Small or medium-scale focused research project (STREP)
FP7-ICT-2013-11

Water analytics and Intelligent Sensing for Demand Optimised Management

WISDOM

Project Duration: **2014.02.01 – 2017.01.31**
Grant Agreement number: **619795**
Collaborative Project

WP.5

D 5.4 | DAPP

Large Scale Replication Plan for Public Release (Final)

Submission Date:
12.05.2017
Re-submission Date:
12.05.2017
Due Date
31.01.2017

Status of deliverable:

DrA	WoD	Rel	**Del**	AppR

Nature of the document:

R	P	D	O

Dissemination Level:

PU	PP	RE	CO

Project Coordinator: Daniela BELZITI
Tel: +33 4 93 95 64 14
Fax: +33 4 93 95 67 33
E-mail: daniela.belziti@cstb.fr
Project website: www.wisdom-project.eu

DOCUMENT INFORMATION

Submission Date:	12.05.17
Version:	V1.5
Author(s):	Alberto Musetti (DAPP) Elenia Duce (DAPP), Keith A Ellis (INTEL), Tom Beach (CU), Sophie Straiton (DCWW), Julie Macain (ICL), Eric Pascual (CSTB)
Contributor(s):	Eugene Ryan (INTEL), Manuel Ramiro (ADV)
Reviewer(s):	External Adam De Benedictis (CCC), External Felice Alfieri (RINA), Daniela Belziti (CSTB), Mathew Preece (CCC), Julie Mc Cain (ICL)

Document History

Date	Version	Name[1]	Remark
14.01.2017	V0.1	Elenia Duce (DAPP), Aberto Musetti (DAPP)	First Draft
16.01.2017	V0.2	Tom Beach (CU), Eric Pascual (CSTB), Julie Mac cain (ICL), Sophie Straiton (DCWW)	Almost all sections complete
20.01.2017	V0.3	Sophie Straiton(DCWW), Lydia Trzcinski (CSTB), Anna Taliana(DCWW)	First Revision
24.01.2017	V0.4	Keith Ellis (INTEL), Sophie Straiton(DCWW)	Implemantation of missing sections
25.01.2017	V0.5	Tom Beach (CU)	Finalization of Deliverable
26.01.2017	V0.6	Adam De Benedictis (CCC)	Document (external) review
30.01.2017	V0.7	Daniela Belziti (CSTB)	Document review
30.01.2017	V0.8	Felice Alfieri (RINA)	Document (external) review
31.01.2017	V0.9	Mathew Preece (CCC)	Additional improvements and review
31.01.2017	V0.9	Julie MccCann (ICL)	Additional improvements and review
31.01.2017	V1	Tom Beach (CU)	Document ready to be submitted to EC
09.05.2017	V1.2	Keith Ellis (INTEL)	Additional improvements
10.05.2017	V1.3	Elenia Duce (DAPP)	Finalisation of Document
11.05.2017	V1.4	Keith Ellis (INTEL), Tom Beach (CU) and Lydia Trzcinski (CSTB)	Additional improvements
11.05.2017	V1.5	PSC	PSC review & final edits

[1] Main Contributor and Partner

COPYRIGHT

Table of Contents

LIST OF FIGURES

List of Tables

EXECUTIVE SUMMARY

This document presents the results from task 5.4 – Large Scale Replication Strategy. This deliverable firstly describes the WISDOM Integration Approach together with the overall replication strategy (Section 1), both of which are informed by the WISDOM business model, described in deliverable D5.6. The WISDOM business model envisages the operation of the WISDOM system as a service, enabling water utilities to outsource their data management, while providing them with value-adding water network services which act on their data.

This deliverable then considers each of the WISDOM components in turn with respect to scalability and/or replicability/adoption. In Section 2 the core components of the WISDOM architecture, which are essentially context/industry agnostic, are discussed in relation to scalability. Then a full replication and implementation plan is provided for each of the WISDOM services which are based on the analytics based research of the project, Section 3. Lessons learned by partners within the project are outlined in Section 4.

1. The Proposed Value of Work & Progress Against Objectives

The main value of the work rests in supporting, in part or in whole, the adoption and utilization of the project outcomes through the development of a Replication Plan, which aims to be a type of blue print that captures all sharable and pertinent project knowledge. The production and apt dissemination of that plan (this report) directly addresses the DoW objectives below, the primary premise being that third parties can understand and learn from the WISDOM experience.

Objective 1: The WISDOM Replication Plan will capture all pertinent project knowledge that supports the adoption and utilization of the developed solution by non-consortium members. The replication plan will include the methodology, approach, analysis of the WISDOM system, parts of the software development, parts of identified Business Models, and all practical and technical information for design, engineering, installing and commissioning learned from the pilot activities.

The deliverable documents the replication plan that supports the adoption and utilization of the WISDOM platform by non-consortium members. It explains how the WISDOM generic components, independent of a specific water network, can scale to support both larger water networks and a larger number of water networks. It also describes how the water network specific components of WISDOM can be successfully adopted for different water networks (Sections 2 and 3).

Objective 2: Guidelines will be defined to assure the dissemination of the Replication Plan to all relevant private and public organizations identified by the project partners over the course of the project.

This document presents our guidelines for dissemination of this content in Section 5.

It should be noted, that progress against objectives has been impacted with respect to what can be public shared at the time of publication. The reason being, the consortium is faced with the challenge of conflicting commitments that span three project deliverables, namely:- **D5.4** Large scale Replication Plan for Public Release (this document), **D5.6** WISDOM Business Model **& D6.2** Exploitation. There is an onus on the consortia to exploit the project results **(D6.2)** and to outline a likely business model that compliments commercial exploitation **(D5.6)** yet conversely there is a requirement to support large scale replication **(Objective 1 & 2).**

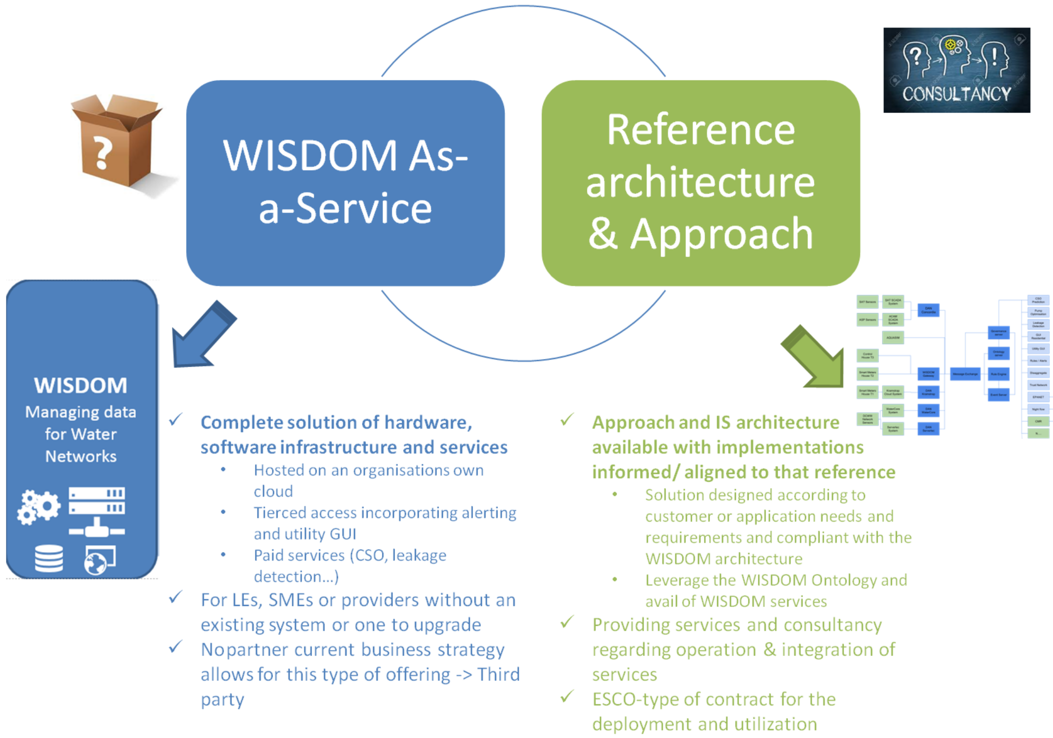

Figure 1 the two WISDOM exploitation routes

As per Figure 1, there is essentially two exploitation paths open to the consortium, that of a ***'commercial product offering'*** or that of a '***reference architecture and approach***'. There is an obvious tension here as the consortium cannot commercially exploit **(D6.2)** the WISDOM implementation if it has made it freely available for third party replication **(D5.4).** More specifically, given DCWW are conducting an internal assessment to understand if the platform and/or services, can be exploited, it is necessary to restrict D5.4 to identifying the methodology and technologies utilised and not the specific heuristics that drove the integration of those components / technologies.

In short, DCWW are assessing if the platform and/or services, can be exploited within their organisation. Failing that, if third a party DaaS licensing agreement is not likely, the path to open source the platform can be taken and an accompanying more detailed replication / implementation plan (this document) supplied. However, services will remain subject to partner's individual intellectual property rights given licensing is the primary option open to some partners.

The sections that follow reflect this challenge.

1. Introduction

The aim of the Replication Plan is to capture all pertinent project knowledge that supports the adoption and utilization of the WISDOM approach by non-consortium members, in line with the WISDOM business model that is presented in D5.6, but summarised here.

In the remainder of this document, the overview of the WISDOM integration approach and of the project's replication Strategy is provided in this Section. In Section 2, the scalability of the WISDOM system is addressed through an analysis of the WISDOM architecture and all related components. To better understand the WISDOM system, a technical guide that outlines the steps taken in creating the WISDOM core system components is also provided in Section 2. Section 3 includes practical guidelines and a detailed analysis of the technical specifications and of the implementation aspects for the replication of the WISDOM services. Section 4 documents the lessons learned during the pilots' implementation and at the ICT level. Finally, the Section 5 reports the guidelines for the dissemination of this replication plan.

