use case we want to run analytics on weeks' (or even months') worth of data in order to identify demand trends and patterns. The results from these computationally intensive processes will inevitably help with the Machine Learning and Forecasting described in WP6 (*D6.1.1*), which are primarily used to balance out supply with demand and reduce carbon.

We now assess the use of storlets to run computation on the data directly in the object store. storlets are tailored for store object processing so it will work well for pre-processing data as a filter and retrieve it from the object store only transmitting the aggregated or filtered values. However, as mentioned previously, storlets could be inefficient when running computations on large objects, as they require reading all the content into memory or indexing the raw object into an efficient data structure. Databases are more commonly used over object stores; however, storlets are written in Java which is a well-known language for application engineers. Therefore, it should be relatively easy to develop these processes and implement them into the COSMOS system. A benefit of storlets is that only authorised users can get access to certain metadata during searches. This keeps all of the energy data secure and private and only certain buildings/flats have access to each other's energy readings, for example. Furthermore, storlets allow us to quality check the mass amounts of data we see as we see it which improves the overall quality. In terms of scalability, storlets are sandboxed which means that they are only given access to certain storage objects. This is especially important in the future if we want to allow arbitrary users to write storlet code for the COSMOS platform.

Construction of these storlets seems fairly straightforward as long as they are executed efficiently. The benefit of them is that they work directly on the data in the store and hence the processes execute locally, but there are certain implications with that. In reality, it is important for us to test and compare efficiencies of different storlets with different data sets. In the Camden scenario, analyses and computations on different data sets vary in efficiency and therefore we must decide whether object stores and storlets are suitable for each of the tasks.

Finally, we note that the Object Store storelets are computationally more expensive than accessing the storage object alone. Therefore more CPU may be needed, for example processors and RAM that would normally not be required for a storage node. These, of course, have an inherent cost that we weigh up against the reduction in network traffic benefits that storlets provide. This can be justified as long as we can efficiently run these processes; specifically by running them as low priority and data streaming processes in a asynchronous modality.

5.1.2.3. Decentralized & Autonomous Things Management

The technology in WP5 is that of Case-Based Reasoning used to generate actuation plans using the underlying state-space in the COSMOS system as input. This provides COSMOS with a lightweight way of creating and sharing decision knowledge that can be further assisted by technologies used in WP6 such as Machine Learning analytics and Experience Sharing with model based reasoning between Virtual Entities. The Experience Sharing architecture in particular relies on a system where buildings/flats can communicate with each other freely and quickly in order to pass information and knowledge between one another to aid with the decision making process. In the Camden scenario, the case bases will have to be stored in a level on or above the VEs in the COSMOS structure. This would make it is easy for them to communicate with each other and also for sensors and actuators within each VE to communicate.

From a resource efficiency standpoint, Case-Based Reasoning (CBR) is a much easier system to implement than, say, Model-Based Reasoning (MBR). The reason for this is that MBR usually requires more history than the current state space and requires computation in excess of vector distances. Also with MBR any addition of dimensions to the system, i.e. attributes requires recalculation of model parameters. On the other hand, CBR has its own drawbacks such as how dynamic it can be as it is very limited to only what is in its shared case base. For this reason, it is difficult to assess how useful this approach will be when making logic-based decisions in a network of heating systems. However, we can say that this technology is suited to numerical problems and the main obstacle to overcome is the short-term delay in the learning process.

The social aspect of this work package also creates a few technical difficulties. The non-uniformity of the devices in the VEs such as multi-sensors and actuators means that writing the software locally will require a lot of work. The multi-sensors and displays in the Camden project all have software embedded directly on them, so they benefit from not having to wait for a boot up process as well as being extremely efficient. The downside however, is that a CBR approach with a cloud based storage is the only reliable and justifiable way of creating a case base at the moment. Hence, making the decision making process very local is feasible, but training is not yet possible. We can; however, make data available at REST endpoints and therefore VEs can request case bases by simply using HTTP requests.

An attempt to put CBR on small powered peer devices has been made and in Year 2 there will be further evaluation on the feasibility of low power, autonomous devices for CBR.

Finally, the cost of having to design and program each different type of device in a range of virtual entities is likely to make us question the decision of making things local. Furthermore there will be a high cost incurred (in the form of time and effort) when researching methods of Experience Sharing when decisions are made locally, for example between devices such as light bulbs and multi-sensors in a flat.

5.1.2.4. Reliable & Smart Network of Things

The Machine Learning based technology used in this deliverable aims to solve the problem of missing data and future prediction. Building a statistical model on the vast amounts of data that our virtual entities collect allows us to predict expected values that are missing due to some sort of error in the system. This could be, for example, from a faulty sensor which isn't sending data at the right time or because the Internet connection has dropped in a particular flat in a block. In the latter case, data accrued from similar/close-by flats could prove useful in predicting the missing value(s). The Maximum Likelihood based methods described in WP6 *(D6.1.1)* have been extensively used in research for many years and have proved to be easy to use, stable and reliable solutions. Furthermore, the use of Kalman Filters has been discussed and we expect to see good results from this slightly more complex technique. The reason for this is because not only does it achieve the 'on-the-fly' requirement but it is also a self-correcting system implying that our predictions should theoretically get more accurate as they learn from their mistakes. Another technique called Artificial Neural Networks (ANN) was discussed in this deliverable however it has fairly been dismissed for now as it poses a few new problems without adding many benefits. Finally, the use of Kalman Filters is a good direction to work in as it is fits a very broad class of problems, namely the issue of predicting missing values in real time like we see in Camden's Heat Network.

The building of the model in this technology is relatively straight forward as the data is structured as a time series, which the Kalman Filter works well with. The data prediction and imputation stage can easily be placed between the data input and analytics layers and hence not causing an issue between different components in the COSMOS system. Furthermore, the Kalman Filter is simply an adaptive Maximum Likelihood estimator so we do not run into the issues that new and experimental techniques have.

