D4.4 Semantic Interoperability Design Requirements

digital-water.city
digitalwater_eu

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Deliverable N°	Semantic interoperability design requirements
Related Work Package	WP4
Deliverable lead	21-SINTEF
Author(s)	Audun Vennesland, SINTEF Peter Halland Haro, SINTEF Bård Johan Hanssen, SINTEF
Contact for queries	audun.vennesland@sintef.no
Grant Agreement Number	n° 820954
Instrument	HORIZON 2020
Start date of the project	01 June 2019
Duration of the project	42 months
Website	www.digital-water.city
License	This work is licensed under a Creative Commons Attribution 4.0 International License
Abstract	D4.4 describes design requirements for implementing a semantic interoperability middleware architecture in the DWC project. Design requirements in the form of data requirements and software component requirements are derived from scenario descriptions developed in collaboration with the Milan and Paris cases in DWC. The data requirements are mapped to existing and standardised data models while the software component requirements are mapped to relevant generic software components from the FIWARE Framework.

Dissemination level of the document

X	PU	Public
	PP	Restricted to other programme participants
	RE	Restricted to a group specified by the consortium
	CO	Confidential, only for members of the consortium

Versioning and contribution history

Version*	Date	Modified by	Modification reasons
D1	30.06.2020	Audun Vennesland	First draft of structure and contents
D2	22.11.2020	Audun Vennesland	Draft version of full report sent to reviewers
R1	08.12.2020	Audun Vennesland	Version addressing comments from internal review
R2	08.12.2020	Nico Caradot	Review by the coordinator
S	09.12.2020	Audun Vennesland	Final version for submission

* The version convention of the deliverables is described in the Project Management Handbook (D7.1). *D* for draft, *R* for draft following internal review, *S* for submitted to the EC and *V* for approved by the EC.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Table of content

List of figures

digital-water.city
digitalwater_eu
digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

List of tables

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Glossary

Table 1. Glossary

Term	Description
API	Application Programming Interface.
CAP	Operator of Greater Milan wastewater treatment
CoP	Community of Practice
DWC	Digital Water City
Environment system	The "environment system" includes everything that is not a part of the target system, and which interfaces the target system directly. The term system should here be interpreted in a broad sense and encompasses both stakeholders and technical systems.
EWS-BWQ	Early Warning System for Bathing Water Quality to be developed in Paris.
EWS-SWR	Early Warning System for Safe Water Reuse to be developed in Milan.
FIB	Fecal Indicator Bacteria
FIWARE	FIWARE is an initiative that defines reusable open-source components and standardised specifications for context data management.
HTTP	Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML.
JSON	JavaScript Object Notation. A lightweight data interchange format.
KWB	Kompetenzzentrum Wasser Berlin
M2M	Machine-to-Machine
MQTT	Message Queuing Telemetry Transport (MQTT) is an OASIS standard messaging protocol for the Internet of Things (IoT).
NGSI	Next Generation Service Interface. A protocol to manage context information.
OPC-UA	Open Platform Communications Unified Architecture (OPC UA) is a machine-to-machine communication protocol for industrial automation.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

OWL	Web Ontology Language. A Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things.
RDF	Describes data by defining relationships between data entities expressed using URIs (Uniform Resource Identifiers) and related via triples in the form (subject-predicate-object).
REST	Representational State Transfer. A software architectural style that defines constraints to be used for creating web services.
SIAAP	Syndicat Interdepartmental pour L'Assainissement de l'Agglomeration Parisienne, Operator of greater Paris wastewater treatment
SPARQL	A query language that can be used to express queries across diverse data structures following the RDF specification.
Target system	The system for which the architectural description is created. The term system should here be interpreted in a broad sense and encompasses both stakeholders and technical systems.
UAV	Unmanned Aerial Vehicle
UML	Unified Modelling Language, a modelling language for describing system- and software architecture.
View	A representation of a whole system from the perspective of a related set of concerns.
Viewpoint	A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purpose and audience for a view and the techniques for its creation and analysis.
WWTP	WasteWater treatment plant
XML	eXtensible Markup Language. A markup language that defines a set of rules for encoding, storing and communicating data.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Executive summary

This report defines a set of design requirements for achieving semantic interoperability in the DWC water value chains. Starting from an initial analysis of relevant data requirements and system components performed in task 4.1 (described in deliverable D4.1 [1]), the requirements collection process described in this report has focused on (1) deriving a uniform specification of data entities relevant for the DWC project as well as (2) identifying the set of software components that are needed for the semantic interoperability middleware in DWC. Non-functional requirements beyond interoperability, such as security and scalability, have not been considered in this work. The data specifications and software components result from an analysis of scenario descriptions of the Milan and Paris cases and their digital solutions. The approach used for capturing and describing the requirements in this report is guided by the ARCADE Framework[1], resulting in a conceptual architecture of semantics and generic software components that can guide other deployments in DWC. The conceptual architecture will also prepare for further developments in subtask 4.3.2 (Developing the DWC reference ontology) and subtask 4.3.3 (Development of the semantic interoperability middleware).

[1] http://arcade-framework.org/ (last accessed 07.12.2020)

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

1. Introduction

1.1. Summary of DWC and objectives of work package 4 and subtask 4.3.1

DWC aims at creating digital solutions to link water management in the physical world to the digital spheres such as sensor networks, real-time monitoring, machine learning etc.

24 partners from 10 countries work together in the case study cities Berlin, Copenhagen, Milan, Paris and Sofia and support the utilities and municipalities in improving water quality, return on investment and public information about water-related issues.

The core of work package (WP) 4 is to ensure that the digital solutions are designed and developed in a way in which data and information are exchanged in a cyber-secure and interoperable way. This includes a risk analysis and proposition of risk reduction measures to protect data and systems from unauthorized access as well as the description of semantic models and interoperability design guidelines. The tasks focus on the digital solutions and their impacts on the existing infrastructures, but not on general cyber-physical risks.