The reminder of this section outlines the WISDOM replication strategy that is informed firstly by the WISDOM integration approach and secondly, the tensions we have found between open replication and fruitful exploitation of project results by WISDOM partners (as described in the executive summary).

1.1. The WISDOM Integration Approach

The WISDOM approach focuses on generating 'Analytics' based domain insights. To act on such insights or indeed to acquire the data on which such insights are based, one requires stakeholders and information systems to have appropriate access to accurate data in near real-time. As such, the ICT partners within the project had the task of connecting / integrating the domain assets of the pilot sites with service instantiations of the analytics based on the research completed in WP3.

In general, the connection has been a more onerous task than it might seem, as heterogeneous data formats, application integration times and batch processing times inhibited the free flow of accurate and timely information amongst systems. Additionally, as per Figure 2, there are many within the industry focused on connecting the assets of the physical world with services of the cyber world. This burgeoning array of potential options can make the task of design and development more taxing.

As shown in Figure 2, some of the many alliances and organisations target Machine-to-Machine (M2M) communications and according to the Global Standards Collaboration Machine-to-Machine Task Force, more than 140 organizations are involved in standardization worldwide.

Figure 2 IoT alliance overview (Source: March 2015 http://postscapes.com/internet-of-things-alliances-roundup)

Like for many of the organizations in Figure 2, integration i.e. technical and syntactic interoperability was a prime requirement, as was semantic interoperability. However, there were many aspects to also consider including; existing legacy systems, multi-ownership, cost sensitivities, security, privacy and reliability.

The WISDOM project focussed on addressing these design considerations by taking a System-of-Systems (SoS) approach that was cognisant of a brownfield scenario, in which there are many existing systems controlling assets at a district / network level. A SoS approach allows for direct actuation but more importantly it offers the ability to act as a recommender system whereby the existing management systems can remain in control. This is often more desirable for organisations that have vested interest in utilising existing systems or who want ultimate decision making and/or raw data to remain within their own area of control, which can often be on site.

SoS formed a conceptual approach, but the architecture for the WISDOM Information System (IS) followed a 3 tier structure that incorporates both cloud and edge processing, this is shown in Figure 3. The platform is cloud hosted, so typically are the services, while edge compute is largely on site, i.e. embedded within the water network or domestic properties and can run services locally. The ability to run services locally using edge compute can be favourable given real-time processing requirement or security/privacy concerns in sending data off site to a remote cloud server. While consistent with the SoS approach, the IS can act as a recommender system or as an end-to-end system controlling actuation. The core components of the implementation are discussed in section 2 while the domain specific service implementations of the algorithmic research conducted within the project are discussed in section 3. The approach is underpinned by the WISDOM ontology, also outlined in section 2, which addresses semantic interoperability a prime enabler/barrier to scalability.

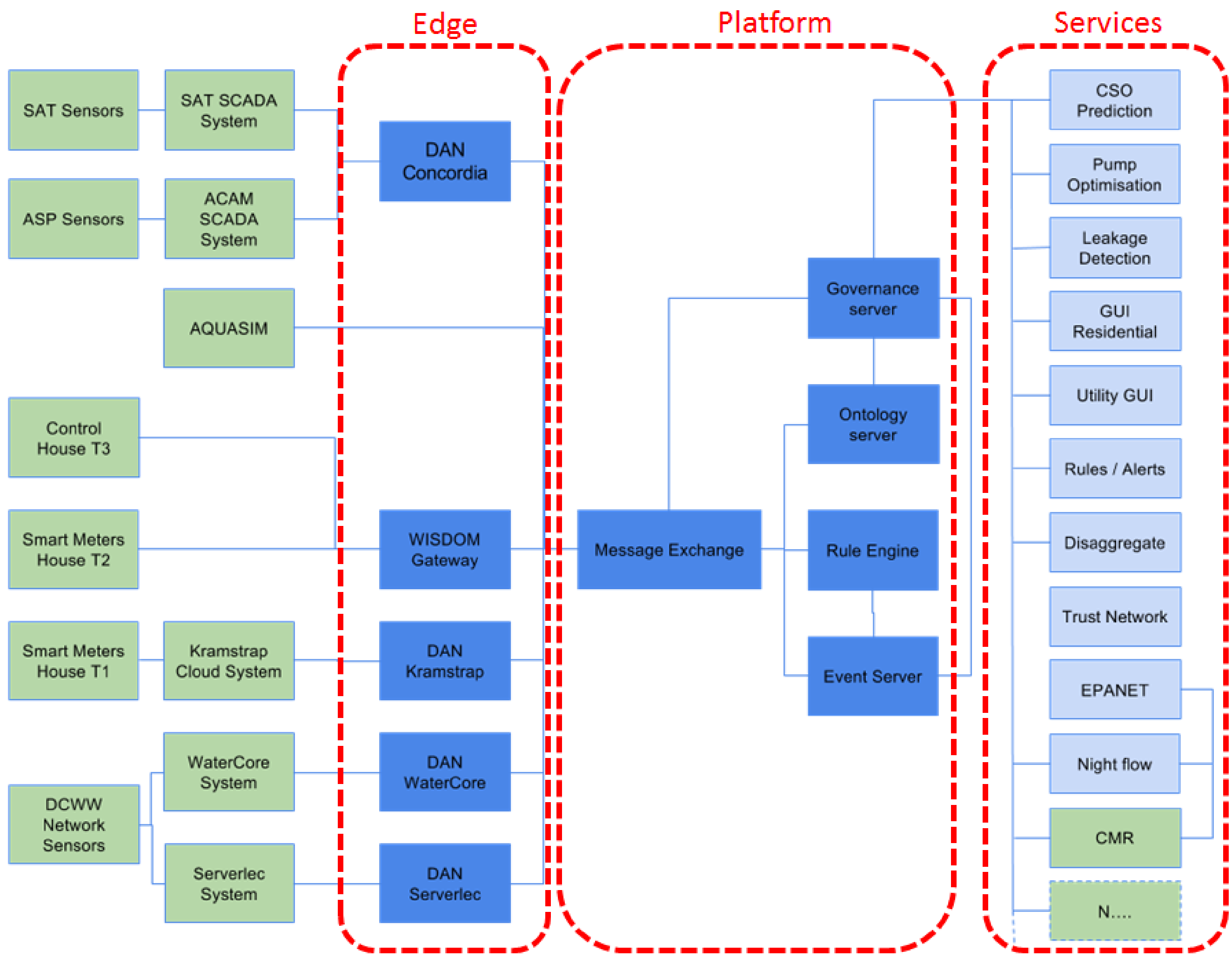

Figure 3 The WISDOM 3-tier system map

1.2.WISDOM Replication Strategy

The replication strategy has been built around the project's business model, but has been restricted based on the tensions found between exploitation and replication (as described previously).

The WISDOM business model can be summarised as offering water network management as a service. Within this business model water network utilities utilise WISDOM to manage their data, by connecting their existing sensing infrastructure to WISDOM via a DAN (Data Access Nodes) and the development of ontological representation of their water network. Once the data is stored and managed by WISDOM water network operators can then take advantage of a variety of WISDOM services utilising this data stored within the system.

Due to the tensions we have found between exploitation and replication our replication strategy has consisted of; (a) documenting how the WISDOM platform can scale, allowing it to support a higher quantity and larger sized water networks without the need for replicating core WISDOM components (Section 2) and (b) documenting how the WISDOM services can be ported to new water networks (Section 3).

This restricted replication strategy has been taken due to DCWW's possible desire to exploit the WISDOM platform internally, and publishing the WISDOM code and information on how to replicate the platform publically would harm their competitive advantage in using the platform.

In the future, if DCWW were to decide not to exploit the platform, the WISDOM replication strategy can be expanded to provide a full open-sourced version of the WISDOM platform. It should be noted, however, that there are no plans to open-source services, as all partners have well defined exploitation routes for these (see D6.2).

For the purposes of this replication strategy, as published in this document, the WISDOM components are divided into two categories:

- **Generic components** that acquire, process and store data, and are independent of a specific water network
- **Water network specific components** that require reconfiguration and re-deployment for each water network on which the WISDOM system is implemented.

The division of WISDOM components into these categories is shown in Table 1 below.

Table 1. WISDOM Component Categorisation

Generic Components	Network Specific Components
Message Exchange Server	WISDOM Ontology
Time Series Database (Event Server)	WISDOM Rule sets
Ontology Server	User Interfaces
Governance Server	CSO Prediction service
WISDOM Edge processing & analytics	Pumping Optimisation Service
WISDOM Rule Engine	
WISDOM API	
WISDOM Data Acquisition Node (DAN) and Gateway	

Due to their different natures, this document will consider these two categories differently. For components, independent of the water network upon which WISDOM is operating, their scalability will be examined to show how the WISDOM as a service concept can be scaled to support larger and a greater number of water networks. Secondly, for components that require reconfiguration/re-deployment onto each new water network their replicability will be examined.

It should be noted that two services are not considered for replication as part of this strategy; (a) water usage disaggregation and (b) leakage localisation. These services have not been considered as part of this replication strategy due to their low TLR level. Over the course of the project they were not fully integrated into the WISDOM platform thus not fully deployed in a real pilot (leakage localisation was tested in DCWW test facility, and water usage disaggregation utilised data from the pilots but was not exposed to consumers). This means further research and development work is required in order to assess their widespread applicability and thus replicability to a wider variety of water networks when deployed within the WISDOM platform.

2. Scalable approach/architecture for increased replication

Figure 4, illustrates the main components which have been chosen in developing the cloud hosted platform of the WISDOM Information System (IS). The important aspect to note here is that the WISDOM system has, in the main, utilised open source technologies to aid adoption/replication. This information can be used as a reference architecture to assist others in utilising these open source components in a similar way.