Technically speaking, the model will need to run algorithms on large amounts of data which implies that we need to consider things like efficiency and the required computing power behind the system. However, these are all considerations rather than issues or possible stumbling points so it is fair to say that the use of this technology is definitely technically possible. Imputation of missing data points and the ability to extrapolate for prediction purposes not only makes our dataset more complete and reliable but also provides us with useful information for the Complex Event Processor. This makes the technology financially viable, especially because it is an inexpensive system to develop and relatively easy to maintain.

From our initial assessment, there don’t seem to be any obvious risks from the use of this technology. These techniques are known to be very reliable and hence we don’t expect to run into the issue of poor prediction performance, resulting in poor decisions made by the CEP.

The next technology we look to assess is that of situational knowledge acquisition and analysis in the Heat Network use case. The ability to forecast certain events and allow an autonomous system to make a decision based on strict rules would improve the COSMOS system as a whole. An important point to note is that the manual setting of rules and patterns limits how useful the integration of the CEP is in the system. Implementing a method for automatically generating rules for the CEP would allow the system to perform well in dynamic scenarios, therefore making it highly scalable and reusable in different decision making sections of COSMOS. The cost of this adaptive technology, of course, is the difficulty incurred when designing the Machine Learning system, as it must be extremely versatile. One can argue, however, that the benefits outweigh the costs as we eliminate the human errors when manually setting rules for the CEP and that makes the system easier to maintain as a whole.

As mentioned in the previous paragraph, the construction of this technology may prove difficult. An example of how Machine Learning can be applied to a generic CEP can be found in the WP6 documentation (*D6.1.1*) and this is our main motivation in trusting the theory to be applicable.

The application of this technology on the heating network scenario has huge potential, as it is highly scalable. The ability to apply a dynamic rule set to any given real-time decision making process makes the system not only autonomous but extremely easy for an application engineer to develop for. This, in turn, makes the vast amount of work going into the development of the adaptive CEP financially worthwhile.

As with any autonomous system, quality control and bug testing is very important during the development stages. This will lower the chance of any risks when the system is running on its own. In this particular scenario, a badly designed system would make poor use of the CEP and consequently could waste energy and money.

Finally we assess the use of Experience Sharing as a technology to help the Virtual Entities (in this case we can take a building or flat, for example) act in a more autonomous way when detecting events and solving problems through decision-making. As the Internet connection is likely to be extremely stable in this scenario, the sharing of solutions between the VEs' case bases should be extremely reliable. We must note however, that the VEs communicate using SPARQL queries and REST HTTP requests so a friendly interface must be designed and implement to aid the VE developers. The technology is extremely reusable as it is so generalized as a concept. The idea of storing solutions to reoccurring problems in a central and easily accessible storage space can benefit any system within COSMOS.

The construction of the storage system is expected to be simple, but the design of the API for searching is likely to be slightly trickier. The concept of having a case base to search through using key words works well in theory, but needs to be constantly tested and improved to ensure that it works on a practical level. We believe that this will make the system more autonomous for solving quick and simple problems and hence will make COSMOS more efficient as a whole. However, when it comes down to dealing with more complex issues, a lot of care has to be taken in the design of the query section to be able to deal with more unique scenarios.

In this particular use case, the efficacy of the system will improve greatly through Experience Sharing. When all of the VEs have access to a platform on which they can communicate and help each other make decisions, the COSMOS platform will be able to map the supply of energy much closer to the true demand levels. This in turn will help prevent unnecessary use of energy and hence reduce carbon. Furthermore, this platform aids in the development of applications aiming to reduce the levels of demand. An example of this would be if a VE's state switches from occupied to unoccupied, the CEP could automatically ensure all unnecessary devices are not drawing power and hence demand is lowered.

6 Requirements

In this section, we evaluate the list of Requirements that have been put together over the course of COSMOS. We aim to evaluate our progress for each of the requirements on the three main criteria: consistency, correctness and completeness. We aim to achieve all three criteria for each of the requirements as this shows that we have fully satisfied the needs of COSMOS.

Approximately half of the requirements have been met fully due to the design and implementation of the technologies in the use case scenarios. As the use cases are so diverse and test the system so thoroughly, we find that satisfying the needs of the requirements are consistent not only across both the London and Madrid systems, but also within these systems. The consistency of these technologies for all aspects of all use cases in each of the scenarios has been noted in Section 5 and this is verified in our evaluation of the requirements.

We now look at the requirements that fall into the category of consistent and correct but not yet complete. About 20% of the requirements have been marked with the 'Mostly Met' label as the aforementioned technologies do solve the issues that the requirements propose and do it in a smart, efficient and scalable way. Furthermore, these technological solutions can and have been adapted to fit different aspects of COSMOS and work well with all components in the system. The final criterion of completeness; however, has not been met because there are still parts of the requirement that have yet to be fulfilled.

The final evaluation bucket we look at is the 'Partially Met' one, which only has satisfied correctness, without having achieved consistency or completeness. Just fewer than 20% of the requirements fall under this category, which is pleasing at this stage. This label is given to requirements that have the potential of being met due to the use of the aforementioned technologies; however, it is just the theory behind these technologies that lead us to believe that these problems can be solved. But, in terms of ensuring that the entire requirement can be met across the board without any loopholes or errors, the requirements that fall into this bucket fall short.

7 Recommendations

7.1 Overall Recommendations

In this section we will highlight the courses of action we wish to take in the upcoming year, based on our findings in Chapter 5. We aim to objectively suggest areas of COSMOS to focus on further, whilst recommending particular topics to research, concepts to develop further and techniques to continue improving upon. Finally, we look into the requirements that have not been met yet and discuss ways of making them correct, consistent and eventually complete.

