# COSMOS

Cultivate resilient smart Objects for Sustainable city application

Grant Agreement Nº 609043

# D7.4.2 Smart heat and electricity management: Evaluation and recommendations (Updated)

## WP7 Use Cases Adaptation, Integration and Experimentation

**Version:** 5.0
**Due Date:** 12/02/2016
**Delivery Date:** 12/02/2016
**Nature:** R
**Dissemination Level:** PU

**Lead Partner:** Hildebrand Technology Limited
**Authors:** Joshua Cooper
Abie Cohen
Susana Espino
Katie Longman
**Internal Reviewers:** Juan Rico
Achilleas Marinakis

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 | Hildebrand | Initial write-up |
| 0.2 | 11 March 2015 | Joshua Cooper | Hildebrand | Initial review |
| 0.3 | 22 March 2015 | Abie Cohen | Hildebrand | Review changes and updates |
| 0.4 | 23 August 2015 | Joshua Cooper | Hildebrand | Penultimate review and updates |
| 1.0 | 26 October 2015 | Abie Cohen | Hildebrand | Final review |
| 1.1 | 27 October 2015 | Achilleas Marinakis | NTUA | Internal review |
| 2.0 | 27 October 2016 | Abie Cohen | Hildebrand | Penultimate changes |
| 2.1 | 29 October 2015 | Juan Rico | ATOS | Internal review |
| 3.0 | 30 October 2015 | Abie Cohen | Hildebrand | Final changes |
| 4.0 | 26 January 2016 | Susana Espino | Camden | Review following Commission's comments |
| 4.1 | 29 January 2016 | Angela Murphy | Camden | Review following internal review |
| 4.2 | 04 February 2016 | Katie Longman | Hildebrand | Penultimate review and updates |
| 4.3 | 07 February 2016 | Juan Rico | ATOS | Internal review |
| 5.0 | 11 February 2016 | Katie Longman | Hildebrand | Final changes and formatting |

| Nº | File Name | Title |
|---|---|---|
| 1 | Appendix A | Risk Assessment Table |

**Annexes:**

# Table of Contents

## Acronyms:

| Acronym | Meaning |
|---|---|
| 3DES (TDES) | Triple Data Encryption Algorithm |
| API | Application Programming Interface |
| CBR | Case-Based Reasoning |
| CEP | Complex Event Processing |
| CHP | Combined Heat and Power |
| COSMOS | Cultivate resilient smart Objects for Sustainable city applicatiOnS |
| D | Deliverable |
| FC | Functional Component |
| GPIO | General Purpose Input Output |
| GVE | Group VE |
| HMM | Hidden Markov Model |
| IoT | Internet of Things |
| ML | Machine Learning |
| SQL | Structured Query Language |
| SVM | Support Vector Machine |
| VE | Virtual Entity |
| VE2VE | Ve to VE |
| WP | Work Package |
| XP | Experience |
| Y | Year |

# 1. Introduction

This deliverable focuses on re-evaluating the adoption of the COSMOS technologies in real-world Smart City cases. We will use the Use Case Scenarios formulated in Year 2 as the basis of our assessment. We will also re-evaluate the technologies by looking at their assessments from Year 1, and at their current consistency, correctness and completeness.

Once again, these measures can be expanded into a list of criteria with which we will evaluate each of the technologies in WP3-WP6 for Year 2. In our use of these measures, we will implement a cost versus benefit 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 recommend what our next steps should be for Year 3, and where we should focus our efforts in terms of research activities and productivity.

In this Work Package deliverable we will:

- Provide background material describing the current situation in each of the new Use Case Scenarios from Year 2, as well as the ones from Year 1;
- Define a clear set of evaluation criteria for any given technology;
- Re-identify the technologies used in COSMOS, particularly the ones implemented in Year 2 prototypes;
- 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 extended recommendations based on these 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

We will follow the same methodology for evaluating the COSMOS technologies as we used in D7.4.1.[1]

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 be applied to specific Use Cases.

We will then collect and list all of the technologies used in COSMOS, which have been described in WP3-WP6, looking at the London Use Case Scenarios and evaluating how each of the technologies will be used. This is done in a very structured manner, using the criteria to test different aspects of the technologies.

Next, we will consider the requirements in D2.2.2[2] and assess whether the use of the technologies developed in COSMOS will meet with necessary standards and solve any corresponding issues. These are the *Updated* requirements from Year 2.