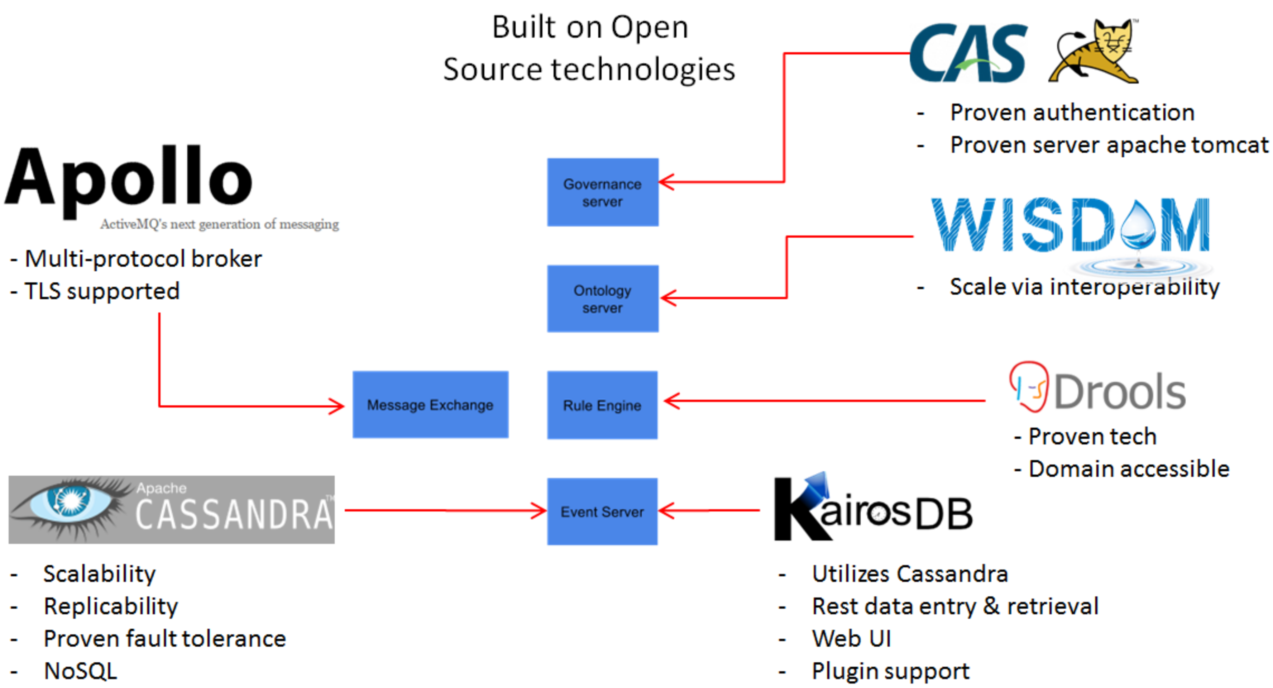

Figure 4 The open source technologies integrated within WISDOM

Our main goal is to promote the use of the WISDOM system as a service (as described in our business model). We primarily focus on considering the base technologies used in WISDOM regarding scalability, extensibility, security and reliability i.e. in the ability to deal with increased users, connections, devices etc. as they scale.

The sub-sections that follow recap the domain agnostic components of the WISDOM system, based on these open source technologies, and highlight the features that enable scalability. Firstly, the core components as they relate to the cloud and edge platform are introduced. Secondly, given the importance of the WISDOM ontology for interoperability and hence adoption, the ontology server implementation is discussed in terms of handling increased users/devices. Thirdly, the novel and generically applicable edge processing approach adopted in the WISDOM case to address leakage localization and detection is discussed. This is followed by an overview of the WISDOM rule engine and API.

2.1. Core System Components

Figure 5, provides an overview of the core system component architecture. Historical events are stored by the event logging service, while the WISDOM ontology service stores a snapshot of the current water network state. Data is acquired through the WISDOM DAN or Gateway (typically embedded within the water network or residential homes) and messages are routed via same to the Message exchange server and from here to the event and ontology services. The Governance server manages authentication.

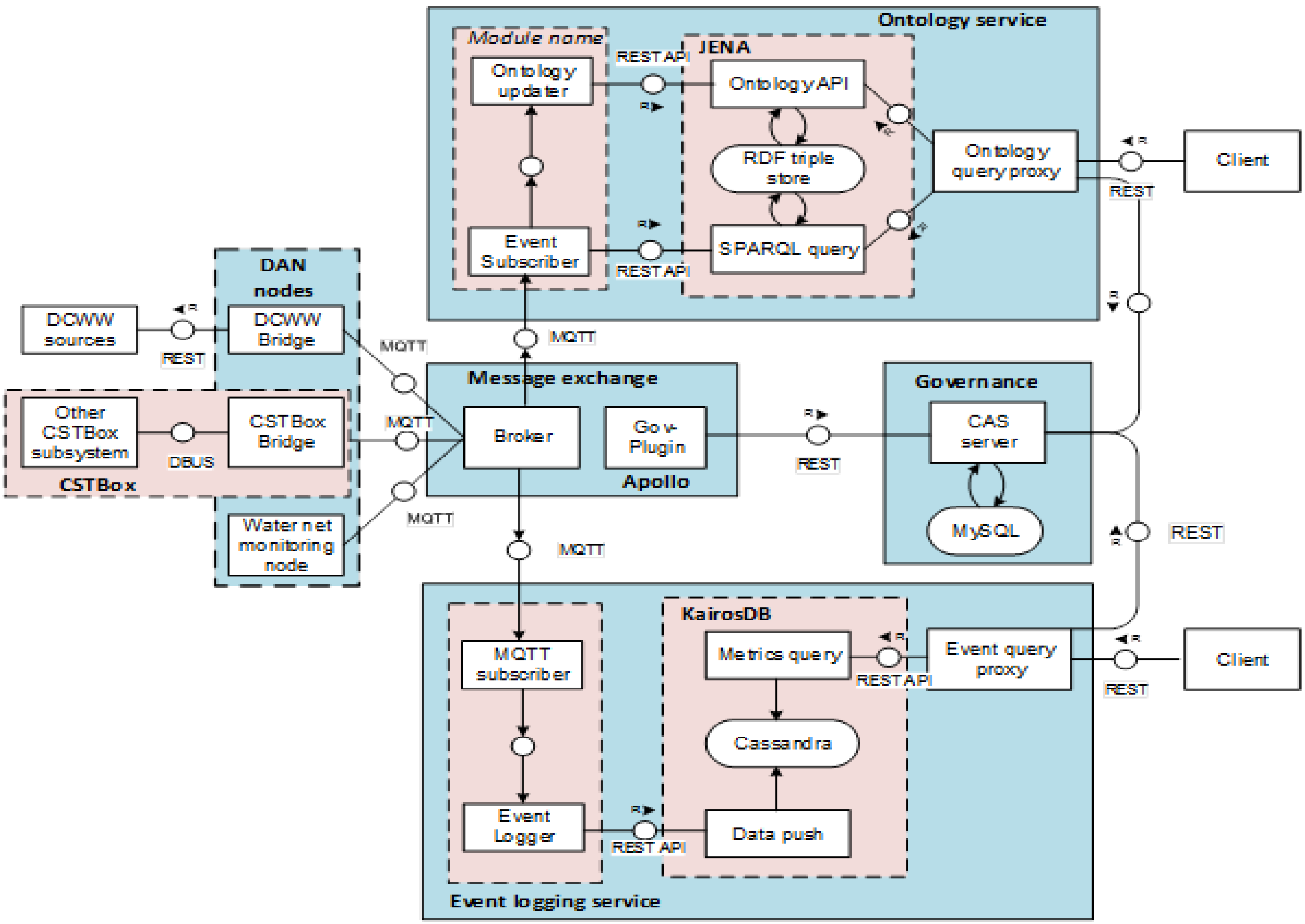

Figure 5: System components of the WISDOM secure data storage system.

In the paragraphs that follow each component is summarised and aspects that aid scalability are outlined.

2.2.Message Exchange Server

The message exchange server is based on an Apollo[2] broker instance. It utilises open source release version 1.7.1, which is customised through several configuration options. Apollo can act as a multi-protocol broker and supports STOMP, AMPQ and MQTT as underlying messaging protocols. Apollo additionally has a REST API for management and configuration.

The current DAN / Gateway nodes that act as clients to the broker make use of the MQTT protocol. The latter has been selected for the flexible aggregation capabilities based on topic names and consortium familiarity with the client side libraries.

Secure insertion of data is based on several features configured with Apollo:

- The use of SSL certificates for the encryption of data between the DAN nodes and the Apollo broker. Messages are secured over Transport Layer security (TLS);

- The mapping of user/password based authentication and authorisation of MQTT messages to the governance service.

The interaction with the governance service has been achieved by the implementation of a custom authentication plugin, which is invoked by Apollo to perform authentication with the governance service, by extracting the credentials from MQTT connection requests. Only clients that are successfully authenticated can connect to the broker and submit messages to it.

In terms of scalability, within Apollo scale by design is addressed and standard messaging protocols are supported out of the box. The MQTT protocol used in WISDOM and JSON payloads are open standards and are thus likely to lead to easy adoption. In terms of technical scalability Apollo's non-blocking, asynchronous architecture allows it to be fast and scale very well on multi-core systems using a minimal number of threads.

If further scalability is required, the option to spin up additional instances of the broker is a standard and supported option for scaling. The routing topology that can be implemented can enable highly-scalable systems. Routing can be simplistic or quite complex, and when designing a routing topology for a scalable, complex system it must be elegant in design. If kept clean and decoupled, components can throttle resource usage quite nicely with varying loads.

2.3.Time Series Database (Event Server)

The event storage system consists of a time-series data store for the storage of historic system events. The time-series database is based on the latest version of KairosDB. It has been configured to run on top of an Apache Cassandra cluster. Two additional components have been developed to allow the efficient integration with the message exchange service and its exposure to external clients:

[2] https://activemq.apache.org/apollo/

- The 'EventLogger' is written in Java and Scala and provides an efficient means to selectively subscribe to messages from the message exchange and load this into the time-series database.