Subtask 4.3.1 of WP4 aims to identify design requirements for achieving semantic interoperability in DWC water value chains and for preparing further developments in subtask 4.3.2 (Developing the DWC reference ontology) and subtask 4.3.3 (Development of the semantic interoperability middleware). In order to prepare for the development of semantically interoperable middleware, the work described in this report targets three objectives: (1) capture relevant data requirements from DWC water value chains and identify common data concepts; (2) identify the software components that are needed to achieve the desired functionality from DWC water value chains and perform a mapping with existing technologies that can be used to realize the components. In this work the scope of DWC water value chains is represented by the implementations in Milan and Paris; (3) document the requirements as generically as possible in order to support reusability in other deployments taking place in the DWC project. With respect to the latter objective, the structuring and notation used in this report is guided by the ARCADE Framework, a domain- and technology independent architectural description framework for software intensive systems.

1.2. Structure of this document

Chapter 2 describes the approach for collecting and representing the design requirements using the ARCADE Framework; chapter 3 describes the expected "to-be" scenarios in Milan and Paris respectively using the context and component views from ARCADE; chapter 4 specifies the elicited data entities relevant for accomplishing the scenarios described in chapter 3; chapter 5 describes the software components relevant for the interoperability middleware architecture, while chapter 6 summarises conclusions and relevant further work.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

2. Overall Approach

The approach used for capturing and describing the requirements needs to take into account that the requirements will be used as a basis for developing interoperability middleware (objective of task 4.3.3) as well as defining the inherent semantics covering aspects related to DWC to ensure interoperability on the semantic level by the use of a common semantic model (its development is performed in task 4.3.2). On this basis, the approach for eliciting design requirements needs to combine (1) more “traditional” requirements engineering principles with (2) a tailored ontology engineering approach for detailing requirements and development of a semantic model relevant for DWC solutions.

For the former the ARCADE Framework will be applied to guide the requirements collection as well as the development of a systems architecture for the interoperability middleware. The ARCADE Framework is a domain- and technology independent architectural description framework for software intensive systems. It aims to assist in creating, understanding, and describing the architecture of software systems. The framework prescribes how an architecture should be designed for reusability and maintainability and lends itself well to capture and preserve both conceptual and detailed concepts that can be generalised and transferred to other implementations in the DWC project and beyond.

Table 2 describes the viewpoints from ARCADE that will be used in work. The views created from the *Context Viewpoint* and the *Component Viewpoint* are relevant both for the semantic model and the more “technical” parts of the interoperability middleware, while retaining a quite generic focus. The *Realisation Viewpoint* focuses on how the architecture can be realised by means of specific technologies. In DWC this primarily means using FIWARE[2] components, but with necessary extensions that allow for semantic interoperability in DWC data exchange. The language used for describing these views is the Unified Modelling Language (UML). A short description of the UML notation used in this report is provided in Annex B.

Table 2. ARCADE Framework

Viewpoint	Description
Context Viewpoint	Describes the environment to the target system in terms of its business-related aspects, other involved technical systems and the mapping of business aspects to the target system.
Component Viewpoint	Describes the system in terms of its subsystems and information objects, and document how subsystem interaction and information processing is carried out in order to provide the desired behavioural effect.
Realisation Viewpoint	Describes the realisation of the target system in terms of its subsystems. The view will describe how to structure implementation and deployment the target system. In particular, the Realisation View will describe how the requirements elicited in the previous views can be realised using FIWARE.

[2] https://www.fiware.org/ (last accessed 07.12.2020)

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Figure 1 shows the views applied and how the approach consists of a combination of the ARCADE Framework. It further shows how the design requirements elicited in this work will be applied for the developments to be made in tasks 4.3.2 and 4.3.3.

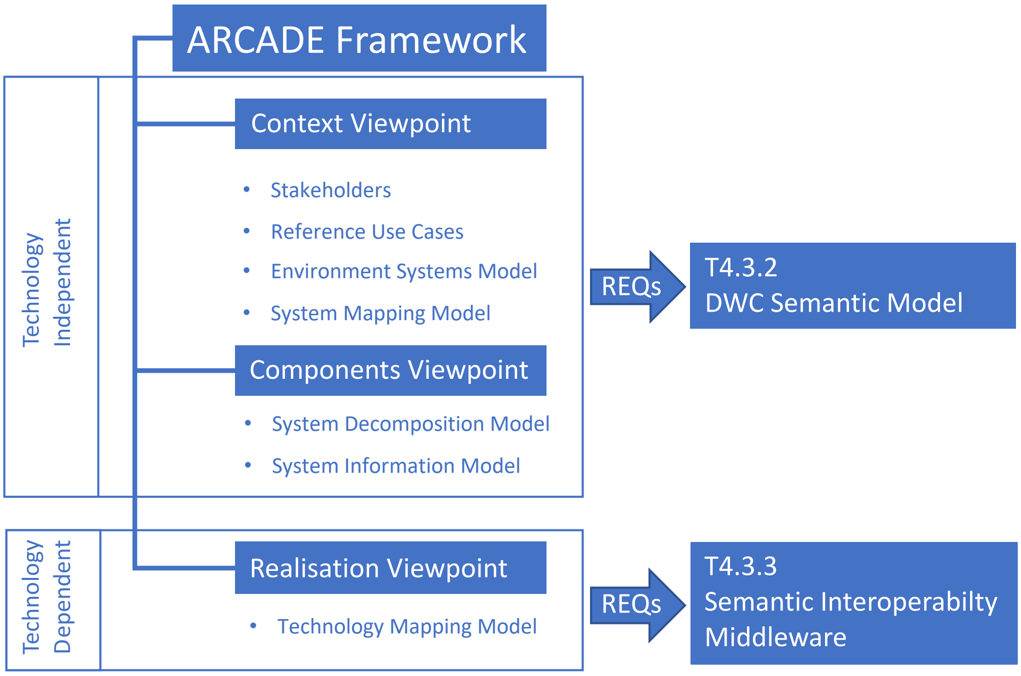


Figure 1. The ARCADE Framework

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

3. Scenario description of the Milan and Paris cases in DWC

This chapter describes scenarios that define how DWC digital solutions will be implemented in Milan and Paris considering that the semantic interoperability middleware is in place. The scenarios are described by means of a *Context View* which defines the relevant stakeholders and systems that are involved, the functionality required for realising the solutions and their objectives; and a *Component View* that provides a generic description of the software components that are needed to realise the scenarios.