Device Security

The first course of action we recommend is for the COSMOS team to research and test whether we can generalise the hardware security board to fit any *Thing* in the IoT space. This implies that we should aim to create a uniform hardware-based solution for all VE sensors. Having to invest time and effort in developing the same technology for slightly different platforms is wasteful and in order to increase efficiency we should aim to create one security board that can work with all sensing entities. It is also important to look into possible holes in the system that hackers could infiltrate and therefore find out what components could be compromised. WP3 aims to ensure end-to-end security & privacy, which implies that it is imperative that we develop methods of making the system impenetrable.

Object Store and Storelets

Next, we need to find the most efficient way of running the storlets on large batch data. As storlets were not designed to run procedures on chunks of data, we must look at varying the truncation size of the data to maximise efficiency and keep the data clean and organised. Moreover, we should look for a way to adapt the analyses to run as steaming processes to benefit from the use of storlets. Being able to achieve this would eliminate the need to look into optimum truncation sizes and techniques.

Also there seems to be scope to create an object store binary format that would lend itself to aggregations in the time and space domains. This format could either be pre-computed or indexed for fast, lightweight computation. This would aid the portability of the data between systems as well.

Case Based Reasoning and Experience Sharing

Another important recommendation is to find the best way of allowing the VEs to communicate their experiences, not just their raw data or state space. We must find the balance between the speed of having logic done locally and the efficiency of having logic done in the highest level of COSMOS. This is particularly crucial for the implementation of Case Base Reasoning and Experience Sharing. It is also recommended that we are constantly looking to extend the case base so that it can deal with a multitude of different scenarios. The usefulness of this technology heavily depends on the size and diversity of the case base and therefore we must aim to constantly be extending and refining it.

There should be an effort made to understand the archetypical cases that may apply for a wide range of applications. For instance VEs that have mobility, VEs that describe environmental conditions and how they may link to generalised actuation plans i.e. change heating, lighting or humidification.

Machine Learning
In terms of analytics, we should compare different Machine Learning techniques for classification and regression for archetypical use case scenarios such that general reuse is possible. Researching many possible ways of modelling our system so that end users can interact with these complex technologies is of paramount importance in COSMOS, as we need these models to make sense to human observers and application developers. There is also great benefit in getting the system to adapt dynamically and improve over time in an unsupervised way.

We should aim to run quality control and bug testing thoroughly on the CEP-based technologies, as this technology may have limitations in large deployments, especially is rule sets are authored by multiple parties.

Furthermore, we should follow a trial and improvement approach when developing the Experience Sharing API, to understand if the best experiences are winning and there are not conflicts in experience ratings that cause poor results.

Practical System Issues
Finally, it is recommended that we research how to make the communication in the Heating Network as reliable and efficient as possible. Issues such as a volatile Internet connection can cause issues such as missing data values and infrequent data transfers. This issue seems to have been either accepted or overlooked, but it is extremely important that we find ways of ensuring the data is regular and complete, as the entire COSMOS platform relies on it or ways of working around data quality become directly addressed by COSMOS.

7.2 Unmet Requirements

The requirements listed below have been marked as *Unmet* as there is no clear documentation on how and where these have been satisfied in the COSMOS project. In this section, we will go through each of the requirements and suggest ways of moving them forward with the aim of meeting them fully.

Four of the unmet requirements are in WP3, which deals with ensuring security and privacy in all components of COSMOS. Neither a secure boot process nor a secure update mechanism has been mentioned in the documentation and we can therefore assume that no progress has been made and neither of these features has been implemented yet. These requirements are extremely important as they prevent the system from being penetrated and allow COSMOS to update itself seamlessly with minimal effort. The same reasoning applied to the requirement that there should exist a secure execution environment, where the core apps run. As this has not been discussed, the only reasonable recommendation would be to consider ways of achieving it and aiming to test and implement a simple yet effective solution. Finally, we have stated that we want Virtual Entities to be able to directly use hardware security functions and this has not been described in WP3. It is important that we start by discussing how feasible this requirement is and propose ways of achieving it without compromising the integrity of the system.

WP5 has a few unmet requirements that we will now look into further. The concept of experience sharing has been discussed in depth regarding its usefulness and the benefits it could provide COSMOS. However, no real implementation of a taxonomy or similar framework

UNI ID	Description
3.5	secure boot in order to have the device, every time, in a safe and known state
3.6	secure update mechanism (e.g. update each device on its own)
3.10	secure execution environment (e.g. split the execution environment into secure - where the core apps are running, and unsecure - where the non-vital apps, which require more processing time and are not system critical, are running)
3.11	allow high level applications to use core hardware security features (e.g. remote configuration authentication performed using the secure element -> the software just triggers the element and the security part is handled in hardware)
4.9	Publishing sub-system offer data broadcasting based on semantic analysis results
4.11	System should provide the capability to define processing configurations/topologies, including fail safe configurations
5.2	An XP taxonomy (or taxonomies based on other properties, characteristics or descriptions of the objects) could be developed and allow semantic look-up.
5.4	Human Users (individuals and groups/companies/public services) should have their own representation in COSMOS (e.g. through the use of VEs of Human Users).
5.8	COSMOS could get as input the classification of the App-Requests depending on the use cases (e.g. "waste management", "traffic control").
6.16	It could be possible for an object to issue a Call for Tender, in order to advertise its specific needs and get experience-sharing proposals from other objects.

has been proposed and therefore cannot be market as a met requirement. Similarly, there has been no mention of creating a representation of Human Users in COSMOS so that we can provide access to certain users for certain components. Finally, we have discussed in great length the IoT reference architecture and how VEs are structured and fit into the domain model; however, no work has gone into App-Requests.

Finally, there is no documentation on a Call for Tender feature whereby objects in the COSMOS space can broadcast their needs and XP. This would benefit the communication side of the system and improve the experience sharing features. Our recommendation is that we start looking at ways of advertising these needs and characteristics and attempt to implement them in a use case scenario to test its efficacy.