Finally, based on results from the evaluation of the technologies in each of the new Smart Heat Management Use Case Scenarios, we will recommend Next Steps to take in COSMOS. Using the costs and benefits we find the technologies provide in different domains, we can help to direct the research activities of the project.

# 3. Evaluation Criteria

In this deliverable, we reiterate the criteria that we use to evaluate the COSMOS technologies: the same evaluation criteria that were used in the D7.4.1 deliverable.

In order to properly assess the efficacy of the technologies used in COSMOS and their applications to the different Use Case Scenarios, we need to test them against a set of criteria. This ensures that each evaluation is fair and comparable.

It is important that these criteria capture the key measures for evaluating technology: consistency, correctness and completeness. The terms are defined as follows:

**consistency** how technically feasible, reliable and extendible is the technology

**correctness** does it satisfy the problem at hand and how convincingly

**completeness** is it actually acceptable to implement such a technology, weighing pros against cons.

When looking at these measures, we realised that they can be broken down further into fundamentals, and grouped into four main criteria blocks: functionality, construction, realisability and impact.

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

        The extent to which the designed product satisfies requirements.

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

    b. *Ease of use*

        Users' ease of use, 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. This includes scalability and the ability to be used in (dis)similar contexts.

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

2. **Construction**

   a. *Structuring*

      The partitioning of the product in logical or physical components.

      What architecture is used? How complex is the system? How do the different components in the technology communicate with each other, and how efficiently do they do so?

   b. *Convincingness*

      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 can take place? Has it been used in another COSMOS component? What is the likelihood of the problem being solved by using this technology as a solution?

3. **Realisability**

   a. *Technical realisability.*

      The level of certainty that it is technically possible to produce the technology.

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

   b. *Economical realisability.*

      The business case for the technology.

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

4. **Impact**

   a. *Risks.*

      Risks during development or use stages.

      Does the technology introduce new problems? Are there any privacy or security issues inherent to this technology? Are there authorisation 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.[3] 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 the problems that cities currently experience. The two Use Case Scenarios that we will focus on are Heating Networks in the London Borough of Camden and the Bus System in Madrid. In order to solve issues that arise in the two Scenarios, we developed certain technologies in WP3-WP6. These technologies, when combined, produce the overall COSMOS system, ranging from hardware to software, and from servers to sensors. Each of the technologies mentioned in this section fulfils a specific role and has a purpose in COSMOS.

This section provides us with a clear list of the technologies used in COSMOS (Year 2), described fully in WP3-WP6, and whose applicability is described in the next section:

**(WP3) D3.2.1:** End-to-end Security and Privacy[4]

- Privelets
- Node-RED Security

**(WP4) D4.2.1:** Information and Data Lifecycle Management[5]

- Integration with Analytics Framework

**(WP5) D5.1.1:** Decentralised and Autonomous Things Management[6]

- The Planner
- Social Analysis
- Network Runtime Adaptability Module

**(WP6) D6.1.1:** Reliable and Smart Network of Things[7]

- Inference/Prediction Functional Component
- Pre-processing Functional Component
- Event (Pattern) Detection Functional Component

# 5. Use Case Scenarios

## 5.1. London Borough of Camden Heat Network

### 5.1.1. Use Cases

#### 5.1.1.1. Heating Control

| |
|---|
| **Use Case:** Heating Control |
| **ID:** 5 |
| **Brief Description:** The EnergyHive system is measuring the temperature of the properties where it is installed and has the ability to control the delivery of heat through a valve. A new tablet has been deployed within the property that allows for a set point and schedule to be entered. Feedback from users has been that they would like the system to automatically help them set a programme and manage efficiencies on an ongoing basis, for instance: detection of whether or not they are at home; using the weather forecast to help with program and supply-side management when the solar thermal is available for use. The tablet is a COSMOS-compatible device and it can act locally to run case-based reasoning in an efficient manner. |
| **Primary Actor(s):** Resident |
| **Secondary Actor(s):** Mechanical and Electrical Engineer; Sustainability Officer |
| **Preconditions:** EnergyHive system must be installed within a resident's premises. |
| **Main Flow:**<br>1) Resident will select an autopilot function on their tablet.<br>2) Autopilot will determine a recommended set point for the temperature in the house.<br>3) Set point can be overridden by the resident.<br>4) The system will learn the patterns of occupation and adjust the run programme to turn off the system based on un-occupied property; the resident can override.<br>5) Savings should be quantified over using a normal time-based programmer. |
| **Postconditions:** An improvement in the efficiency in the heating system should be reported. |