3.1. Milan

Table 3 describe the digital solutions that will be implemented in Milan:

Table 3. Digital Solutions in Milan

Digital Solution	Description	Technology Provider
DS1. Sensor for real-time bacterial measurements	DS1 sends measurements for bacteria concentrations in surface water to DS3.	Fluidion
DS3. Early warning for safe water reuse (EWS-SWR)	Receives data from various sensors (including DS1) and processes these data on an online platform. The processing includes machine learning, visualisation of statistical data and aims to predict contamination risks.	ISS
DS4. WebGIS platform (upgrade) for improved decision-making related to safe water reuse.	The WebGIS platform will receive processed data from DS3 and visualizes the data received using existing tools. The platform integrates data from multiple sources and will include new users (farmers).	CAP
DS5.1 Unmanned Aerial Vehicle (UAV) for monitoring of water stress.	Multi-spectral imagery sensors are mounted on an UAV. UAV and ground sensor data are processed to map water stress.	UNIMI
DS5.2 Web-based platform that performs matchmaking between irrigation water demand and safe water availability.	Data from the solutions above will be integrated in this web-based platform. Farmers send requests to the platform and receive irrigation advice and safety warnings. The WWTP managers can have a visualization of water demand and supply.	UNIVPM
DS6. Web-based serious game on water reuse – carbon – energy – food – climate nexus.	Provides near real-time audit of water reuse – carbon – energy – food – climate nexus. Aims to engage the public to overcome social and economic barriers to water reuse.	UNIVPM

The emphasis in this report will be on solutions DS1, DS3, DS5.1 and DS5.2 and how they will interact in the Milan implementation.

3.1.1. Context View

This section describes the core functionality of the Milan case in terms of target system stakeholders, stakeholders operating in the environment of the target system, and a set of reference use cases.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Target System Stakeholders

Figure 2 illustrates the target system stakeholders and systems in the Milan case and Table 4 includes a more detailed description of them.

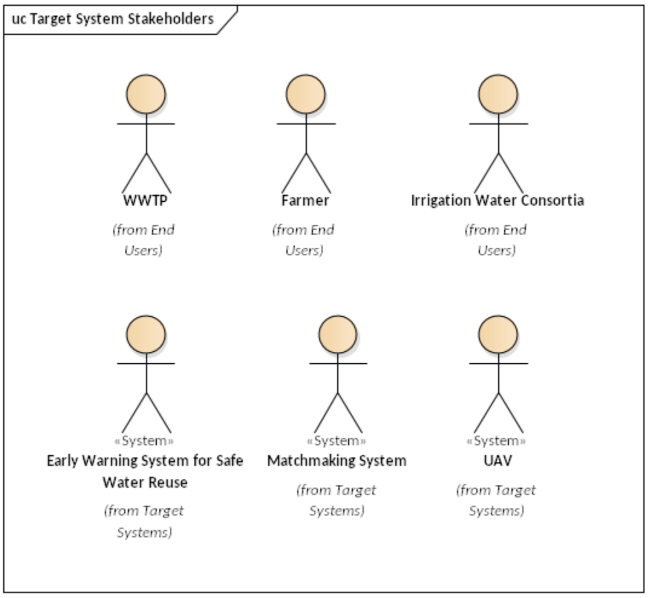


Figure 2. Target System Stakeholders and Systems in Milan

Table 4. Description of Target System Stakeholders and Systems in Milan

Stakeholders and Systems	Description
Matchmaking System	This system will find the optimal match between water demand from the farmers and the offered irrigation water from the WWTP and irrigation consortia. Will be developed as a web-based platform that farmers and irrigation water consortia also can interact with.
Farmer	A farmer has a need for irrigation water with a certain level of quantity and quality.
WWTP	The wastewater treatment plan will provide effluent water to be used for irrigation and provide data about its quantity and quality.
Irrigation Water Consortia	Responsible for supplying water to the farmer.
Early Warning System for Safe Water Reuse (EWS-SWR)	This early warning system will analyse microbial and toxic contamination linked to water reuse and inform relevant stakeholders whenever quality thresholds are compromised according to the results of WP1.
UAV	The Unmanned Aerial Vehicle (UAV) will provide multi spectral imagery in order to detect water stress.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Environment System Stakeholders

The Environment System Stakeholders relevant in the Milan case are illustrated in Figure 3. These are other systems that the target system needs to interact with in order to, on the one hand, provide quality measures of the irrigation water at the treatment plant, and on the other hand perform various types of measurements that help to quantify the water demand. In the Milan case there are sensors both at the WWTP to measure various water quality parameters and in the fields to perform measurements on the crop-soil system.