- The 'event query proxy' implements a RESTful query API of the event storage system and exposes it to external clients or other system components. It is based on the Java spring framework. Received API calls are authenticated and authorised against the WISDOM governance service. Upon successful authentication, the event query proxy maps the queries to the time-series database and returns the query results to the requesting entity.

Time series databases on top of cloud infrastructures scale linearly with the number of nodes employed in the cluster and in a general sense this is how scalability of the event service is addressed within WISDOM. While specifically KariosDB in combination with Cassandra have shown good results in terms of out of the box scalability [1].

On a somewhat related note, Cassandra specifically addresses reliability through replication. Copies of the data are replicated across potentially several servers in a Cassandra cluster. Due to the possibility of servers being unavailable, delays in the network, and for performance reasons, NoSQL databases usually have a mechanism called 'eventual consistency' for guaranteeing that replicas are up to date. Cassandra uses its own version of eventual consistency called 'tuneable consistency' whereby it is possible to specify the behaviour of the replica system to a fine degree of control.

2.4.Ontology Server

The ontology server consists of a water value chain domain ontology common across pilot sites and a separate instantiation of this ontology to create a separate knowledge base per pilot site. In this way, the domain ontology could easily be instantiated for further pilot sites. The resulting knowledge base is stored as persistent RDF triples. A RESTful web service, with an API which provides GET, PUT, POST and DELETE functionality via SPARQL queries as well as a selection of convenience GET functions for the most common tasks identified.

More specifically this server consists of two components:

- The ontology web service which connects to other WISDOM components, such as the WISDOM message exchange, and provides the API by which other WISDOM components interact with the ontology. The component performs its tasks by querying and updating the underlying ontology data, provided by Apache JENA TDB.

- Apache JENA TDB is the second component of the ontology service and has been deployed to store and manage querying of ontology data, this is an off the shelf component that allows the manipulation of ontology data using SPARQL queries.

To allow the WISDOM system to scale managing both the data from either a larger number of water networks or larger water networks must be ensured. The ontology web service is primarily a CPU bound component that interprets events from the WISDOM message exchanger and provides an API by which other components access the WISDOM ontology. As this service is stateless, scaling of this service is trivial and can be achieved by duplicating the service on additional servers, with each servicing connecting to the same JENA TDB instance.

The JENA TDB service itself scales to store ontologies far bigger than those currently utilised by WISDOM. WISDOM's ontology is currently approximately 100K triples and JENA TDB has been tested up to 100M triples[3]. While the tests cited in [4] are extreme, they show that the size of WISDOM's ontology for our pilots is orders of magnitude less than the maximum that TDB has been tested with, showing its ability to scale to larger water networks. In multiple network scenarios requiring capacity beyond one TDB instance, it is likely the ontology server will involve separating different water networks on to different TDB instances as the segmented nature of control in the physical networks most likely will not warrant holistic connection, linking and management. In such a scenario, the ability to scale, as in dynamically across nodes, does not arise. However, it is conceivable that dynamic scaling may be required and in such a case, scalability best practice can be utilised.

2.5. Governance Server

The Governance server is based on the Jasig Central Authentication Service (CAS)[4]. It utilises the open source release version 4.0.3, which is customised through several configuration options. The WISDOM governance service forms a central role in the WISDOM approach to security but its operation is described here for completeness. The server provides two main functionalities: authentication and authorisation. Therefore, two distinct applications have been setup and deployed on an Apache Tomcat[5] application server. The token based approach of CAS is designed with scalability in mind as tokenization is stateless and reduces the overhead involved post initial authentication unlike session based approaches [2, 3]. CAS itself has modest computing requirements such that any modern enterprise class server hardware is going to be sufficient to handle tens of thousands of users in typical deployment scenarios. Through clustering, CAS can provide high availability and scalability scenarios. The clustering can be achieved by implementing a multi-node CAS deployment running on multiple VMs or physical hosts. This approach is attractive since it allows true zero down-time maintenance of the service at the cost of a marginal increase in deployment complexity.

2.6. Edge Processing – Data Acquisition and Actuation (DAN) Services and the WISDOM Gateway

A key element of the WISDOM approach is the utilisation of edge processing to provide the interface between the WISDOM system and sensors and actuators deployed on the water network or in consumer properties. Edge processing was used in this context for three reasons:

- Adopting a partially de-centralised architecture lends itself to a highly-scalable solution, giving higher potential for replication and scale-up.

[3] http://lists.w3.org/Archives/Public/public-sparql-dev/2008JulSep/0029.html
[4] Jasig home page http://jasig.github.io/cas/4.0.x/index.html
[5] Apache Tomcat http://tomcat.apache.org/

- Edge processing can also be utilised to perform computation on the incoming data, prior to transmission to the WISDOM cloud based system, further enhancing the scalability of the approach.

- Edge processing also allows asset owners to restrict what data is transmitted from their systems into the WISDOM cloud; to us this is especially important in the domestic context. While this does not aid the scalability of the system, it does provide additional reassurance to end users, thus increasing the take-up.

The primary component used to achieve this are WISDOM's Data Acquisition and Actuation (DAN) services and the WISDOM Gateway (see Figure 5), which are responsible for providing the interface between the WISDOM system and sensors/actuators deployed on the water network or in a consumer's property.

The DAN/Gateway node consists of two key components; a protocol adapter to convert the data from the native format used by the underlying sensing system into a format understandable by the WISDOM system and a message producer to insert the data into the WISDOM event pool. In terms of scalability the DAN and the WISDOM gateway are designed to be per system or residence implementations. As such, the DAN/Gateway are a trade-off that allows for on-site data services in direct response to real-time and/or security and privacy requirements. The 1:1 mapping that has been adopted so far has a limited impact on scalability of the approach but from our research, the approach is more likely to be adopted / replicated given multi-ownership and existing legacy systems. This physical WISDOM gateway itself uses the Intel Edison and the CSTBox framework to execute the DAN software as described above, to collect data from our in-home sensing systems.

Finally, it should be noted that for a DAN/Gateway to convert data from a water network's legacy systems (or in home monitoring systems) it must be modified so that it can understand the data formats being considered, this is described in more detail in Section 3.

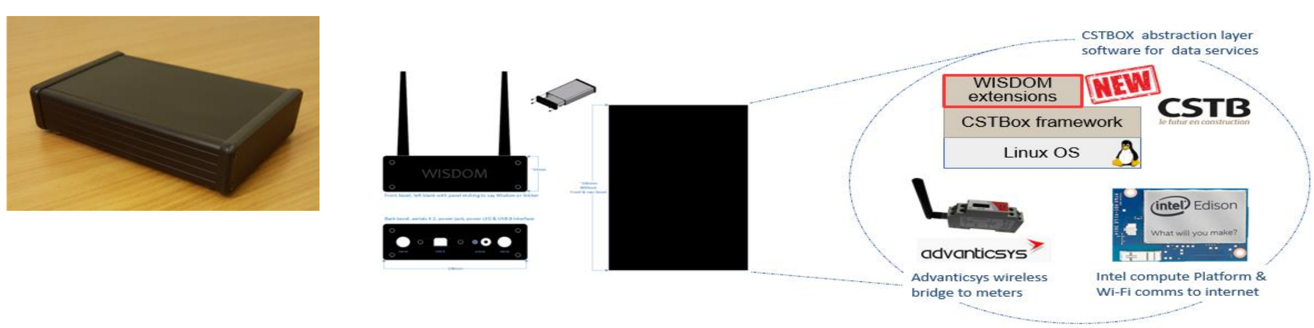

Figure 6 The WISDOM Gateway

2.7.WISDOM Rule Engine

The key aspect in our motivation for adopting a standardised rule engine is that it provides a readily updatable means of adding rules that relate to a specific business context. As such, a rules engine is a well understood way of allowing for extensibility and scalability of business intelligence.

The rules engine that is used in WISDOM is Drools which is provided as a cloud based service. There are various ways of specifying business rules in Drools, either via DRL (drools rule language) scripts specifying the rules and Java classes to carry out actions or via the integration of an Excel spreadsheet that refers to inputs and conditions, as well as actions to be taken. This enables business users (non-programmers) to specify and implement rules without having to program in a programming language. This approach allows domain experts such as mechanical engineers to dynamically enter rules in a domain specific language following a simple syntax using a commonly used application like 'Excel'. It is for the latter reason that Drools was chosen as it was felt this would aid in adoption and ease of replication.

In terms of scale, further instances of the Rule Engine on separate servers can be deployed as required. The publish-subscribe mechanism described in the Message Service section outlines how the consumption of messages/events by the Rule Engine is independent of the production of these events and independent of how many message brokers are provided. To further provide for scalability, the Rule Engine has been deployed by wrapping JBoss Drools libraries in a Play Framework application. The various plug-in classes used by the Rule Engine (for example, the Scala classes that interact with the Event Server or Email servers for alerts) are written in Play using Futures, a non-blocking technique known to aid throughput and enhance scalability.

2.8.WISDOM API

WISDOM exposes three external APIs for securely querying and inserting to the WISDOM IS.

- The **governance API** enforces the secure access across all system components of the WISDOM IS.
- The **data insertion API** handles insertion of system events and information into the WISDOM event storage and ontology service.
- The **data query API** provides access to historic information of the data store as well as information from the ontology.

The Governance system implements two communications; CAS-protocol[6] and REST-protocol[7]. In this way, the Governance handles both GUI applications, where users are redirected after authentication (the "classic" web page for user login), as well as in services (where a RESTful interface is needed). The remainder of the APIs adopt a purely RESTful approach. For more detail on the specific APIs see deliverables D2.3 and D2.4.