#### 5.1.1.2. Building Performance Management

| |
|---|
| **Use Case:** Building Performance Management |
| **ID:** 6 |
| **Brief Description:** The boiler systems within buildings have master programmers and temperature settings that are controlled by a Trend boiler control system. There are also verification instruments installed within buildings to measure the effects of the boiler control; they can provide feedback to inform the run-time commands to the boiler control as well. A more granular view of the energy demand, including trade-offs with electricity usage, is desired so that individual residential premises are getting higher comfort while balancing the energy input. |
| **Primary Actor(s):** Mechanical and Electrical Engineer; Sustainability Officer |
| **Secondary Actor(s):** Energy Performance Officer; Resident |
| **Preconditions:** EnergyHive system must be installed throughout each building in the estate, as well as boiler controls and verification systems. |
| **Main Flow:**<br>1) Temperature readings are collected at distribution level within Camden heat networks.<br>2) The energy balance model will be run against the Trend readings, and the temperature/electricity readings showing performance indicators (degree hour per kWh) against a network model for the delivery.<br>3) Normalisation for seasons and weather conditions should be applied (subtract degree hours inside versus degree hours from weather).<br>4) Sensors will be installed wherever suitable on the district heat network to manage distribution losses. |
| **Postconditions:** Ranked performance of the buildings' heat networks is reported to enable interventions to improve network inefficiencies. |

#### 5.1.1.3. Capital Planning/Energy Performance

| |
|---|
| **Use Case:** Capital Planning/Energy Performance and Commissioning and Quality Assurance |
| **ID:** 1 |
| **Brief Description:** The EnergyHive system in each building enables Capital Planning/Energy Performance Officer 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 Actor(s):** Capital Planning/Energy Performance Officer |
| **Secondary Actor(s):** Mechanical and Electrical Engineer; Sustainability Officer |
| **Preconditions:** EnergyHive system must be installed throughout each building in the estate. |

| |
|---|
| **Main Flow:**<br>1) Sustainability Officer identifies an opportunity for environmental improvement of system.<br>2) Engineer selects appropriate technology for instalment.<br>3) EnergyHive system provides detailed information as to the effect of the change in the system.<br>4) Capital Planning/Energy Performance Officer uses EnergyHive information to assist in cost/benefit analysis. |
| **Postconditions:** The Capital Planning/Energy Performance Officer decides whether to roll out the proposal. |

#### 5.1.1.4. Identification of Opportunities

Using machine learning, identify where energy savings opportunities exist. This will help Sustainability Officers to suggest projects that can then be put through the Capital Planning Use Case.

| |
|---|
| **Use Case:** Identification of Opportunities |
| **ID:** 7 |
| **Brief Description:** The EnergyHive system running in planning mode can use machine learning to suggest opportunities for efficiency. This is largely an unsupervised learning exercise where cause and effect models can be run with comparisons to other like buildings or similar conditions that have been observed. |
| **Primary Actor(s):** Sustainability Officer; Energy Performance Officer |
| **Secondary Actor(s):** Rents and Billing Services; Air Quality Officer |
| **Preconditions:** EnergyHive system must be installed throughout each building in the estate. |
| **Main Flow:**<br>1) Sustainability Officer creates model constraints for parameters to optimise (i.e. cost or carbon savings desired with physical systems).<br>2) Model runs within system bringing up bands of savings that can be made from changes in input parameters.<br>3) System provides control ranges that would have to be implemented in order to make potential savings. |
| **Postconditions:** A quantified opportunity for efficiency within the energy system is presented for evaluation. |

### 5.1.1.5. 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 in half-hourly intervals and manage the CHP and boiler accordingly. |
| **Primary Actor(s):** Resident |
| **Preconditions:** Specialised Instalments:<br>1) Gas Flow meter to CHP from boiler to regulate the Gas supply.<br>2) Control system with temperature sensor on boiler.<br>3) Flow meter/temperature sensor on Solar Thermal.<br>4) Heat meter in each dwelling.<br>5) Communication infrastructure between sensors and hub. |
| **Main Flow:**<br>1) System predicts the estate's heat and electricity demand for a half-hour period.<br>2) System calculates required gas supply and distributes to CHP and boiler accordingly.<br>3) Carbon produced is measured.<br>4) Individual resident heat consumption is monitored. |
| **Postconditions:**<br>1) The resident is charged for their personal heat consumption.<br>2) Prediction errors are logged to improve system on later iterations. |