8 Conclusion

Within Year 1 it is difficult to assess technology that is new and immature. Most of the evaluation has been done on either design documentation or prototypical systems.

Our evaluation is optimistic for Year 2 in key areas where innovation is occurring namely, CBR, storelets and hardware security. Specifically we see

- CBR making a big impact in the way low resource devices can become intelligent, that CBR case bases can be exchanged for experience sharing and generalisation for CBR to be widely applied
- Storelets redefining the mix of computation and store on cloud nodes, binary file formats that can change the way sensor and VE data is exchanged and secured and commercial potential in treating storelets as units of intellectual property and can be rented or sold as a utility service
- Hardware security is helping the resource limitation and bandwidth constraints with in the IoT space by focusing on elliptical curve cyphers over block cyphers that add significant data overhead and processing in encryption and transmission of small packets of data

There is clearly more work to be done in Year 2 to integrate COSMOS services and align them to the IoTA reference architecture. City services are clearly wanting to adopt IoT and having working systems that can realise business processes will have a large impact.

9 Appendix

9.1 Requirements

UNI ID	Description	Rationale	Evaluation
3.1	communication shall take place over standard interfaces (e.g. I2C or SPI for Sensors and Ethernet between devices)	Using standard communication interfaces minimizes the development overhead and maximizes the code reuse	Met Fully - Consistent, Correct & Complete
3.2	data must be checked for functional corectness (e.g. identify defect and/or disconnected sensors and/or devices)	Transmitted data has to be valid in order to conserve bandwidth and assure the integorty of the entire system	Met Fully - Consistent, Correct & Complete
3.3	data must be "secured" in oder to allow a high enough security level: - eavesdropping -> encryption - data modification -> Encryption + integrity checks - replay attacks -> integrity checks - identity theft -> encryption + authentication - non-repudiation -> digital signature + encryption + authentication	only secured data can be trusted - plain text information can be modified while "traveling" over the Internet	Met Fully - Consistent, Correct & Complete
3.4	secure storage for the on-device secret informations (e.g. encryption keys)	a secure storage element is needed in order to provide a root of trust for the hardware secure boards	Met Fully - Consistent, Correct & Complete
3.5	secure boot in order to have the device, every time, in a safe and known state	the system needs to execute only trusted software and run into a known state - for this very reason a secure boot mechanism is essential	Not Met
3.6	secure update mechanism (e.g. update each device on its own)	secure update provides the means to upgrading the system	Not Met
3.7	secure enrolemment mechanism (e.g. enroll each device in the system; if one device fails it will be automatically disabled)	each device needs to be uniquely identifiable and addresable	Met Fully - Consistent, Correct & Complete
3.8	remote configuration	all VE should be remotely configurable	Met Fully - Consistent, Correct & Complete
3.9	hardware root of trust (e.g. let the software rely on a secure element rather than make it secure on its own)	each VE with a hardware security board should be able to use the hardware security features as a root of trust - software should only handle the high level security operations	Mostly Met - Not Complete
3.10	secure execution environment (e.g. split the execution environment into secure - where the core apps are running, and unsecure - where the non-vital apps, which require more processing time and are not system critical, are running)	secure VEs should have a clear separation between security & privacy critical apps and "the rest"	Not Met
3.11	allow high level applications to use core hardware securit features (e.g. remote configuration authetication performed using the secure element -> the software just triggers the element and the security part is handeled in hardware)	secure VEs should be able to use directly hardware security functions whereas software only handles small parts of the communication & configuration -> HW root of trust	Not Met
3.12	use a standard OS which is verified and trusted (e.g. Linux)	standard OSes provide the necessary ifrastructure, are verified and can be used "free of chanrge" (e.g. Linux)	Met Fully - Consistent, Correct & Complete
3.13	use a secure server backend for key and data storage as well as for device enrollement	backend intrastructure is needed (e.g. Keystone)	Met Fully - Consistent, Correct & Complete
3.14	make the security "stuff" mostly transparent to the end user	security should "just be there" - users should not care about the infrastructure but rather use it	Met Fully - Consistent, Correct & Complete
3.15	a unified API should spread over all VEs	all VEs should have the same API and signal via a flag which security level is provided	Met Fully - Consistent, Correct & Complete
3.16	There should be a mechanism which enforces authentication and access control to the cloud storage.	This is necessary to protect the large amounts of data that will persist in cloud storage.	Met Fully - Consistent, Correct & Complete
3.17	There should be a mechanism which ensures that metadata search results only contain data that the relevant user has read access privileges for.	This is necessary to prevent leakage of information via metadata search to unauthorized users.	Met Fully - Consistent, Correct & Complete
4.1	There must be a mechanism to collect raw data and make it persistent.	Data will be continually streamed into the system and it should be stored for further analysis. The raw data could be data produced by VE's such as temperature readings or data tracking the location of a bus.	Met Fully - Consistent, Correct & Complete
4.2	There should be a mechanism to map raw data to a format that is suitable for subsequent search and analysis. This requires metadata extraction and possibly data transformation.	For example metadata could describe when and where the data was collected. Metadata might also describe VE's and their social properties.	Met Fully - Consistent, Correct & Complete
4.3	There should be a mechanism to search for data according to its metadata.		Met Fully - Consistent, Correct & Complete
4.4	There should be a mechanism to perform data analysis.		Mostly Met - Not Complete
4.5	This mechanism would define APIs that are available to the application developer in order to implement application specific analysis.		Met Fully - Consistent, Correct & Complete
4.6	The mechanism for data analysis should enable computation to run close to the stored data in order to reduce the amount of data sent across the network.		Mostly Met - Not Consistent
4.7	Raw stream data processing (predict anomalies or off-normal events) should be possible		Met Fully - Consistent, Correct & Complete
4.8	System should offer CEP data persistence (post processing to detect behavior patterns).		Met Fully - Consistent, Correct & Complete
4.9	Publishing sub-system offer data broadcasting based on semantic analysis results		Not Met