The architectural properties of REST address scalability. REST's client–server separation of concerns simplifies component implementation, reduces the complexity of connector semantics, improves the effectiveness of performance tuning, and increases the scalability of pure server components. Layered system constraints allow intermediaries—proxies, gateways, and firewalls—to be introduced at various points in the communication

[6] http://jasig.github.io/cas/4.0.x/protocol/CAS-Protocol.html
[7] http://jasig.github.io/cas/4.0.x/protocol/REST-Protocol.html

without changing the interfaces between components, thus allowing them to assist in communication translation or improve performance via large-scale, shared caching. REST enables intermediate processing by constraining messages to be self-descriptive: interaction is stateless between requests, standard methods and media types are used to indicate semantics and exchange information, and responses explicitly indicate cacheability [5].

2.9. Conclusion

This section has described how the WISDOM system can scale to support larger, or a greater number of water networks. This has been achieved through the scalability of each component within the overall system, which have each been developed in a such as way as to promote scalability. As such, all the components of WISDOM are fully scalability in one of two ways; (a) being able to scale to additional computing resources using a clustering model or (b) being able to scale to multiple instances utilising load balancing.

3. Replication in Action for WISDOM Services

This section explains the replication steps that are needed to replicate the WISDOM services in a new water network. It describes the modules that need to be adapted so the new network may start using the WISDOM system.

3.1. WISDOM Ontology

For a new water network to make best usage of the WISDOM system an instance of the WISDOM ontology must be created for that network, this follows a typical object-oriented approach, where classes represent descriptions of types of objects, and these are instantiated for actual objects. For example, a pipe class may describe that pipes have diameters and locations, whereas pipe_01 has a specific diameter and location. This section will briefly summarise how the WISDOM ontology could be instantiated for a given water network.

The process of instantiating the domain ontology into a graph database will depend on the client's existing data sources and their structures. If the new water network currently uses either EPANET modelling or GIS based software, then automatic instantiation is possible. WISDOM has provided automatic instantiation script described in D2.2 which may be used to instantiate the ontology automatically from an existing GIS schema and telemetry system, which many modern utilities will have in more economically developed countries.

This script does require some manual configuration of mappings to the client's specific data structures, after which it can be executed from command line. This script will then instantiate the core technical and cyber concepts of the system. Manual checking should be undertaken to ensure the quality of the data in the new ontology. Depending on the use cases of the target system, social data and/or domestic data may also be required, which would require manual elicitation through interviews and surveys. Following instantiation of the ontology into a graph database, this can be merged with the domain ontology (using an installation script) and loaded into the triple store (by simply placing the file into a specified directory). More detail regarding this script and the nature of graph databases can be found in D2.2.

There are several circumstances, however, where the automatic process may not be sufficient. This includes situations where a water network has no GIS or EPANET data to build the ontology from. In this situation, there is no choice but to build the ontology manually. Additionally, there may be circumstances where the core WISDOM ontology may need modification to work with specific requirements of the new water network. In this case a requirement engineering procedure should be followed. This is described in Annex A.

3.2. Data Acquisition and Actuation (DAN) Services

WISDOM has currently developed several DANs to allow it to connect to the legacy systems deployed in our pilots. Currently DANs exist for DCWW's telemetry system (PRISM) and SCADA systems. A key element of the replicability of the WISDOM system is the development of new DANs as, for each new legacy water network sensor/telemetry system that must be connected to WISDOM a new DAN must be developed.

Within the WISDOM project a framework for DAN servers has already been developed, to port this framework to a new water network sensor/telemetry system the following steps need to be followed:

- Examine the API/data format and frequency by which the telemetry/sensor system is to provide data to WISDOM.
- Develop a parser for the incoming data stream.
- Develop a translation between the parsed incoming data stream and the API provided for the WISDOM messages exchange.

Once these steps are completed the new DAN can be deployed and configured with the water network telemetry/sensor system.

This process of developing new DANs is incremental as, eventually, the WISDOM system will contain a sufficient library of DANs enabling it to connect to multiple different types of telemetry/sensor systems.

3.3.WISDOM Gateway

As described previously, the WISDOM Gateway is responsible for connecting sensors to monitor household water/energy usage to the WISDOM cloud based system. The WISDOM gateway incorporates the Intel Edison (utilized as the compute platform, with built in Wi-Fi connection to the internet), the Advanticsys DM108 wireless bridge for connecting to water and energy meters, a Linux OS and the CSTBox framework. Its main role is to manage and translate the data from in-home sensing to a format readable in CSTBox environment. This device is deployable individually in homes that wish to monitor their water usage.

Both the operating system and the gateway software are open source and platform agnostic and therefore easily replicable. The key issue in replicating the WISDOM gateway to a new set of consumer properties is adapting it to a new set of field devices (i.e sensors) that may use a different communications medium (i.e. zWave instead of wireless Modbus). If these modifications are required then it could easily be achieved by firstly connecting new receiving devices to the Intel Edison board that is used as the compute platform within the gateway and, secondly, by reconfiguring the software to receive data from these devices. The ease of this adaption is possible for two reasons:

- The Intel Edison provides USB connectivity, allowing the easy connection of multiple wireless receivers.

- CSTBox, on which the gateway is based, already supports a variety of communications technology for receiving of sensor data.

3.4. WISDOM Rules

The rules that have been implemented so far in the WISDOM rule engine are specific to the Welsh and Italian pilots. The rule engine mechanism is totally generic and context agnostic therefore replication of the mechanism is straight forward while replication of the rules will be bespoke to the network implementation.

To achieve the replication of the WISDOM rule engine to a new network, creation of new rules (or adaption of existing rules) must take place. The WISDOM rule engine provides convenient ways to do this. Each ruleset can be coded in Excel spreadsheets or directly in the DROOLS rule language (DRL) and then loaded into the Rule Engine. This deployment is simple and, once created, rules can be deployed by simply dropping the Excel or DRL files into a directory.

3.5. User Interfaces

The WISDOM user interfaces that have been developed provide access and render data from the WISDOM message exchange, ontology service and time series database. As these components are all generic, the replication of the WISDOM user interfaces in new water networks is largely a task of localisation. More specifically this will consist of the following steps;

- branding of the user interface to a specific water utility,
- localisation of the user interface into any required languages.

3.6. CSO Prediction Service

To enable the replication of CSO level prediction service on a new water network, firstly, a full sensing infrastructure on the waste water network under consideration is required. However, given the regulatory and operational requirements of waste networks, most network operators will already have CSO levels and flows closely monitored across the network.

Our approach utilises a model construction algorithm to generate the data driven model required to predict CSO levels in each network, thus the following steps can be applied to run this model construction algorithm on a new network:

1) **Preparing historical monitoring data:** retrieve the maximum amount of historical data for all sensors installed on the network and organise this data by the type of reading i.e., CSO Levels, CSO percentages, weather data, flow data, and any other field data. The CSO model generator uses CSV files for its input to avoid data translations issues.
2) **Configuration:** The model construction algorithm must be configured by specifying a list and the ranges of field parameters and the required data resolution.
3) **Execute the CSO model construction algorithm:** This must be run using the historical data and configuration defined previously. This will create a new data driven CSO prediction model for the new water network.

4) **Creation of new instance of the CSO prediction service:** A new instance of the CSO prediction service, using this new model, must then be created by simply duplicating the existing service (developed within the project) using the new model file.

Following these four simple steps a new CSO prediction service for a new water network can be created to allow prediction of CSO levels on this network.

3.7.Pumping Optimisation Service

The replication of the network operational optimisation has two key pre-requisites:

- The existence of a site-specific network hydraulic model.
- An existing monitoring system to monitor key network entities including service reservoir levels and pump running status. These systems are usually deployed by most network operators for safety and operational reasons.

Once these pre-requisites are met all that remains to be done is the configuration of the optimisation module. To configure the optimisation module, the following parameters must be provided, using a configuration file (which can be generated via a user-interface):

1) pump identities and their characteristics (variable or fixed speed),
2) valve characteristics (active or non-active),
3) energy tariffs,
4) reservoir identities and operational levels,
5) minimum head requirements

With these parameters specified, the optimisation model can then operate on the given network. In its operational phase, real-time reservoir level, pump and valve statuses together with the time epoch will be automatically measured from the field and received via the WISDOM monitoring and the optimisation model can provide continuous updating of operating strategies for the network being considered.

3.8.Conclusion

This section has described the replication steps that are needed to retarget the WISDOM services (as developed within the project) onto a new water network. As each of these services, as well as the water networks they are targeting, have their own unique characteristics the replication steps for each service are different. This is an important point to note in the context of the WISDOM business model (described in D5.6) as the steps that must be carried out to make a service work on a new network are critical in understanding how best to exploit each service.

4. Lessons Learned

There have been several valuable lessons learned during the WISDOM project that can be utilised to either aid the replication of WISDOM by enabling smoother deployment on new water networks or provide more general advice to aid the wider water domain in the adoption ICT technologies.

4.1. Lessons Learned in SMART Meter Deployment

This section outlines some of the lessons learned in our Welsh pilot, in relation to the smart meter deployments that have been carried out. This was an especially challenging deployment as it is the first of its kind to take place in Wales, thus is considered here in detail.

4.2.Communications Protocols

One of the issues encountered during the WISDOM and project, and something that must be considered if the pilot deployments were to be replicated, would be careful selection of the communications protocol for smart meters.

When deciding on which SMART meter should be deployed, one of the first key questions to discuss with the manufacturer is the communication protocols the meter uses. This is generally radio protocols like - PLAN, LON/OSGP, Meters&More, PRIME and G, less so GSM/GPRS, WiFi or 802.15.4 etc. and can use either a licensed or unlicensed frequency.

In the UK, if the meter is using a licensed frequency, a license for this needs to be granted by OFCOM[8]. We have established that this should be done as early as possible within the metering procurement process. This is because the ability for OFCOM to grant the license which is required by the manufacturer is not guaranteed and communication between OFCOM and the metering manufacturer takes time. When applying for a licence, the metering manufacturer will specify whether a private or shared licence is required to comply with data security regulations and at what frequency range the meters can communicate over.