### 5.1.1.6. 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, with support from a council Sustainability Officer, assists the user in setting a heating schedule in accordance with their budget. |

| |
|---|
| **Primary Actor(s):** Sustainability Officer; Energy Performance Officer; Resident |
| **Preconditions:**<br>1) EnergyHive system implemented in each dwelling.<br>2) Valve up/down control system to the radiator. |
| **Main Flow:**<br>1) Resident accesses their customer account to view balance.<br>2) Resident can set a heating schedule.<br>3) Resident is given tariff and projected balance for a given schedule. |
| **Postconditions:** User can optimise their schedule to minimise their consumption. |

## 5.1.2.Residents' Health and Safety

Camden currently provides a well-being service to their residents called 'WISH Plus'. This provides a way for Camden residents to improve their health and well-being by using a range of Warmth, Income, Safety and Health services, all under one roof.

Visits to residents conducted in preparation for the COSMOS project have provided additional opportunities to make referrals to the WISH Plus team.

The Equality Impact Assessment (EIA) undertaken for the Heat Metering project has been used to determine the profile of residents who will potentially be using the COSMOS platform, and has been taken into consideration when selecting the sample of volunteers for the Pilot.

Risk Assessments were also conducted for residents taking part in the Pilot (see Appendix A). The assessments concluded that any potential risks arising from the Pilot can be controlled or mitigated by the implementation of the measures described in the Assessments.

Improving the health and welfare of city dwellers, and making warmth affordable and manageable, especially for those at risk of fuel poverty and the related illnesses that poor or unaffordable heating causes, are all at the heart of this Use Case Scenario. In addition, the Pilot will be run in accordance with Camden's stringent corporate Health and Safety guidelines, valuing its diversity ethos and strategic objectives as set out in the Camden Plan.[8]

## 5.1.3.Technologies

### 5.1.3.1.End-to-End Security and Privacy

The Privelets component is responsible for the authentication process on top of any kind of VE2VE communication in order to avoid possible VE impersonation. VE2VE communication includes:

- Accessing a VE's IoT service from another VE;

- Decentralised discovery;
- Recommendation service between VEs;
- Information (e.g. Experience) Sharing.

By preventing VEs impersonation, the Privelets component gives added value to the Trust and Reputation model introduced in the context of WP5, while by ignoring repetitive requests, it enables the VEs to protect themselves against wasting valuable computational resources caused by malicious attacks.

The Privelets component's source code is developed in Java programming language and depends on Jetty Server, Apache-Jena, Pellet-Jena, JSON-simple and other Java libraries. The prototype also relies on FreeLan, an open source software, in order to establish the COSMOS VPN and connect it with the VEs.

Privelets satisfy the requirement of adding an authentication layer to the hardware in a simple yet robust way. The programming language and associated technologies are easy to use and widely well-known. It is easy to design, implement and, most importantly, maintain. Furthermore, an additional benefit of using the object store technology is that it ties in well with the COSMOS other elements, as they are all built with the other components in mind.

In terms of construction, they are very straightforward and intuitive to set up in the system. Their efficacy and reliability, however, remains to be seen, as this will be proven during larger-scale prototypes later on in the project. This is important to note as simple testing may not show up all issues and/or limitations.

Technically speaking, it is relatively simple to use this technology in the scope of COSMOS. It could be argued that this is excessive, and merely to satisfy requirements. While it is true that this technology is fairly new and untested, and perhaps a more established technology could have been used to ensure efficacy, we might argue that the choice of using Privelets is fully justified as we are trying to build an innovative Smart City solution with advanced capabilities and functionalities.

There is no real risk of using this technology, as it deals with the authentication layer of the components. Of course, any reliability issues that occur would create huge risks and require an entire re-think of how the issue can be tackled.

The visual tool for wiring the Internet of Things is a platform-independent software framework that has been developed bearing in mind its usage not only in big servers or the cloud, but also in small, embedded computers such as the Raspberry PI, Arduino and similar others. Traditional IoT development can be very technical: access to the GPIO and other hardware modules requires skills in C or assembler, and output of data to web services or sending tweets and emails requires the use of complex APIs. Node-RED[9] takes care of the technicalities to allow focus to remain on the logic of the workflow. While most programming in Node-RED is done visually using pre-defined functions or 'nodes', any additional functionality can be added in JavaScript.