4.10	System should provide meanings to define events taxonomy, including reasoning with unsafe/uncomplete events		Partially Met - Only Correct
4.11	System should provide the capability to define processing configurations/topologies, including fail safe configurations		Not Met
4.12	CEP capability should provide support to be used as as situation awareness tool.		Partially Met - Only Correct
UNI.432	COSMOS should provide a virtual identification system.	"A universal identifier should be defined as standard ID in order to map it to the specific ID used in every type of system (TCP/IP, RFID, ...)"	Met Fully - Consistent, Correct & Complete
6.0	The system must provide mechanisms in order to characterise objects (meta-data).	Provide a way to classify and relate Objects (e.g. location, nature of data, addressing, environment, availability).	Met Fully - Consistent, Correct & Complete
UNI.414	COSMOS shall enable the dynamic discovery of virtual entities and their services. This is to be done based on the specification of the service and the virtual entities.	"Augmented entities are the core concept proposed for IoT and to enable applications that do not have to be a-priori configured for a fixed set of augmented entities, discovery at runtime must be possible."	Partially Met - Only Correct
UNI.041	COSMOS could provide historical information about the physical entity.	"A method for clarification whether the Cold/Hot Chain has been violated or not is required. To be able to do this, the continuous context information (e.g., temperature) of the things needs to be collected. This is for example of major importance to avoid any damage to the pharmaceutics during the transport and storage process."	Met Fully - Consistent, Correct & Complete
5.1, UNI.509	The COSMOS system must provide mechanisms in order to characterise services, applications and experiences (e.g. for look-up purpose).	Provide a way to classify and correlate services and/or experiences (purpose charasteristics, actions, result).	Mostly Met - Not Complete
UNI.425	COSMOS must provide a service identifier and the identifier must use a service/resource description for retrieval.	"The system must consider the description of a service/resource for the semantic indexing on which the search will be performed."	Met Fully - Consistent, Correct & Complete
UNI.415, UNI.401	COSMOS could enable the dynamic discovery of virtual entities and their related services based on a geographical location or other geographical parameters.	"Geographic location is one of the most important aspects for finding relevant virtual entities. Spatial relations are of prime importance in the physical world."	Met Fully - Consistent, Correct & Complete
UNI.416	COSMOS must enable the lookup of service descriptions of specified services for a virtual entity with the VE identifier as key for the lookup.	"It is important to find the services related to a virtual entity that may provide information about the virtual entity, allow to actuate the virtual entity, enable interaction with the virtual entity or use them for the formation of new services."	Met Fully - Consistent, Correct & Complete
5.2	An XP taxonomy (or taxonomies based on other properties, characteristics or descriptions of the objects) could be developed and allow semantic look-up.	e.g. It is easier then for the VEs to find XP of other VEs rather than asking their "friends".	Not Met
5.3	COSMOS must provide mechanisms for automated grouping of simple objects into a complex object.	An VE could provide its attributes to the system which will match them with VEs metadata in order to create a GVE. e.g. although we might have the VE of a room, we can form the VE of a smart building by "merging"/connecting the VEs of its rooms.	Mostly Met - Not Complete
UNI.409	COSMOS must allow storage of VE changes, including structural changes (e.g. changes in the aggregation of multiple VEs constituting one Grou of VEs - GVE).	After the formation of a GVE, the changes in its structure or components (e.g. new VEs) must be monitored.	Met Fully - Consistent, Correct & Complete
5.4	Human Users (individuals and groups/companies/public services) should have their own representation in COSMOS (e.g. through the use of VEs of Human Users).	Through these we can know who is the admin of another VE and who has access to it. It can be decided which admin has priority on certain VEs (e.g. if an individual and a municipal/city service both need a public VE, priority will be given to the city service).	Not Met
UNI.046	The system shall support storage of user data.	e.g. The preferences of a bus passenger (e.g. buses, lines) could be stored for future use.	Partially Met - Only Correct
5.5	COSMOS must provide mechanisms for distributed data-storage (Cloud Storage).	"Provide an accessing mechanism to distributed data and latency (ie. P2P networks)"	Met Fully - Consistent, Correct & Complete
5.6	The system must be able to accept certain parametres that describe a new application request.	Using an application that already exists or requesting from the system the creation of a new service (from the services that can be offered by VEs of all kinds) based on these parametres.	Partially Met - Only Correct
UNI.426	COSMOS must be able to accept and manage semantic queries from the user and return Resources/Services.	COSMOS should have interfaces to enable the user make queries for the discovery, lookup and resolution functions.	Met Fully - Consistent, Correct & Complete
UNI.253	The orchestration engines could support setting preferences for selecting services involved in composition.	"Users can have the possibility to prefer one service over another for any reason."	Partially Met - Only Correct
5.7	Part of the input from an IoT Application Request could be a certain group of VEs that must/should/could be used (maybe forming a corresponding GVE for the application).	Using specific VEs depending on the Human User's needs. Accelerating the search. Users can have the possibility to prefer one VE over another for any reason.	Partially Met - Only Correct
5.8	COSMOS could get as input the classification of the App-Requests depending on the use cases (e.g. "waste management", "traffic control").	This would help to search for certain VEs (of all kinds) for the application.	Not Met
5.9, UNI.251	COSMOS must provide a feedback to the user who sent an application request.	Offering requested information message to the user, or, in case of an application using actuators, a success or failure/exception message.	Mostly Met - Not Complete
5.10	Service should remain available after ending its assingment to an application.	Service's life cycle has to include a period of persistency once the service is designed, in order to be invoked again from other app if needed.	Partially Met - Only Correct
5.11	The COSMOS system should offer mechanisms to build and maintain objects' reputations (according to various criteria). In addition the system must be able to quarantine an object, the reputation of which, does not meet given criterion	Management system has to be able to put in quarentine or out of service an object which has reached a predefined low level of reputation, to avoid spamming a service or application.	Mostly Met - Not Complete