However, the ability for OFCOM to grant the licence that meets the manufacturer' specification will be dependent on the geographical area you are placing meters, as radio traffic, current signals, signal strength and range all come into play. It may be the case that you will not be granted a private license, but only a shared licence in the area you propose in which case your metering solution may become unusable and an alternative would have to be sought.

A shared licence may be unsuitable as data transmission may have to be at certain times of the day and at a different range to avoid communication confliction with the shared licensee and the original licensee would have priority. If a private licence was granted, but at a different frequency range to what was specified, in our

[8] OFCOM is the communications regulator in the UK

experience it will take the manufacturer a couple of months to develop a meter that communicates at a different frequency.

Also, when applying for a licence, you will need to ensure that you have successfully surveyed the site which is being metered, as the placement of infrastructure for the communication of data (i.e. base stations) will need to be considered to ensure that they are not obstructing other equipment from other providers. This will have a large impact on being granted a licensed frequency as OFCOM will need to know specifics before finalizing any license agreement, any uncertainty will delay the process. The process of obtaining a licence agreement may take several iterations and may not result in the desired outcome.

Using an unlicensed frequency allows you to get around the problems associated with obtaining a licence but needs to adhere to wireless communication laws. This generally means less range and more infrastructures required to cover a large area compared to licenced radio.

4.3. Data Transmission Vs Battery Life

Currently most domestic meters are read on six monthly cycles either by a meter reader, or by "drive-by" technology. However SMART meters allow you to do this on a more frequent basis, but this data needs to be transmitted over some wireless communications medium.

Therefore, when considering the higher reading rate and transmission rate you need to consider the battery life of the meters, as using too many transmissions would deplete the meter's battery more quickly and it would be costly to constantly replace the battery. A balance needs to be considered between battery life and data readings to achieve the needs of both the Water Company and the requirements of consumers. Therefore, when replicating this in the future, the needs of the company and consumers in terms of what data are required and its granularity should be considered at the start of scoping a solution.

Additionally, and, more generally, energy consumption versus data transmission needs to be considered in any wireless network application, as does the employment of edge processing techniques to reduce range of transmission (hence power consumption) while not losing valuable information.

4.4. Data Protection - Security Vs Openness

As SMART meters allow transmission of data over a fixed network, two considerations need to be accounted for: data protection and security.

Most SMART meters come with encrypted protocols to ensure data security, thus securing the personal data from the meters. Data protection comes into play as the data is that of the customer / end user of the water and data controllers need to ensure that data protection criteria are met. Through this project we have learned that data protection is an important factor that must be considered with SMART meters. Dependent on the setup and configuration of the meters, the data that is being transmitted can be considered personally identifiable. Additionally, if you use a third party to host / store the data you need to consider the location of this data centre (especially if it is within or outside the EU) and ensure all legislation in the data protection act is adhered to. For example, data protection rules in the UK may not map to those in other countries and so legal agreements need to be made with the host / vendor during the procurement process. In the process of conducting our deployment we looked at case studies, such as those carried out by Thames Water, to determine if any lessons could be learned from them. Finally, outside legal advice on customer data protection may be required, i.e. on the suitability of the existing contract terms for supply, management and usage of data from smart meters. All this process is timely and costly, and must be considered when managing deployments of this nature.

The juxtaposition between this need for security and privacy and the need to innovate and add valued add serves is faced with the tension to be both open and private. This is one of the prime reason for the WISDOM architecture incorporating an edge processing layer to offer services locally, or to process data locally allowing for non-raw data being shared if more agreeable by the stakeholders involved. Additionally, one can incorporate clustering anonymisation techniques that give enough granularity to be of value to third parties without allowing for individual identification. This has been done in Amsterdam in the framework of a Smart City project for example with respect to energy data[9].

4.5. Metering Standards

When looking at the SMART meters there are several standards which need to be adhered to, including demonstrating compliance with the Water Supply (Water Fittings) Regulations or Scottish Byelaws, measurement accuracy, IP ratings and installation fittings. You may need to use multiple vendors and models depending on requirements such as size and material of the pipe, flow rate and pressure within the pipe and fitting requirements. To allow multiple meter makes and models to operate on the same water network, you may need to consider utilising the OMS (Open Metering Standard) to ensure that all meters communicate in the same way.

4.6. Infrastructure / Equipment

When using SMART meters, to gain the most benefit from them you need to look at fixed network applications. To do this, you need to consider that most of these outlets need access to power supplies rather than batteries to ensure constant transmission. Therefore, you need to consider power, topology, line of site, and site configuration (urban, rural) as these elements play into the successful integration of your systems. Any one of these could impact your deployment, for example you may not have the range in a rural environment to reach all outlying properties, therefore may have to deploy drive by etc. and decrease the read frequency. You may also be required to obtain permission from local councils to install equipment on lampposts or buildings which takes time and needs to be factored into installation timelines.

[9] https://www.jaarcongresecp.nl/uploads/bestanden/2013/bob_mantel.pdf

4.7. Lessons Learned in Choosing Wireless Networking Technologies

Motivated by experience in deploying the WISDOM gateway, which communicated with the WISDOM system using the local internet gateway, it was determined that this is often not ideal for the following reasons:

- Absence of an internet connection is more frequent than envisioned i.e. in some groups of consumers (i.e. the elderly) internet adoption is far lower than average.
- People are often not willing to share their private network (and Wifi credentials) with external parties.
- People sometimes shut down their Internet gateway when not using their connection.

Alternatives should thus be considered. Motivated by this, a study was undertaken to understand the best of the modern wireless networking protocols to use to communicate data from residences and water networks.

Particularly Low Power Wide Area (LPWA) communication technologies were considered as an alternative solution to GPS technologies. LPWA systems have recently emerged as a viable alternative to cellular and mesh networks for providing cost effective IoT connectivity in cities and beyond. With several manufacturers and consortia now announcing national rollouts using different LPWA technologies, IoT solution and connectivity providers are now faced with the dilemma of what technology may be applicable in different environments or scenarios. Unfortunately, there was little empirical evidence that provides a good understanding of the performance trade-offs of these solutions applied to different application requirements and deployment contexts. For example, water networks run through a diversity of environments and these affect the communications. In rural areas vegetation or people here tend to be the main obstructions. The delivery of data is delay-tolerant, that is the exploitation of the data is typically non-time critical. On the other hand, the built environment brings additional challenges, it remains a delay-tolerant class of application but the penetration of radio through built spaces can greatly impact the design choice for IoT. Figure 7 represents the equipment we used to better understand these conditions. From this we see that LoRa, XBee 868 and nWave transceivers were utilized and all were tested in both a built up and more rural conditions.

Figure 7 hardware approach for testing different LPWA radio in the field

Further testing was carried out in man holes representing underground communication which is perhaps the most hostile environment that such technologies could be used in (bar submerged completely). Testing the system in the environments that such technologies are deployed ensures that they will be replicable to other such environments e.g. a city Like London will behave like Cardiff, a rural Welsh countryside will behave like La Spezia coast lines etc. Overall LoRa showed that it was the most reliable candidate.

The choice of networking (LoRa) and an edge based computational approach (as utilised in WISDOM) ensures that the system can be retrofitted to water networks with minimal structural change (except access to the network components). That is, wiring etc. is not required as the nodes are battery powered and communications are wireless. In the residential context, the use of LoRa removes the dependence on WiFi being present within the property. These factors increased the replicability of the system and again is essentially context agnostic.

4.8. Lessons Learned in Piloting Research in Water Networks

One of the key lessons learned in this project by the research partners is the difficulty in piloting water research, compared to other fields.

During the project, it was found that, compared with other City infrastructures, water industries are much more conservative. This is because water is of core importance to a civil society and distribution down-times are not tolerated lightly. It is for this reason that water industries are not as fast as, say, some of the energy suppliers, in the up-take of leading edge technology. Many distribution networks are of the form that was laid down in the 1960s or even earlier. Yet water networks and the distribution of water and waste has a considerable amount of scope to become more efficient and resilient with the addition of modern technologies; especially those that permit remote online sensing and control.

In parallel, sensing and control technologies are becoming more mature and are being used in anger in other fields; from transport to precision agriculture. Yet with this maturity there is a reluctance from water companies to take on board this technology as it is not tested in their domain. That is, sensing and control technologies that use modern telemetry and computer networks has not been tested on real water networks and on pipes both over and underground with all the challenges those environments bring. In a sense, they are correct. Such technologies, which have great potential to provide step-changes in the sustainability of water resources, are relatively new and one cannot imagine stopping water distribution just so that experimenters can test equipment.

Water companies are reluctant to use modern technology because it has not been tested on real water networks, and at the same time researchers cannot test their technologies on real water networks because water companies cannot risk untested technology being placed on the live water network.

The only reasonable way to progress this situation is to provide funding for large live water test-bed construction. They need to mimic the exact behaviour of a live water network to test and provide trust to the water companies that the new technologies with not disrupt the operation of their assets. They need to be large. Currently many water companies (such as Welsh Water in the WISDOM project) have reasonably sized test water rigs. These are primarily designed to help train staff. These and University based water rigs (of similar size) are used by many researchers building new technologies and anomaly detection algorithms etc. to supplement the simulations (e.g. EPAnet) produced to evaluate this new research. They can evaluate static and non-scale aspects of the subject. However, there are several aspects they cannot test. These aspects require that the sheer size of the network is that of numbers of kilometres to mimic the vastness and scale of a real system. One cannot test data communication on a test rig for example (many are indoors too). A multi-kilometre test facility will enable new communications technologies to be tested in realistic (city and nonurban) environments and this in turn will ultimately provide the scale and dimensional guarantees required by water companies to invest in water technology research earlier. Further, the topological structure of this facility should reflect the topologies seen in urban and country environments and this structure will better test leakage localisation and control algorithms and technologies more realistically and at the same time provide guarantees regarding the system density requirements.