This new technology is a brilliant example of how to build a Smart City using state-of-the-art technologies that satisfy current system requirements and needs. Its simple-to-use interface and wide variety of plugins and components makes Node-RED a scalable and sharable technology.

The interface makes structuring clear and simple, and the concept of data flows inspires confidence that the components using this technology will run smoothly and as expected. In terms of technical realisability, it could be argued that this technology is perhaps too much work to implement for the job that it satisfies. Similarly, it may be possible to find a more lightweight technology to satisfy the same requirements without the expensive frills of Node-RED.

#### 5.1.3.2. Information and Data Lifecycle Management

In COSMOS, we store historical data for VEs in object storage (based on OpenStack Swift) in a format that is amenable to access by Spark SQL for the purpose of analytics of the data. Our driver allows optimised access to this data. All COSMOS services that make use of historical data can benefit, including applications that use machine learning on historical data and applications that analyse or report on past behaviour or the activity of VEs.

The driver is used to help access data from the COSMOS storage system, and is tailored to suit the particular structure and implementation style of the object store. In terms of functionality, this driver is limited in one sense, as it is specifically designed to work with OpenStack Swift and Spark SQL, but it could potentially be used when developing the aforementioned technologies in Year 3.

Due to the fact that it is a driver just for data access, its structure and convincingness remains strong and intact. As it simple ‘works’, we have no reason to question its setup or ability to provide easy and cheap access to all of the data being stored in the COSMOS servers.

In terms of realisability, both technically and economically speaking, this technology is certainly feasible, as it provides us with exactly what we need while being lightweight and cheap. It should be noted however that the development costs of this technology would have to be higher than alternatives in order that it works with the current setup that COSMOS is using.

Once again, we have assessed the risks of this technology and concluded that there are no major concerns in using the driver to access the data from the object store.

#### 5.1.3.3. Decentralised and Autonomous Things Management

As is stated in D5.1.1, the functional component that enables the VEs to use CBR is the Planner. The Planner will become part of the VEs during their registration time and will run locally. The main functionality of the Planner is to provide the VEs with the ability to react to problems. This uses a reasoning technique for finding the most appropriate solution to be applied based on similarities to previously encountered or experienced situations. This is a step towards the autonomy of the VEs, as depicted by the goals of Task 5.2 (Autonomous and Predictive Reasoning of Things).

The interaction metrics (Shares, Assists and Applauses) monitored by the Social Monitoring component of a VE are stored locally in the corresponding Followees Lists.

These metrics are calculated in a distributed manner by the VEs on a per-VE basis, and are the main input for the services provided by the Social Analysis and Friends Management components. The social indexes of the several kinds of Friends are extracted from these metrics. These are the Trust Index, the Reputation Index, the Reliability Index and the Dependability Index. Since the social indexes will constantly change, it is important to take their evolution into consideration. Although a VE may have a low Reputation Index when we study a wide time-window, it may have a much greater Reputation Index when we study a smaller and more recent time-window. This means that the specific VE is improving, and this improvement should be evaluated fairly by the system and the community. For this reason, for each Followee in each Followees List, the timestamps (unix time) and the evaluation of the last 3-10 interactions, for example, may be kept so that, when applying simple rules, the evolution of the indexes can be studied.

The Network Runtime Adaptability module is able to dynamically assign resources performing different activities within COSMOS architecture. In this sense, the key objective of this module is to control the resources usage of every single component. The monitoring of resources usage enables the optimisation and prioritisation of processes inside a VE. Besides this, the same functionality can be used in a multi-CPU environment for distributing the computational load, and thus minimising the risk of blocking processes.

These technologies allow the VEs to use the CBR technology autonomously so that decisions are made locally with the knowledge of a decentralised system. Although it does achieve an extremely powerful logic base, it is a complicated system to manoeuvre and needs to be designed with care and caution. Having said this, the concept behind this technology is extremely scalable, and can be extended to the Trust and Reputation models, as well as help integrate autonomy in other COSMOS components.

As the concepts introduced by these technologies are quite convoluted and highly theoretical, its convincingness unfortunately remains to be seen. The structuring of the Planner and the Social Monitoring components in the prototypes will be the pivotal point of assessment for this technology.