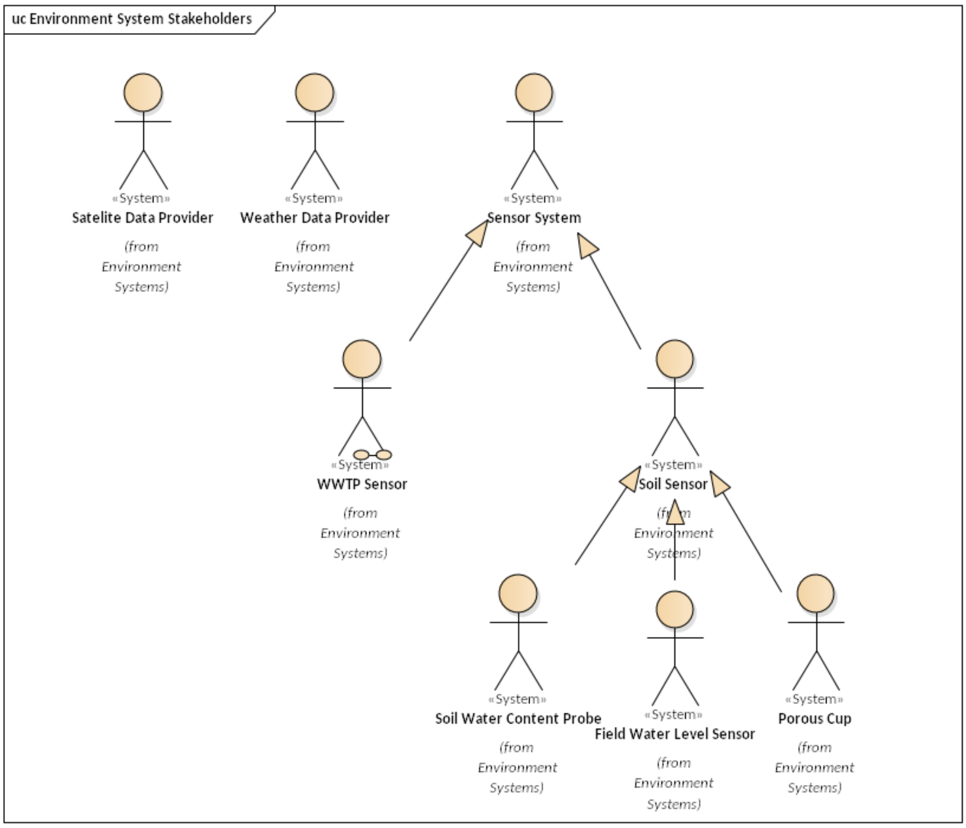


Figure 3. Environment System Stakeholders and Systems in Milan

Table 5 provides a description of each stakeholder.

Table 5. Description of Environment System Stakeholders and systems in Milan

Stakeholders and systems	Description
Sensor System	A generalisation of all sensors (probes), both sensors at the WWTP and ground sensors.
WWTP Sensor	The WWTP Sensor is a generalisation of sensors used for various types of measurement at the wastewater treatment plant. This includes flow

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

	meters, microbiological sensors, ammonium sensors, phosphate sensor, etc.
Soil Sensor	The Soil Sensor is a generalisation of sensors used for soil measurements at the farmers´ crops. This includes field water level sensors, soil water content probes, porous cups.
Soil Water Content Probe	Measures soil moisture in crop.
Field Water Level Sensor	A sensor that measures the water level in an agricultural area.
Porous Cup	Perform soil water sampling in order to evaluate microbial contamination and nutrient contents.
Weather Data Provider	Weather data are provided by means of data retrieval from external weather providers via APIs and from sensors measuring temperature and humidity.
Satellite Data Provider	Satellite data (imagery) will be accessed from external service providers (via an intermediate micro service).

Reference Use Cases

The use cases illustrated in Figure 4 describe the core functionality of the target system to be implemented in Milan as well as functionality provided by the environment. Ultimately, the Matchmaking System will optimally match the water demand from the farmers and the available water from the wastewater treatment plant. At an overall level the stakeholders involved are the WWTP (Peschiera Borromeo), the Irrigation Water Consortia responsible for supplying water to the farms, and of course the farmer with a water demand. The matchmaking depends on two fundamental sub-use cases: (1) the quantification of water demand, and (2) the quantification of irrigation water availability. The quantification of water demand includes to measure the water stress index on and measurements on weather and soil conditions. The measurement of the water stress index is done by the Unmanned Aerial Vehicle (UAV). The input data for measuring the water stress index is provided by imagery from the UAV and satellite data (from external providers) which are combined with measurements from ground sensors and weather data. The quantification of irrigation water availability relies on the determination of the quantity and quality of water from the WWTP based on measurements from flow meters and sensors at the WWTP (WWTP sensors) respectively. The EWS-SWR is responsible for analysing the water quality and whenever quality parameter thresholds are compromised to issue warnings to relevant stakeholders.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