Finally, a funding programme that supports the build of such facilities should take the different terrains of Europe into account; from lush green countryside to dry desert like soil to dense cities etc. This is because such systems will behave differently in different terrains. For example, wireless radio behaves very differently in dry soil compared with dense wet soil and pipe vibrations likewise pertain to both pipe and surroundings too.

4.9. Lessons Learned in Deploying Sensors in a Water Network in Italy

One of the WISDOM partners, as part of their deployments, was deploying for the first time a sensing system in their water network. During this deployment, they faced several problems:

- **Unclear Ordering Processes:** When the work of ordering began for sensors and transmitting devices it was very quickly found that not only do you need to order the devices themselves but also, for every device that you install, you must order a series of components that are also required. This meant that over 200 individual product orders had to be placed. To help with this the services of an external consultant had to be employed to manage this process. This is sub-optimal as it means you are dependent on someone external to your organization for critical tasks.

- **Electricity Connection Issues:** Problems were also found in getting electrical power supplied to the sensor sites. It was found that laws regulate the time in which the electrical company must activate new connections, but the time is counted in working weeks and not in hours. It was found to be very difficult to make the electrical company to respect these time limits. Leading to delays in getting electrical installation beyond when the estimated time to completion of the work is.

- **Sensor Installation:** Installation of the sensors was also found to be challenging as permission had to be granted by the local authority before interrupting the service – even at night. It was also found that the local authority does not respond in a timely manner to these requests. Challenges were also encountered in installing in pipes along a road: you must ask permission to dig the ground, permission to manage road traffic while you are digging, and permission to install the cabinet in which the devices for the data collection and transmission will be installed. For these requests, paperwork had to be prepared and sometimes it took weeks for the local authority to grant approval. This was further complicated because, in some locations, permission to had sought from two authorities. For some local authorities, they even operated a system where you asked for permission – waited 60 days – and you could then continue if you had no response.

- **Post Installation Troubleshooting:** Even after installing the sensors problems were found. Several of the sensors arrived from the manufacturer faulty. This meant, once they were installed, we had to arrange for their technician to visit and then we had to re-install a new sensor (facing all the problems described previously again). To this date, one faulty flow meter (installed in September) is still awaiting replacement by the manufacturer.

4.10. Lessons Learned in ICT Development

As outlined previously, ICT partners within the project had the task of connecting / integrating the domain assets of the pilot sites with service instantiations of the analytics based research completed in WP3. It was additionally highlighted that this process can be more onerous that one might expect, with some 140 organizations involved in standardization efforts to connect physicals assets with cyber based services. Additionally, it was outlined that any approach needed to consider technical, syntactic, semantic interoperability, existing legacy systems, multi-ownership, cost sensitivities, security, privacy and reliability. The following are the high-level lessons learned:

1. Taking a System-of-Systems (SoS) approach that is cognisant of a brownfield scenario, in which there are many existing systems controlling assets at district / network level is a pragmatic approach which appears to be more acceptable to domain stakeholders. The primary reason for this being they can access services of interest in a tailored way while maintaining control of assets i.e. system acts as a recommender

2. 'All models are wrong but some are useful' George E Box. Similarly, all architectures are wrong but some are useful, do not procrastinate, instantiate and get testing/using early.

3. Consider edge processing in your architecture, challenge cloud centric or indeed centralised models. Little data or rather Goldilocks levels of data can be more powerful than fixating on scaled volumes.

4. Obvious, but true, through focusing on well-defined interfaces e.g. APIs, one can decouple system/platform development from the task of services and make the posited offering more extensible while aiding individual team development.

5. Use current best practises (but see no.2), do not reinvent the wheel e.g. TLS for data in transit, existing projects like KairosDB for time series data etc.

6. Visualise the data. Often being able to visualise data e.g. by building a GUI is the only way to get to real requirements, as often a stakeholder will not be able to articulate what is required but given a visualisation can quickly say what it is not.

5. Future dissemination of the WISDOM replication plan

To maximise the replicability of WISDOM, an additional "Replication Guide" will be prepared. The content of this deliverable will be used as a starting point but the content will be focused on domain users that wish to replication or re-use the work done in the WISDOM project. This report will contain the following content:

1. A description of the capabilities of the WISDOM System and a presentation of the WISDOM business model *(taken from this deliverable and D5.6).*

2. A description of the components that have been utilised including links to their source code downloads (where available), contact details for the partner that developed them (so help and advice can be requested) and any dissemination materials about these components (including publications etc…). *(Taken from this Deliverable)*.

3. A (public) summary of the WISDOM exploitation paths in case water networks are interested to exploit components *(taken from D6.2).*

4. Information on the WISDOM ontology – and links to the SemanticWater website (where the Ontology can be found). This will also include a guide on how a water network can start implementing semantic modelling of their operations *(from D5.5).*

The report itself will be disseminated in three ways; (a) via the WISDOM website, (b) to our Special Interest Group and (c) to the list of industry contacts that we have gathered during the project.

5.1.Open Source Replication of WISDOM Platform

In the future, should DCWW decide not to exploit the platform, and no further direct exploitation options are available then the process of open sourcing WISDOM will begin. This will lead to the creation of an open source repository and the addition of extra information to the replication plan.

In total the following steps will be taken:

1. Harden the code and navigate any internal open source processes within the partners holding IP over the code.

2. Establish a team / community to support the 'release'. The will be the individuals responsible for providing information support and contributing initial code to the release in order to foster the start of a "open source community around WISDOM.

3. Ensure licensing and obligations are met, seperate code into appropriate modules or remove any bundled third party libraries to ensure compliance.

4. Prepare documentation to support release and add it to the WISDOM replication guide.

5. Assess the environment and planning actions to increase the potential for scaling-up success, e.g. target & disseminate to organizations with the capacity to implement the offering.

6. Finalizing the scaling-up strategy and identify next steps.

7. Make project available in github

8. Go-live with release

CONCLUSION

This deliverable has described the replicability of the WISDOM system in relation to the WISDOM business model that has been defined in D5.6.

Firstly, this document has described how the WISDOM components that do not require replication (because they are domain independent, and reside in the WISDOM cloud based system), can scale to support both larger water networks and a larger number of water networks.

Secondly, it was described how the WISDOM components that do require replication to be used in different water networks can be successfully replicated to these new networks to increase their adoption in new settings.

This work is all related to the WISDOM business model, which envisions that network utilise WISDOM to manage their data, by connecting their existing sensing infrastructure to the WISDOM system via a DAN (Data Access Nodes) and the development of ontological representation of their water network. Once the data is stored and managed by WISDOM, water network operators can then take advantage of a variety of WISDOM services utilising this data stored within the system.

In this context, we envisage this document being utilised by operators of the WISDOM system (licensed by the consortium) to assist in operating and scaling the WISDOM core platform to support the addition of new customers in the form of water network provides and, additionally, replicating the services developed within the project to enable their utilisation on new water networks.

Finally, this document has presented a series of lessons learned by the consortium in conducting the project, covering areas such as smart meter roll out, general issues faced in piloting research outputs in water networks, and selection of appropriate communications technology. These lessons learned require wider dissemination to a wider group of stakeholders. Our strategy for this will be to provide a specific replication report (the contents of which are described previously) that will be disseminated via our website, our current list of contacts, and the WISDOM newsletter.

REFERENCES

[1] T. Goldschmidt, A. Jansen, H. Koziolek, J. Doppelhamer, and H. PeiBreivold, “Scalability and robustness of time-series databases for cloudnative monitoring of industrial processes,” in Proceedings 7th IEEE Int. Conf. on Cloud Computing (IEEE CLOUD 2014) Industry Track. IEEE, July 2014.

[2] https://apereo.github.io/cas/4.2.x/planning/Architecture.html

[3] https://stormpath.com/blog/token-authentication-scalable-user-mgmt

[4] http://seaborne.blogspot.co.uk/2008/06/tdb-loading-uniprot.html

[5] https://en.wikipedia.org/wiki/Representational_state_transfer

ANNEX A – WISDOM ONTOLOGY ENGINEERING

If modification of the domain ontology is required, the first step is to perform a comprehensive requirement engineering process. Performed correctly, this thoroughly formalises the expectations of the client, and subsequently formalises the technical specification of the semantic web software and knowledge model within the broader system.

From the outputs of the requirements process, the reuse pathway of the WISDOM ontology becomes clearer. Specifically, the use cases, scope, and modelling decisions required by the target system can be compared to those for the WISDOM system and semantic technologies. This could then lead to additional competency questions for the ontology required by the target system if necessary. An ontology curation process is then recommended which follows best practice in reusing the WISDOM ontology and extending it where necessary. It is most likely that the ontology engineering work would not be completed in-house by a utility company, but by a technology specialist or IoT solutions provider contracted by the utility company, with the prerequisite knowledge. The next subsections discuss the requirements engineering process and ontology development stages in more depth.

System and Semantic Requirements

The suggested requirements engineering approach consists of 3 stages: scenario specification, software requirement specification, and competency question setting. The first recommended task is to gather knowledge about the domain, target systems, and intended value proposition of the overall software solution. Following this, formal modelling should be conducted of the business processes involved in the target system, and scenarios for the use of ICT within these should be developed.

Next, an analysis and design process should be used to produce software requirements for the overall software solution, through use case specifications and sequence diagrams. These requirements should then be iterated alongside domain experts, and the previously developed scenarios, to ensure a comprehensive set of requirements was produced. Next, the system architecture can be designed, and the requirements subsequently decomposed for each component, producing a draft set of requirements for each component.