The gains from localising these technologies and attaining a decentralised autonomous system are clear. However, reality dictates that this technology could very well be technically insufficient in terms of reliability and consistency for the corresponding cost and effort.

As these technologies are most beneficial when integrated into the system locally, the added security risks are one of our concerns. Extra time and effort must be spent in WP3 to ensure that the hardware is encrypted and reliable. Furthermore, making decisions locally means that it is more difficult to see how the system is operating as it creates an extra point of failure (although, on the other hand, it also allows the system to operate without internet connectivity). In this sense, the Trust and Reputation model becomes critical when moving from model to real life, because damage caused by malicious users could produce very relevant losses. Effectively, it is a trade-off between scalability and autonomy failures.

#### 5.1.3.4. Reliable and Smart Network of Things

The Inference/Prediction component is responsible for analysing raw data in order to provide high-level knowledge which can be used for automated, proactive and intelligent applications. In this regard, we have implemented several pattern recognition algorithms on the Use Case Scenarios, such as different variants of Support Vector Machines (SVM) and Hidden Markov Models (HMM) for inferring high-level knowledge.

We have also explored several regression mechanisms for time series prediction of data for providing proactive solutions for Smart City applications. In this chapter, we briefly explain the architecture, interfaces and application of the component with the help of a Use Case Scenario.

Pre-Processing is a generic component, and different components such as inference/ prediction FC and Event (Pattern) detection can use it according to their requirements. It involves several functions, ranging from simple data cleaning mechanisms to more sophisticated mechanisms, such as data aggregation or feature scaling.

The Event Detection component is intended to provide the functionality for the near real-time processing of data for Event detection, by providing a hybrid solution based on Complex Event Processing (CEP) and Machine Learning (ML) methods. CEP provides a distributed and scalable solution for analysing data stream in near a real-time manner, but it does not exploit historical data, and the manual setting of rules is a major drawback. Rules for CEP are static, and hence solutions provided by CEP are static as well. Though ML methods exploit historical data and provide more automatic solutions, they are unable to provide near real-time solutions, and scalability is a major issue. In our proposed solution, we exploit both approaches and combined them in order to provide a near real-time solution that is more context-aware, adaptive and exploits historical data.

All of the components developed in WP6 in Year 2 aim to bridge the gap between data and decisions by means of pre-processing, data aggregation, feature manipulation and, primarily, machine learning techniques. The ease of use of these components decreases as we work on more state-of-the-art statistical models and look to solve more complex problems. The big benefit of developing an array of tools and components to handle data from point of extraction to point of decision is that the technologies are highly reusable and extendable. A lot of the fundamentals from the Event Detection component and the Inference/Prediction component can be easily applied and integrated into other COSMOS components, such as the CBR Planner developed in WP5.

Once again, the structuring of these components ranges, becoming more challenging as we approach the more theoretical and experimental machine learning models. Similarly, in terms of convincingness, we do not have a well-established best method of modelling the data sets available to us. Once the prototypes for the Use Case Scenarios are fully up and running and are heavily relying on these components, we will see whether they are able to reliably and accurately provide useful predictions.

The benefits for the CEP are clear motivations for the development of these components. However, once again, we must look at the economical realisability of some of the more theoretical and experimental modelling components. The time and effort spent on developing these techniques could prove to be wasteful if the results are not impressive or that of a state-of-the-art technology. Fortunately, the version of CEP developed in COSMOS is able to run on simple devices, and hence goes beyond the capabilities of current technologies, both in terms of ease of use and target platforms. The improvement assessment provided by the more theoretical contributions will help in the identification of their further potential, as well as the number of domains that will be targeted.

Finally, we have assessed the risks of these predictive tools and evaluated whether they expose the COSMOS system too much, especially when dealing with vulnerable citizens in a Smart City. We have concluded that any risks seem to be minimal, due to the testing and positive results shown in WP6.

# 6. Requirements

In this section, we will evaluate the list of 106 requirements put together over the course of COSMOS that has been provided in the WP2 deliverables. We will assess our progress for each of the requirements according to the three main criteria: consistency, correctness and completeness. We aim to achieve all three criteria for each of the requirements, as this demonstrates that we have fully satisfied the needs of COSMOS.