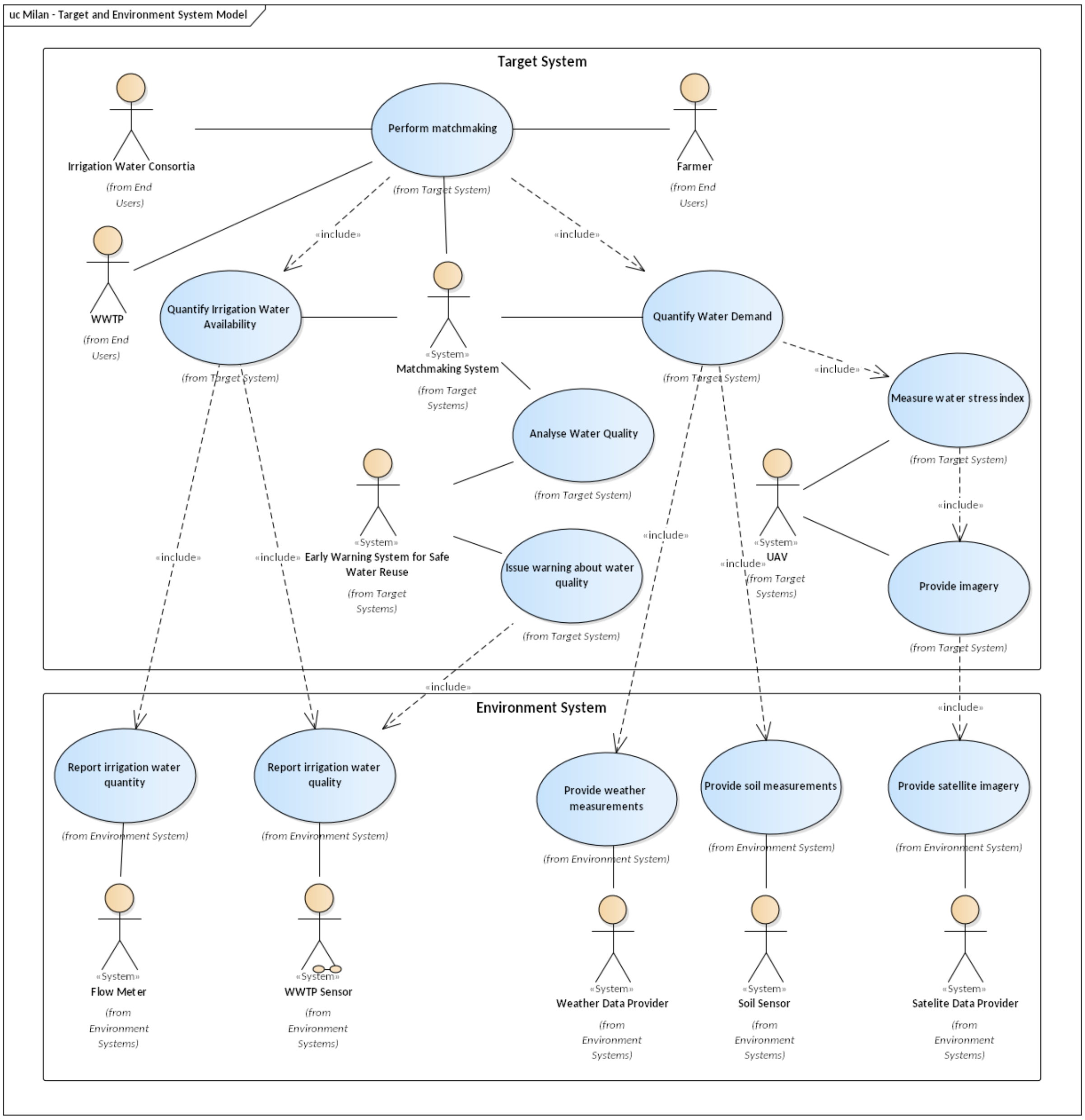


Figure 4. Reference use cases for Milan

System Mapping Model

Figure 5 illustrates the use cases that are part of the target system to be developed and the environment systems.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

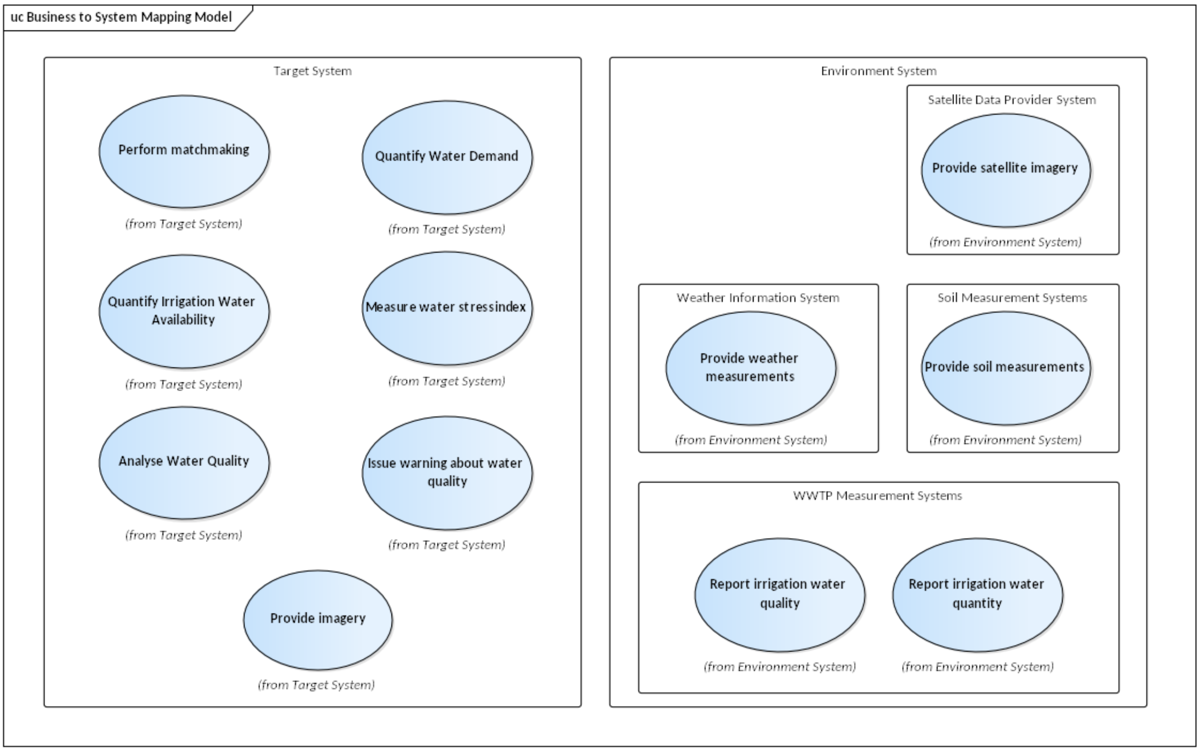


Figure 5. System Mapping Model for Milan

3.1.2. Components View

Based on the description of the context view in the previous section, this section will describe a set of software components needed to realise the implementation in Milan (System Decomposition Model), as well as an initial specification of data to be communicated via the component interfaces.

System Decomposition Model

Figure 6 depicts a set of components relevant for the Milan case. Note that some details are omitted in this diagram (e.g., components that transform between different formats and protocols, separation of ports and interfaces), the purpose is to illustrate the interaction between components within the target system and with the environment. This component diagram, which is technology independent, will be extended in the Realisation View (section 5.2) where the technology to realise the implementation will be included along with an additional level of detail.