Knowledge engineering requirements should then be distilled from the higher-level software requirements, through further knowledge acquisition and scoping. Firstly, a literature review of the semantic resources in the field is recommended, to give context to the model. The software requirements can then be decomposed further into competency questions, conceptually orchestrated through the project's scenarios. These should form a set of formal SPARQL queries which the ontology is then required to answer, and which should be adapted as the project matured and the role of the ontology service becomes clearer. Both the software and ontology requirements should be utilised throughout the project to test and guide the development at each iteration and as a litmus test to assist with validation. Each of the 3 stages of the requirements engineering process are now elaborated in turn.

Platform Impact Scenario Identification and User Requirements

The first milestone of the requirements engineering process is to produce platform-level impact scenarios. These describe the various impact pathways for the software within the client's existing business processes and software frameworks. From these scenarios, project (or whole-system) level requirements should be elicited, from which the knowledge modelling requirements are implied. The development of impact scenarios can begin with informal knowledge gathering through expert consultation, literature review, site visits, and analysis of the existing products and processes at the client organisations. The recommended is now presented, and should be conducted in close collaboration with the industrial stakeholders, to foster early engagement with the developed artefacts, and 'buy-in' of domain experts.

The first stage of the requirements capture process involves achieving a high level understanding of the structure and the processes involved in the client's systems. To achieve this, their water management processes should be collaboratively documented using the IDEF0 [1] functional modelling methodology and network and asset descriptions should be analysed from GIS databases and similar systems. Process models can be produced using the following approach:

1. Document the high-level processes that the water goes through within the system (e.g. treatment).
2. For each process, identify the inputs and outputs.
3. For each process, identify the constraints and mechanisms.
4. Each process in this model should be broken down and the IDEF0 modelling process repeated for each sub-process.

The second stage of the requirements capture methodology builds on the understanding of the client's water processes and topology. This stage involves a) Documenting existing hardware and software, b) Documenting key performance indicators and c) UML (Universal Modelling Language) Use Case Modelling. Documenting the existing technology should contextualise these descriptions within the processes identified through the IDEF0 modelling. To produce use case, the actors listed in the IDEF0 models should be used as a starting point for formalising interactions with software artefacts.

With an understanding of the relevant processes, actors, artefacts, systems and metrics in the client's systems, scenario identification should be undertaken. It is recommended to engage the client in a facilitated workshop, then to develop the resulting ideas into a set of scenarios, grouped by the guiding motivations for the project. Each scenario description should include the following data: scenario name, description, objectives, artefacts to be developed, input data, existing technologies to utilize, output data, actors (during demonstration and at other times), times applicable, and anticipated impact. The scenarios should then be iteratively revised and ranked by the client based on several metrics which ask the question of 'what would a good scenario be?' through scores relevant to the project, such as likely ROI or 'reduction of energy consumption'. Within WISDOM, a detailed dependency analysis was conducted where the high-level goals of the system were stated, and these were mapped to each of the detailed scenarios. This enabled accountability and logical consistency, and empowered a comparison between scenarios based on their interactions towards delivering the high-level goals, although this could be omitted for brevity.

The scenarios provide a critical component in successfully targeting and defining the requirements of the ICT system. The scenarios represent user requirements, as they primarily originate from user elicitation, and use language which is not technological from a software developer's perspective.

System Requirements from Impact Scenario Decomposition

The described impact scenarios serve as a guiding set of initial intentions, which represent a project decomposition and description. These effectively elaborate on the broad goals of the ICT system in a detailed and formal manner. Following this, the next milestone is the development of software requirements at the whole-platform level. A common system analysis and design methodology is recommended where the software solution is considered as a 'black box' through requirement elicitation, iterative refinement and revision, and client validation, as now discussed. Also, meta-requirements are recommended to promote a robust requirement engineering process. These describe requirements of requirements. Those developed for WISDOM can be reused or adapted, and these are documented in D1.3.

Firstly, eliciting an initial set of requirements should be a hybrid top-down, bottom-up process, where the scenarios are used to bound the scope of the solution, and the users and existing systems bound how the scenarios can be achieved. The requirements should therefore be inferred by considering again all the previously produced outputs and attained knowledge, with an emphasis on the views of the solution's users and the scenarios produced. Further use case modelling can also be used at this stage to clarify the target system and compare it to the 'as-is' situation, and the scenarios, and to present the work to the client during iteration. Non-functional requirements can also be elicited through standard questions which address the qualities of the components to be produced. The result of this process should be a set of definitive statements of requirements. These are likely to be varied in terms of terminology, depth of specification and compliance with the meta-requirements, so should be unified and homogenised by abstracting them from their scenario specific contexts and considering the entire set as a description of the overall system. This allows an initial improvement of the requirements as omissions, duplications, ambiguities and variations in terminologies will be exposed.

Following the elicitation and gathering of initial requirement statements, it is necessary to thoroughly analyse these and subsequently revise them until the meta-requirements are adequately met. This process involves considering the requirements within the contexts of the individual scenarios as well as the overall project and industrial constraints such as legislation. This is likely to highlight gaps and redundancy in the requirements. Next, it is recommended to produce sequence diagrams for the main uses of the software in each scenario, which will highlight the required functions and likely inputs and outputs of each. Within WISDOM, an explicit mapping between the 13 discreet scenarios and the functional requirements was used to maintain the train of logic from high level goals to scenarios to requirements, which was discussed previously, although again this may be omitted. After requirements are deemed to describe a sufficient output, they should be checked and revised again for quality, completeness and testability against the meta-requirements.

Once the system-level requirements are deemed sufficient, they should be decomposed further into component-level requirements, including the semantic web software component. The decomposition should primarily use the use cases and sequence diagrams for each component to identify their specific requirements and should again be iterated alongside the system-level requirements and meta-requirements for quality assurance. This allows a natural progression from the project-level requirements to the knowledge modelling requirements of the ontology, which is now described.

Knowledge Modelling Requirement Specification

Finally, after defining the system's impact scenarios and software requirements, the knowledge modelling requirements can be defined. Again, this should be conducted iteratively, in close collaboration with the industrial experts. The ontology scoping and specification is a crucial stage, as it must be detailed and accurate enough for the foreseen queries, utilize sufficient abstraction and breadth for potential future reuse, yet be concise enough to meet the performance requirements of its intended application. This involves domain learning and competency question setting, which are now discussed.

The first stage of the ontology engineering should be knowledge gathering and informal scoping. This requires the ontology engineer to be thoroughly acquainted with the previous outputs and a facilitated workshop with the client is recommended to produce natural language statements which bound the scope of the ontology. This workshop should include an analysis of the WISDOM ontology, which can be used as a starting point for eliciting a scope. It is then recommended to identify and gather related semantic models to gain an understanding of the concepts and modelling decisions which have been used in similar ontologies.

The ontology engineer should then iteratively pose natural language competency questions and compare the body of questions to the previous outputs of the requirements engineering process. These questions represent a range of queries the ontology should be able to answer directly or through inference, and serve as a 'litmus test' of whether the ontology delivers the planned functionality. This then serves as an initial validation of the ontology, with further validation required into whether this 'planned functionality' is sufficient within the intended application, and whether the modelling choices are agreed upon amongst the ICT solution's stakeholders, and ideally beyond. The competency questions should be scenario-driven, by considering the main entities and their properties within each scenario, then forming questions which elicit these properties. Specifically, the competency questions should test sufficient depth and breadth of the ontology into each of the concepts mentioned in the scenarios to fulfil the knowledge management functions specified in the use cases.

As the ontology engineer, will have access to the WISDOM ontology, which is likely to cover many of the concepts required, they should emphasise a comprehensive question set regarding concepts which are beyond the scope of the WISDOM ontology. Any other donor ontologies identified for reuse should then be compared against the competency questions to determine which should ultimately be incorporated. Finally, these ontologies should be integrated and aligned as required with the WISDOM ontology to produce a starting point for the ontology curation process.

The competency questions should also be viewed as a living resource, and should be referred to and updated regularly, as the understanding of the domain and project becomes more mature. It is recommended that this is conducted within further workshops alongside the client. Once the requirements are stable, and assuming additional modelling was deemed necessary, the curation of the ontology can begin, as described next.

Manual Extension to Meet Additional Requirements

As has been recommended in the previous section, the WISDOM ontology curation also began from a set of aligned donor ontologies. It is therefore recommended to adopt a similar approach to the extension of the meta-model as described in D2.2, and which is briefly summarised here. The main stages in extending the ontology are to enumerate the additional required concepts, build a class hierarchy from this which is integrated with the existing model, then define class relationships and data properties before stating restrictions. A full explanation of the nature of the various aspects of ontological modelling is outside the present scope; for an understanding of concepts such as object properties, data properties and classes the reader is asked to refer to literature on the field. The main steps in this process are now briefly summarised in turn.

Concept enumeration involves using the requirements engineering outputs such as the scenarios to list all the types of object and any other key aspects of the domain's vocabulary. As a rule of thumb, nouns will typically represent classes of objects in the domain and adjectives and verbs may represent properties and class relationships. Any data schemas held by the client can also be analysed.

From a list of object types in the domain, the ontology engineer should then iteratively sort these into a hierarchy within the modelling decision already taken by the donor ontologies. Further modelling decisions should be based on the knowledge acquired previously, common sense, and domain expert consultation. This class hierarchy represents a simple categorisation of types of concepts in the domain. This can be extended if needed with class axioms, such as to indicate disjoint or equivalent classes.

Object properties can then be modelled in agreement with the domain knowledge previously acquired. These represent relationships between the objects in the domain which are necessary to express to meet the competency questions such as topological relationships, meteorological relationships, and domain specific considerations. These properties should be described themselves too, such as whether they are transitive, and their cardinality. Relevant domain specific data properties should then be added, which relate the objects modelled to data literals. Again, these should be described, such as whether they are functional, and what the range must be.

Finally, further restrictions can be modelled if needed, to supplement the potential inference and further enforce correct use of the model. This includes SWRL rules, which adopt an 'if A Then B' structure to inferring new statements of truth. Once this first draft has been completed, the competency questions should be used to check if the ontology is sufficient, and typically to guide further iterations of the process.