The percentage number of requirements that meet these different levels of compliance is presented in the following table (Figure 1) according to work done to date:

| Compliance Level | Number of Requirements | Percentage Number of Requirements |
|---|---|---|
| **Fully met:** consistent, correct and complete | 66 | 63% |
| **Mostly met:** not consistent | 4 | 4% |
| **Mostly met:** not correct | 0 | 0% |
| **Mostly met:** not complete | 19 | 18% |
| **Partially met:** only consistent | 0 | 0% |
| **Partially met:** only correct | 9 | 9% |
| **Partially met:** only complete | 0 | 0% |
| **Unmet** | 6 | 6% |

Figure 1. Percentage number of requirements meeting different compliance levels.

Approximately two thirds of the requirements have been 'Fully Met' (are consistent, correct and complete) 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 the Use Cases in each of the Scenarios has been noted in Section 5, and is verified in our evaluation of the requirements.

About 20% of the requirements have been 'Mostly Met' (are consistent and correct, but not yet complete), as the aforementioned technologies solve the issues that the requirements propose, and do so 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.

Fewer than 10% of the requirements have only been 'Partially Met' (are correct, but not yet consistent or complete), which is a pleasing result at this stage of the process. These requirements have the potential to be met with the use of the aforementioned technologies. However, it is just the theory behind these technologies that lead us to believe that problems can be solved. In terms of ensuring that the evaluation criteria can be met across the board without any loopholes or errors, the requirements that fall into this category currently fall short.

## 6.1. Unmet Requirements

The 6 requirements listed in the table below (Figure 2) have been marked as 'Unmet' as there is not enough 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 fully meeting the evaluation criteria.

| Unique ID | Description |
|---|---|
| 253 | The orchestration engines could support setting preferences for selecting services involved in composition. |
| 89 | COSMOS should support reliable time synchronization. |
| 245 | COSMOS must support the creation of new applications through the creation of new GVEs or other mechanisms. |
| 6.14 | It must be possible to describe object skills (and purpose/objective) and to search based on those descriptions. |
| 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. |
| 6.37 | Two levels regarding Trust and Reputation evaluation should be recognised. |

Figure 2. Unique ID Number and description of 'Unmet' requirements.

Three of the Unmet requirements are in WP5. The concept of Experience Sharing has been discussed in depth regarding its usefulness and the benefits that it could provide COSMOS. However, no real implementation of an orchestration engine or similar framework has been proposed, and therefore cannot be marked as a 'Met' requirement. Similarly, there has been no mention of how we plan on precisely implementing time synchronisation across every COSMOS component, and how we plan on maintaining this. Finally, we have discussed the IoT reference architecture and how VEs are structured and fit into the domain model at great length. However, to date, no work has gone into explaining the process of using these GVEs to develop applications.

The other three Unmet requirements are in WP6, which deals with the Network of Things. There is no clear explanation of the 'skills' metadata and how we expect to filter across it or use it in a search query. The VE level of the Trust and Reputation model has been made clear. However, the objective level has not yet been explained.

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 look at ways of advertising these needs and characteristics, and attempt to implement them in a Use Case Scenario to test its efficacy.

# 7. Recommendations

## 7.1. Overall Recommendations

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

### 7.1.1. Privelets and Node-RED

To achieve end-to-end security over all COSMOS components, we utilise the Privelets technology and implement Node-RED as a linking component. We must ensure that all standard protocols are followed to ensure that the encryption is secure and that these technologies are not so heavy that it will cause latency in the system.

### 7.1.2. Planner and Experience Sharing

Another important recommendation is to find the best way of allowing the VEs to communicate their experiences, and 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 constantly look 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 extend and refine it.

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

### 7.1.3. Machine Learning

In terms of analytics, we should use the comparisons of different Machine Learning techniques for classification done in D2.2.3[10] and regression for archetypical Use Case Scenarios such that general re-use is possible. Researching many possible ways of modelling our system so that End Users can interact with these complex technologies is of paramount importance to 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 if 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 ensure that there are no conflicts in experience ratings causing poor results.

### 7.1.4. 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 problems, such as missing data values and infrequent data transfers. So far, this issue seems to have been either accepted or overlooked, but it is important to find ways of ensuring that the data is regular and complete as the entire COSMOS platform relies on it.

# 8. Conclusion

It has been possible to assess technology that is new and improved for Year 2. Most of the evaluation has been done on either prototypical systems or mostly implemented technologies.