5.12	It must be possible for an object to assess the quality of any contribution received from other objects.	Expediting the Evolvement Assessment / Raw Data Analysis, assisting the autonomicity of the system.	Mostly Met - Not Complete
UNI.422	COSMOS should enable the discovery and lookup of associations across multiple administrative domains.	"The Internet of Things will consist of multiple administrative domains with different owners that generally manage their devices, resources, services virtual entities etc. independently. To develop its full potential interactions, including lookup and discovery, across domain boundaries must be possible."	Mostly Met - Not Complete
UNI.406	The discovery service of the system shall support the following location queries: position queries, nearest neighbour queries, navigational queries, and range queries	"The location model shall support the following common location queries: position queries, nearest neighbour queries, navigational queries, and range querie."	Met Fully - Consistent, Correct & Complete
UNI.237	COSMOS shall offer data types for describing the quality of information related to virtual entities.	Different devices provide information with varying quality. An application may have certain quality requirements.	Mostly Met - Not Consistent
UNI.408	The system's services shall indicate what information can be found by a discovery/look-up service.	"Data that companies are willing to provide to the Discovery Services are mainly URL addresses of databases / EPCIS repositories"	Met Fully - Consistent, Correct & Complete
5.13	Objects in COSMOS could be able to publish some operational state (e.g. power status, availability, work load,..) and related predictions.	Each thing into the network should be able to use mechanisms to publish or provide its current state, also its future state based on known facts (battery time) or prediction based on experience (each 7 days object become unavailable because of... reset, low battery, maintenance window...)	Met Fully - Consistent, Correct & Complete
5.14	There should be a mechanism that filters data flows and detects situations (loss of connection, low data quality, data incongruences...) and compose data (averages) for improving network performance and provide a more useful info.	Each smart object or set of smart objects into the network must be able to use mechanisms to perform a light analysis of data and let the management mechanism take decissions.	Mostly Met - Not Complete
5.15	System must be able to classify events based on nature of data, source and evolvement patterns in order to detect/predict (undesired) states (e.g. availability, reliability, serviceability).	Events Identification like first stage of Monitoring function. e.g. Indicate that the object is going to become unavailable or it is no reliable.	Mostly Met - Not Complete
UNI.418-421	COSMOS must be able to track dynamic associations between virtual entities and services, taking under consideration parameteres such as location. This needs to be done in order to determine whether these associations are still valid.	"Due to the mobility of things, as well as devices whose resources are accessible through services, changing services may provide information, allow actuation, or enable interaction with things. In order to provide the currently relevant services for a thing, dynamic associations must be tracked to determine whether they are still valid."	Partially Met - Only Correct
UNI.214	The system's process modeling notation and monitoring could include a graphical representation.	"A graphical process notation offers a symbolism to easily model and document business processes."	Mostly Met - Not Complete
5.16	Evolvement Assessment: produces the optimal (having functionality in mind) subnetwork/subgroup of VEs to be used by the Decision Making mechanism that chooses the VEs that should form the GVE for running a new application/service.	Filtering the results of Monitoring/Evaluation in order to expedite Decision Making / Raw Data Analysis.	Mostly Met - Not Complete
5.17	The COSMOS system must be able to propose mechanisms dealing with the potential sudden unavailability of object at run time (for ensuring continuity of service e.g) [Runtime Adaptability]	Management system can change the Object assingned to a service whereas the service can continue offering its function. Whenever an object becomes unavailable, management system has to be able to find another one able to fulfill the service requests, if possible, using discovery algorithms based not only in Metadata but in Reputation also.	Met Fully - Consistent, Correct & Complete
UNI.701	COSMOS must accomodate fast developmental changes in applications and network.	"New applications can change traffic characteristics in a few months. In the past decade several applications dramatically changed the way how the Internet is used. Nobody has actually foreseen the success of P2P networks, and especially Youtube and Facebook. Thus, the question is whether it is possible to design a Future Internet without having any ideas what the "next big things" could be. If thus the traffic changes are unpredictable, then we need to establish a fast and stable infrastructure without any assumptions on the traffic."	Partially Met - Only Correct
5.18	COSMOS must be able to assess the quality of the Network of objects according to various criteria (like fragmentation, reliability, efficiency,...) and to make predictions about future state.	Obtaining Distribution/Decentralisation.	Met Fully - Consistent, Correct & Complete
5.19	It must be possible to monitor, in real-time, links between the different objects (e.g. social links, dependencies, etc...) under a specific context (e.g. an object collaboration).	Monitoring of the Networks of VEs based on certain parameters. Monitoring the links of the Network of VEs and its raw-data simultaneously.	Met Fully - Consistent, Correct & Complete
5.20	Evaluate simple events and events coming from different sources to detect more complicated facts.	A simple event about a temperature measurement with a really high value can indicate that the sensor is going to fail due to overheat. Or a complex event, like fire, could be detected by temperature sensors and smoke detectors in several smart-rooms.	Met Fully - Consistent, Correct & Complete
5.21	Event detection must apply both to individual objects and groups of objects.	During Decision Making, external events must be taken under consideration.	Met Fully - Consistent, Correct & Complete
5.22	The system should be able to determine the potential impact of an event.	Filtering the results of Events Identification, so that only the important events will reach Decision Making.	Met Fully - Consistent, Correct & Complete
UNI.235	Processing of events shall take quality of informaton (QoI) into account.	"In IoT the quality of information stemming from events is often questionable."	Mostly Met - Not Consistent
5.23	The COSMOS system could differentiate between "normal" (and expected) events and "abnormal" events"	Some events, although being quite important, should not be taken under consideration.	Mostly Met - Not Complete
5.24	The COSMOS system must be able to determine probable causes of an event and could be able to determine casuality between events.	A feature that could accelerate the events identification and impact assessment mechanisms, as well as provide more information on certain events, that could be used during Decision Making.	Mostly Met - Not Complete
5.25	An object involved in a collaboration with other objects (either as master, slave or peer) must be able to access the quality of collaboration as it perceives it. (applies to sharing of XP too)	This is quite crucial for the automated grouping of simple objects into a complex object.	Met Fully - Consistent, Correct & Complete