A multitude of data will be collected from the CAP Control Room (describing water availability and quality), ground sensor systems (describing soil characteristics), meteorological data, and local (from UAV) and "global" (from Satellite data) imagery data. The Matchmaking System will employ all these data, in addition to processed data from the EWS-SWR, to compute the best match of water demand and supply. The EWS-SWR will use measurements related to the irrigation water amount and quality for microbiological contaminant detection in wastewater from the WWTP as well as data from the UAV to inform stakeholders and prevent microbial and toxic contamination linked to water reuse.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

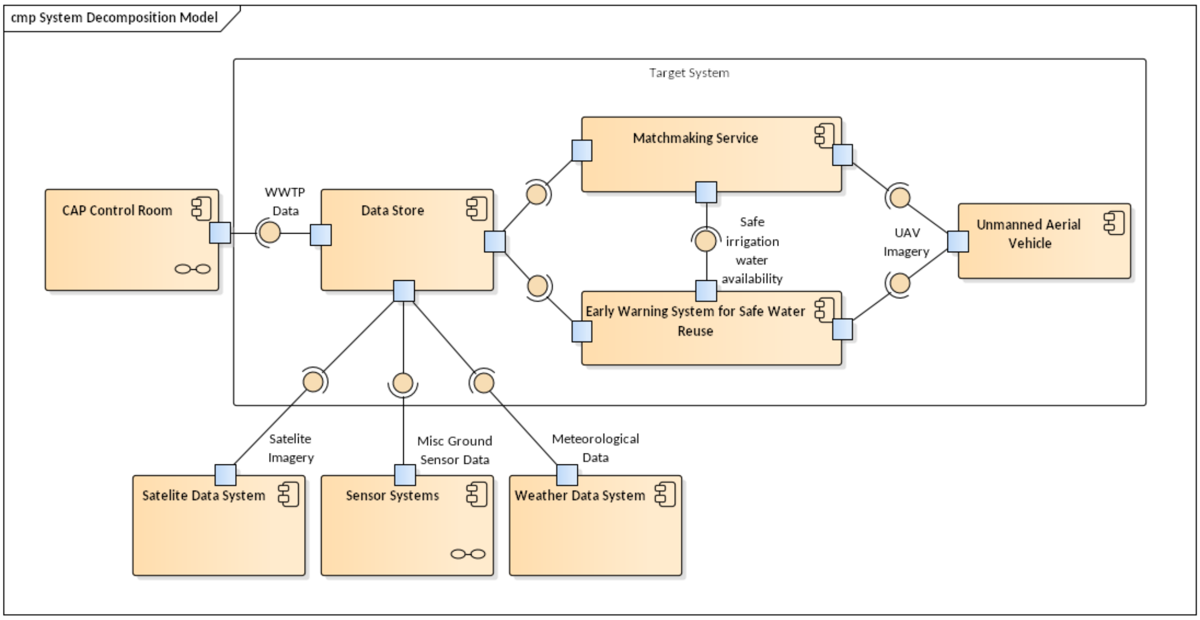


Figure 6. System Decomposition Model for Milan

System Information Model

In this view the initial data requirements from the systems involved in the Milan case are analysed and defined. Looking at the interfaces of the System Decomposition Model in the previous section, the data requirements sections highlighted in yellow in Figure 7 are relevant and will be described in the following. At this stage the data requirements are listed as more or less independent entities, the objective is not to define their relationships or cardinalities (mandatory, optional, multiple). Furthermore, the datatype specification is only indicative at this point. A more comprehensive analysis of these aspects is described in Chapter 4.

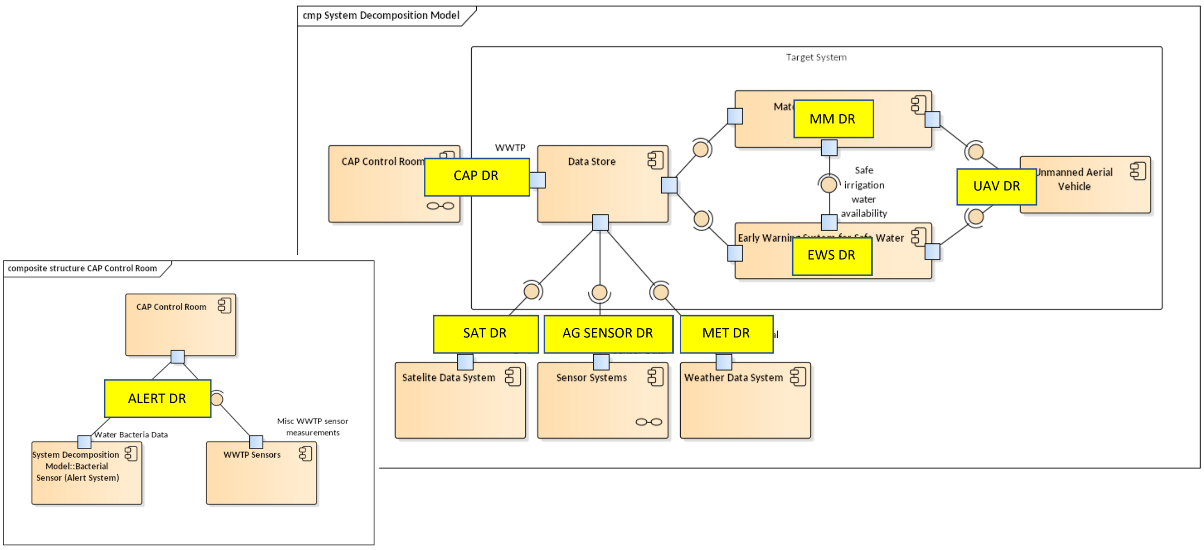


Figure 7. Data Requirements (DR) Overview for the Milan case

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

A fundamental input to the Early Warning System and the Matchmaking System is provided by the CAP Control Room (the WWTP). This interface will provide a variety of data entities that will enable a quantification of the available irrigation water as well as determine its quality.