Our evaluation is optimistic for Year 3 in key areas where innovation is occurring, namely the CBR Planner, Privelets and UrbisAPI. We have observed that:

- Using a decentralised approach allows COSMOS to benefit in terms of efficiency. CBR is making a big impact such that low-resource devices can become intelligent, Case Bases can be exchanged for Experience Sharing and generalisation for CBR can be widely applied.
- Privelets allow us to authenticate communication between devices, whilst the use of Node-RED allows us to streamline these processes and link many components together in an intuitive way.
- Our Next Steps for the UrbisAPI platform will be to add additional key functionalities of User Interface components, so that it is ready to be sold to cities for IoT. In addition, the hardware will be industrialised so that it is durable and compatible with any market and industry, facilitating the sales of UrbisAPI to as many cities as possible.

There is clearly more work to be done in Year 3 to integrate COSMOS services and align them with the IoT-A reference architecture. City systems are expressing a clear interest in adopting IoT, so working systems that are able to realise business processes be sure to have a large impact.

# 9 Apendix A

## 9.1 Risk Assesment Table

| No | Key Objective / Identified Risk(s) [Threat/Opportunity] | Current Controls | Assessment of Risk Score<br>As it is now with current controls | | | Action Plan to Improve Controls and/or additional control measures | Assessment of Residual Risk<br>With control measures implemented | | | Responsible Officer | Timescale/ Review Frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Likelihood (Probability) [L] | Impact (Severity) [I] | Risk factor [IxL] | | Likelihood (Probability) [L] | Impact (Severity) [I] | Residual Risk Factor | | |
| 1 | Resident gets no heating when windows are open | Residents can shut the window for the heating to be re-established | 5 | 4 | 20 | No sensors to be placed in kitchens and bathrooms as residents are more likely to have these windows open. | 1 | 3 | 3 | Hildebrand | weekly |
| 2 | One or more sensors fail and turns off the heating | Resident to contact repairs call centre | 3 | 5 | 15 | Direct help line to be set up. This will reduce response time to resolve faults. Sensors to be tested prior to installation and during commissioning. | 1 | 3 | 3 | Hildebrand/Camden | on installation and then daily |
| 3 | Air draughts from faulty windows turn off heating | check with residents if any issues of draughts | 3 | 5 | 15 | Draft proofing or similar measure to be provided to reduce draughts. Air tests prior to installation to determine extend of issue | 1 | 5 | 5 | Camden | on installation |
| 4 | Residents remove sensors | Pre-selection of volunteers excludes families with children, as these are more likely to pull the sensors. Briefing residents on project objectives and interaction with sensors | 2 | 5 | 10 | Recording occurrences when sensors removed so data is not affected.Checking with residents for reasons why sensors removed | 1 | 1 | 1 | Camden | weekly |
| 5 | insufficient number of volunteers for sensors | compensation to volunteers in the form of gift vouchers | 2 | 5 | 10 | create a reseve list of back up volunteers | 1 | 5 | 5 | Camden | one off |
| 6 | insufficient attendance at survey /workshops | invitation letters and posters publicising the workshops to large number of residnets | 5 | 5 | 25 | incentives in form of vouchers and refreshments for partipating | 2 | 5 | 10 | Camden | one off |

# 10.References


[1] FP7 COSMOS Project Deliverable D7.4.1 - Smart heat and electricity management: Evaluation and recommendations (Year 1 Functionality).

[2] FP7 COSMOS Project Deliverable D2.2.2 - State of the Art Analysis and Requirements Definition (Updated).

[3] Introduction to the Architectural Reference Model for the Internet of Things - FP7 IoT-A Project - http://www.iot-a.eu

[4] FP7 COSMOS Project Deliverable D3.2.1 - End-to-end Security and Privacy: Software prototype.

[5] FP7 COSMOS Project Deliverable D4.2.1 - Information and Data Lifecycle Management.

[6] FP7 COSMOS Project Deliverable D5.1.1 - Decentralized and Autonomous Things Management.

[7] FP7 COSMOS Project Deliverable D6.1.1 - Reliable and Smart Network of Things.

[8] https://www.camden.gov.uk/ccm/navigation/council-and-democracy/camden-plan/;jsessionid=E78A4B91B876CD221AE4F4D24374665D

[9] http://nodered.org/

[10] FP7 COSMOS Project Deliverable D2.2.3 - State of the Art Analysis and Requirements Definition (Final).