UNI. 714	The system management (Decision Making) shall take under consideration the device constraints such as energy and memory.	Although during Decision Making a temporary "optimal" GVE is formed, its structure might need to change because of such characteristics.	Met Fully - Consistent, Correct & Complete
5.26	The COSMOS system must be able to resolve conflicts in attempts to access or initiate collaboration between objects.	Many applications will ask for the same VEs (of all kinds) at the same time.	Met Fully - Consistent, Correct & Complete
UNI.027	COSMOS must support prioritization of services, depending on many characteristics.	"In case of time-sensitive services the system needs to assure that important services are prioritized." e.g. if an individual and a municipal/city service both need same resources, priority will be given to the city service	Met Fully - Consistent, Correct & Complete
5.27	An orchestration functionality within Decision Making is needed.	Crucial for distributed management and orchistrating the rest of the functional components.	Mostly Met - Not Complete
UNI.089	COSMOS shall support reliable time synchronization.	"Services which depend on a precise time need a guarantee that the devices they are communicating to have the right time."	Mostly Met - Not Complete
UNI.245	COSMOS shall support creation of new applications through the creation of new CVEs or other mechanisms.	"Composite services allow added value services based on simple services."	Partially Met - Only Correct
UNI. 707	Support for management operations	"The communication model must provide the basic management operations such as get, set, create, add, delete, act, and notify." Reference: *S. Kim, M. Choi, H. Ju, M. Ejiri, J. Hong, "Towards management requirements of Future Internet", In: "Challenges for next generation network operations and service management.", Springer, 2008, pp 156–166*	Mostly Met - Not Complete
UNI.031	COSMOS must enable centralised or decentralised automated activities (control loops).	"Today, due to sub optimal processes, a lot of time and money is wasted. This situation could be improved a lot by tracking all the items/things, providing context data on them at any time and location, allowing for automated evaluation of the collected data and reacting immediately on a dangerous situation to protect against the break down of items."	Met Fully - Consistent, Correct & Complete
5.28, UNI.015, UNI.100	COSMOS must be able to send orders (Action Triggering) and feedback (e.g. XP) to VEs. The COSMOS system must be able to perform actuation (potentially based on sensor information, but not only). e.g. COSMOS should include means to wake-up sleepy devices.	Based on evaluation, make a decision and trigger an action.	Met Fully - Consistent, Correct & Complete
UNI.508	COSMOS must support intermittent and command-based communication with devices.	"Avoid traffic overhead."	Met Fully - Consistent, Correct & Complete
5.29, UNI.010	The COSMOS objects must act autonomously according to their own objectives and plan.	Obtaining Decentralisation.	Met Fully - Consistent, Correct & Complete
5.30	The COSMOS system must be able to scale so that it can deal with large amounts of data and objects.	Big Data and network complexity are some of the main characteristics of all IoT applications.	Mostly Met - Not Complete
UNI. 704, 706, 708, 715, 719	COSMOS must exhibit self-management behaviour.	"There is no future for a centralised management (in most cases). It is necessary to move the research effort towards self-management approaches."	Met Fully - Consistent, Correct & Complete
5.31	The COSMOS system must achieve its management tasks in a decentralised manner.	"There is no future for a centralised management (in most cases). It is necessary to move the research effort towards self-management approaches."	Met Fully - Consistent, Correct & Complete
6.0	Time series of raw data can be regular and irregular		Met Fully - Consistent, Correct & Complete
6.1	Time/Space should be kept separated from other meta-data		Met Fully - Consistent, Correct & Complete
6.2	Mechanisms are needed for complementing incomplete series of data (based on interpolation or otherstatsitics-based technique)	Due to the network and energy constraints it might happen that an expected "regular" time serie contains gaps, refered in the DOW by "volatility" of information	Mostly Met - Not Complete
6.3	Precision of data time stamps should be at the level of one second		Met Fully - Consistent, Correct & Complete
6.4	All data must be stored by default within a data object		Met Fully - Consistent, Correct & Complete
6.5	Data should be indexed in time/space		Partially Met - Only Correct
6.6	It should be possible to perform prediction of measurements (VE properties) based on existing past measurements (extrapolation)		Mostly Met - Not Complete
6.7	I could be possible to estimate the accuracy of prediction		Partially Met - Only Correct
6.8	VEs (object) must be able to exchange experiences so that object can learn from each other	Therefore the service logics of an object A operating in similar conditions and with similar objective than an object B could be improved, and provide possibly better outcomes	Met Fully - Consistent, Correct & Complete
6.9	Mechanisms need to be implemented for Trust and Reputation between objects	There is a need for distinguishing between objects one can learn from and object that do not provide usefull or accurate enough experiences	Mostly Met - Not Complete
6.10	Mechanisms are needed to evaluate the impact of using another object's experience	Evaluating the expact of reusing an experience helps for assessing the trustability and reputation of an object	Mostly Met - Not Complete
6.11	Repository for VEs experiences is needed	We need a central plae where to store experience so that they can be looked up and retrieved easily	Met Fully - Consistent, Correct & Complete
6.12	It must be possible to annotate experiences so that they can be easily discovered with advanced (Semantic) search criteria		Partially Met - Only Correct
6.13	Trust & Reputation should be based on various criteria like Efficiency, reliability, responsiveness, commitment...	In the context of cooperation between object (e.g. experience exchange) different factors should be considered when establishing a VE reputation	Partially Met - Only Correct
6.14	It must be possible to describe an object skills (and purpose/objective) and to search based on those descriptions	Prior to the exchange of experience, identifying the purpose, nature and skill of an object is required	Partially Met - Only Correct
6.15	It should be possible to select the experience of an object based on the fullfilmeny of its objective. Higher success guaranties meaningful and useful experience	Some object experience may result in poor achievement or outcome compared with the initial objective. Those experience should be ignored by other objects and only good experience w/ good outcome should be reused	Mostly Met - Not Complete
6.16	It could be possible for an object to issue a Call for Tender, in order to advertise its specific needs and get experience-sharing proposals from other objects.	This is an alternative to 6.14	Not Met
6.17	Each level of processing from raw-data, information, event, knowledge... should be able to enrich the already existing meta-data		Mostly Met - Not Complete
6.18	It should be possible to select the experience of a VE, no matter its reputation level	A VE may have low reputation but with positive trend, or it is the only one which can offer its experience	Met Fully - Consistent, Correct & Complete
6.19	Mechanism should be implemented in order to select what kind of experience must be shared		Met Fully - Consistent, Correct & Complete
6.20	Predictive algorithms should be implemented so as to estimate whether the shared experience was efficient, helpful, etc.		Partially Met - Only Correct
6.21	All information about situational awareness has to be associated with context.	Same data in different situation could have different meanings.	Met Fully - Consistent, Correct & Complete