Table 6. [CAP DR] CAP Control Room data requirements

Data Requirement	Datatype	Definition
WWTP Sensor Data	Sensor Entity	A multitude of sensors are used at the WWTP. These include pH sensor, REDOX sensor, DO sensor, Conductivity sensor, TSS sensor, Turbidity/TSS sensor, Ammonium sensor, Nitrate sensor, Phosphate sensor, Temperature sensor, UV-transmittance sensor, UV-organic load sensor, Total Organic Carbon (TOC) sensor. The WWTP will employ the Fluidion Alert system for bacteria measurements.
WWTP Water Flow Data	Water Measurement Entity	Describes the water flow at different stages at the WWTP. This entity must support the definition of topological location specification that enable to state that a measurement is related to a particular process stage (e.g., influent or effluent).
WWTP Water Level Data	Water Measurement Entity	Describes the water level in water storage tanks at the WWTP.
WWTP Device Status	Status Entity	Describes the status of device (e.g., electromechanical equipment) at the WWTP.
WWTP Location Data	Location Entity	Specification of the geographical and topological location for which a measurement is made.
WWTP Timestamp	Time Entity	Timestamp for when a particular measurement (sensor, water flow, water level, device status) is valid.

The EWS-SWR will receive all data from the CAP Control Room (specified in Table 6) and use that along with data from the UAV to analyse water contamination.

Table 7. [EWS-SWR DR] Early Warning System data requirements

Data Requirement	Datatype	Definition
WWTP Sensor Data	Sensor Entity	A multitude of sensors are used at the WWTP. These include pH sensor, REDOX sensor, DO sensor, Conductivity sensor, TSS sensor, Turbidity/TSS sensor, Ammonium sensor, Nitrate sensor, Phosphate sensor, Temperature sensor, UV-transmittance sensor, UV-organic load sensor, Total Organic Carbon (TOC) sensor
WWTP Water Flow Data	Water Measurement Entity	Describes the water flow at different stages at the WWTP. This entity must support the definition of topological location specification that enable to state that a measurement is related to a particular process stage (e.g., influent or effluent).
WWTP Water Level Data	Water Measurement Entity	Describes the water level in water storage tanks at the WWTP.
WWTP Device Status	Status Entity	Describes the status of device (e.g., electromechanical equipment) at the WWTP.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Alert	Alert Entity	Describes alerts that enable identification of misfunction that may influence the process.

The Matchmaking System requires data from the CAP Control Room (see Table 6), ground sensor data (see Table 9), meteorological data (see Table 10) and processed data from the EWS-SWR (see Table 7).

Table 8. [MM DR] Matchmaking System data requirements

Data Requirement	Datatype	Definition
WWTP Sensor Data	Sensor Entity	A multitude of sensors are used at the WWTP. These include pH sensor, REDOX sensor, DO sensor, Conductivity sensor, TSS sensor, Turbidity/TSS sensor, Ammonium sensor, Nitrate sensor, Phosphate sensor, Temperature sensor, UV-transmittance sensor, UV-organic load sensor, Total Organic Carbon (TOC) sensor
WWTP Water Flow Data	Water Measurement Entity	Describes the water flow at different stages at the WWTP. This entity must support the definition of topological location specification that enable to state that a measurement is related to a particular process stage (e.g., influent or effluent).
WWTP Water Level Data	Water Measurement Entity	Describes the water level in water storage tanks at the WWTP.
WWTP Device Status	Status Entity	Describes the status of device (e.g., electromechanical equipment) at the WWTP.
Ground Sensor Data	Sensor Entity	Describes various soil measurements provided by sensors and loggers, such as water level sensors, soil water content probes, and porous cups. See [AG SENSOR DR].
Meteorological Data	Meteorological Data Entity	Describes various meteorological measurements, such as rainfall, temperature, windspeed, wind direction. See [MET DR].

Table 9. [AG SENSOR DR] Agricultural sensor data requirements

Data Requirement	Datatype	Definition
Soil Water Level	Water Measurement Entity	Measure water level in crop fields.
Water Content Probes	Sensor Entity	Measure soil moisture in crop.
Porous Cup Logger	Sensor Entity	Perform soil water sampling in order to evaluate microbial contamination and nutrient content.

Table 10. [MET DR] Meteorological data requirements

The data requirements in this table are described as enumeration values that needs a measurement value (expressed as e.g., a decimal value) in an accompanying attribute.

Data Requirement	Datatype	Definition
Rainfall	Meteorological Data Entity (as enum value)	Part of measurement from weather stations to measure the water need of crops.
Air Temperature	Meteorological Data Entity (as enum value)	Part of measurement from weather stations to measure the water need of crops.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Wind Speed	Meteorological Data Entity (as enum value)	Part of measurement from weather stations to measure the water need of crops.
Wind Direction	Meteorological Data Entity (as enum value)	Part of measurement from weather stations to measure the water need of crops.
Solar Radiation	Meteorological Data Entity (as enum value)	Part of measurement from weather stations to measure the water need of crops.

Table 11. [UAV DR] Unmanned Aerial Vehicle data requirements

Data Requirement	Datatype	Definition
Usually, multi-spectral images that output images with pixel values in reflectance units (in some case float, in other case uint16)		

Table 12. [SAT DR] Satellite imagery data requirements

Data Requirement	Datatype	Definition
Usually, multi-spectral images that output images with pixel values in reflectance units (in some case float, in other case uint16)		

3.2. Paris

Table 13 describes the digital solutions to be implemented in Paris. DS2.1 and DS18 are considered part of the target system, whereas DS1, that will be an integrated component in the setup of the Paris case, will be considered a part of the environment system.

Table 13. Digital Solutions in Paris

Digital Solution	Description	Technology Provider
DS1. Fully automated sampling and measurement device for indicating bacteria in surface water.	DS1 sends measurements for bacteria levels in surface water to DS2.1.	Fluidion
DS2.1 Early Warning System for bathing water quality (EWS-BWQ).	Data are collected from various data sources and providers and integrated in a web-based data platform; data connection to a web application running on a separate server integrating data processing and output for further visualization and decision making. The EWS-BWQ quality consists of two sub-components: (1) the deterministic model ProSe developed and maintained by SIAAP and (2) a statistical model developed and maintained by KWB.	KWB-SIAAP
DS18 Bathing water quality visualisation for public information (mobile application).	DS18 is a mobile application that will provide bathing water information to citizens. This is called the Public App in the following. Related to this application, there will also be an Expert App which will convey information about bathing water quality to bathing site managers allowing them to make informed decisions and communicating these decisions to the Public App.	SIAAP

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

3.2.1. Context View

This section describes the core functionality of the Paris case in terms of target system stakeholders, stakeholders operating in the environment of the target system, and a set of reference use cases.