6.22	A perception of all relevant things within current configuration.	The perception of things within current network provides the basis for situational awareness. (check 6.12)	Met Fully - Consistent, Correct & Complete
6.23	A things should provide information about their behavior.	A knowledge about status, dynamic behavior, attributes of things is necessary to evaluate situational awareness (especially at real time).	Met Fully - Consistent, Correct & Complete
6.24	It must be possible to detect malfunction of things.	In order to avoid negative impact on application/business level.	Met Fully - Consistent, Correct & Complete
6.25	Things should provide information about operational constraints	E.g. limited power source, operational range, etc...	Met Fully - Consistent, Correct & Complete
6.26	Identify and utilize redundant information.	Different services may provide semantically identical information. For example portable/mobile devices temporary at same location.	Mostly Met - Not Complete
6.27	Inappropriate collaboration between smart autonomous devices shall be automatically detected.	Some objects may influence correct behavior of other objects. This include e.g. conflicting integration of new device into network, missing support for versioning etc...	Mostly Met - Not Complete
6.28	Suspicious/unexpected behavior of things should be automatically detected.	To collect and evaluate awareness knowledge from experience/historical behavior or unstable quality of data streams at real time.	Mostly Met - Not Complete
6.29	It must be possible to evaluate probability of service termination.	In order to better evaluate situation on the network and make appropriate decision, it is helpful to filter "expected" termination of service from accidental termination of operation. For example some portable devices operate only during emergency situation or limited time periods, etc...	Partially Met - Only Correct
6.30	Mechanism to discover services that supersedes existing services.	Maybe better requirement description would be: "Devices providing higher quality of service for same information should be detected and preferred." The idea is to detect such situation at real time.	Partially Met - Only Correct
6.31	It must be possible to distinguish emergency situation.	Some situations require immediate reactions.	Met Fully - Consistent, Correct & Complete
6.32	Situation of network/things must be evaluated accurately and completely.	The idea is that without having a overall/complete picture about state of whole network, it is not feasible to interpret situation just from information provided by particular devices. In general, any information has different meaning in different context that's why overall context is important.	Met Fully - Consistent, Correct & Complete
6.33	Filtering of "false positives".	For correct interpretation of situational messages, we have to separate intended behavior from unexpected behavior.	Mostly Met - Not Complete
6.34	It must be possible to select the experience of a VE, no matter its reputation level	A VE may have low reputation but with positive trend, or it is the only one which can offer its experience	Met Fully - Consistent, Correct & Complete
6.35	Mechanism should be implemented in order to select what kind of experience must be shared	This depends on the specific application's demands	Met Fully - Consistent, Correct & Complete
6.36	Predictive algorithms must be implemented so as to estimate whether the shared experience was efficient, helpful, etc.	This influences the trust & reputation score of a VE	Partially Met - Only Correct
6.37	Several levels regarding trust & reputation evaluation could be recognised	Evaluation coming from COSMOS platform (objective level), evaluation between VEs (subjective level)	Met Fully - Consistent, Correct & Complete
7.1	Some virtual entities will have unidirectional flow, i.e. they will publish their status to be used by other virtual entities. Others will have bidirectional flow, i.e. They will publish and consume the status of other entities	Certain VE may not need to use data from other	Met Fully - Consistent, Correct & Complete
7.2	All virtual entity will have the version data attribute for each element in the structure of their information	In order to simplify the scaling system	Met Fully - Consistent, Correct & Complete
7.3	Any entity will be accessible in real time	Logically, any real-time information must be provided in real time	Met Fully - Consistent, Correct & Complete
7.4	Virtual entities with a geographic component will display the geographic coordinate where the event occurred	Each VE should report its geographical position, if necessary	Mostly Met - Not Complete
7.5	Every entity will disclose time validity of its data (in number of seconds)	Define its time validity	Met Fully - Consistent, Correct & Complete

9.2 Evaluation Percentages

Met Fully - Consistent, Correct & Complete	53.1%
Mostly Met - Not Consistent	2.3%
Mostly Met - Not Correct	0.0%
Mostly Met - Not Complete	22.7%
Partially Met - Only Consistent	0.0%
Partially Met - Only Correct	15.6%
Partially Met - Only Complete	0.0%
Not Met	6.3%