Target System Stakeholders

Figure 8 illustrates the target system stakeholders in the Paris case and Table 14 provides a more detailed description of them.

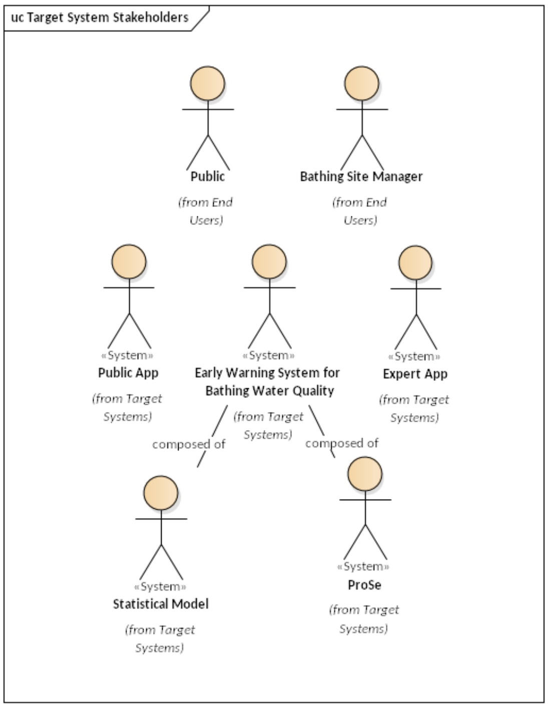


Figure 8. Target System Stakeholders and Systems in Paris

Table 14. Description of Target System Stakeholders and Systems in Paris

Stakeholders and Systems	Description
Public	These are citizens with an interest of receiving information about the status of bathing sites, complementary information on water quality and how it is determined. This information will be provided via the Public App, a mobile application that will communicate bathing water contamination risks to citizens.
Bathing Site Manager	The bathing site managers make informed decisions on whether to close/open bathing sites using information available on the Expert App.
Public App	The Public App is a user interface towards the citizens and provides status of bathing sites. The app might

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

	also contain complementary information on water quality, how it is determined... The amount and nature of the information shared on the app will be discussed and decided during the CoP meetings. The Public App might also receive a subset of information, such as predictions of the status of bathing sites for the week for example from the Early Warning System.
Expert App	The Expert App is a user interface towards the Bathing Site Manager. As well as for the public app, the CoP meeting will help us determine how much information we need to provide for them to make a decision (Flowrates, ALERT results…). Other information, such as attendance/crowd at a site, water temperature etc., can be provided by the managers. The Expert App receives a subset of information from the Early Warning System.
Early Warning System for Bathing Water Quality (EWS-BWQ)	This system, which integrates the Statistical Model and ProSe, will determine the Fecal Indicator Bacteria (FIB) concentration in the bathing sites.
Statistical Model	The Statistical Model uses Bayesian regression or Machine Learning to simulate FIB concentration in the bathing sites. The input data for the statistical model is rainfall, river flow and discharge information (from WWTP and stormwater overflows).
ProSe	ProSe will simulate FIB concentration in bathing sites based on flowrates and water quality measurements at upstream rivers, and discharge information (from WWTP and stormwater overflows) as input data. It is based on the resolution of hydraulic and bacterial equations.

Environment System Stakeholders

Figure 9 visualises the environment system stakeholders in Paris, and a description of these is provided in Table 15.

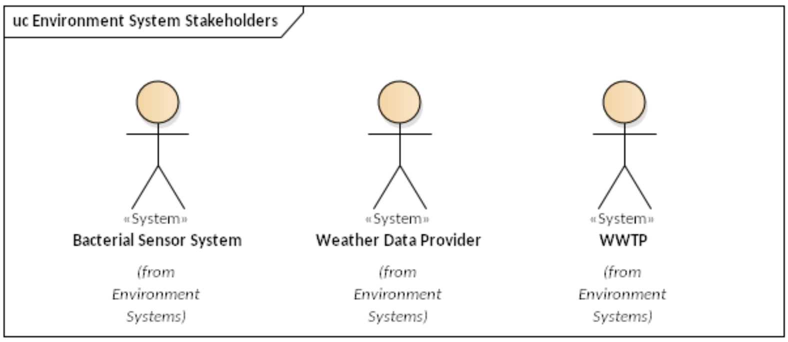


Figure 9. Environment System Stakeholders and Systems in Paris

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Table 15. Description of Environment System Stakeholders and Systems in Paris

Stakeholder	Description
Bacterial Sensor (Alert System)	Measures Fecal Indicator Bacteria measurements in the Seine and Marne rivers and communicates these measurements to the EWS-BWQ.
Weather Data Provider	Provides rainfall data to the EWS-BWQ.
WWTP	Provides information about discharge from the WWTP and stormwater overflows to the EWS-BWQ. This information is provided by the system EDEN/MAGES that process and communicates raw source data from various operators.

Reference Use Cases

The use cases depicted in Figure 10 describe the core functionality of the target system to be implemented in Paris as well as functionality provided by the environment. Note that this model does not suggest a workflow, just the core functionality and stakeholders / systems involved. The EWS-BWQ is responsible for the prediction of water quality at the bathing sites. The predictions are made by the Statistical Model which employs numerical simulations provided by ProSe.

In order to arrive to these predictions, input data from different environment systems need to be retrieved. These data come in the form of discharge data (from the WWTP system EDEN/MAGES), meteorological data (from Weather Data Providers) and bacterial measurements (from the bacterial sensors provided by the Alert system). Finally, predictions are communicated to the Expert App where the bathing site managers use these predictions along with other information to make decisions concerning the bathing sites. These decisions are then forwarded to citizens via Public App.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

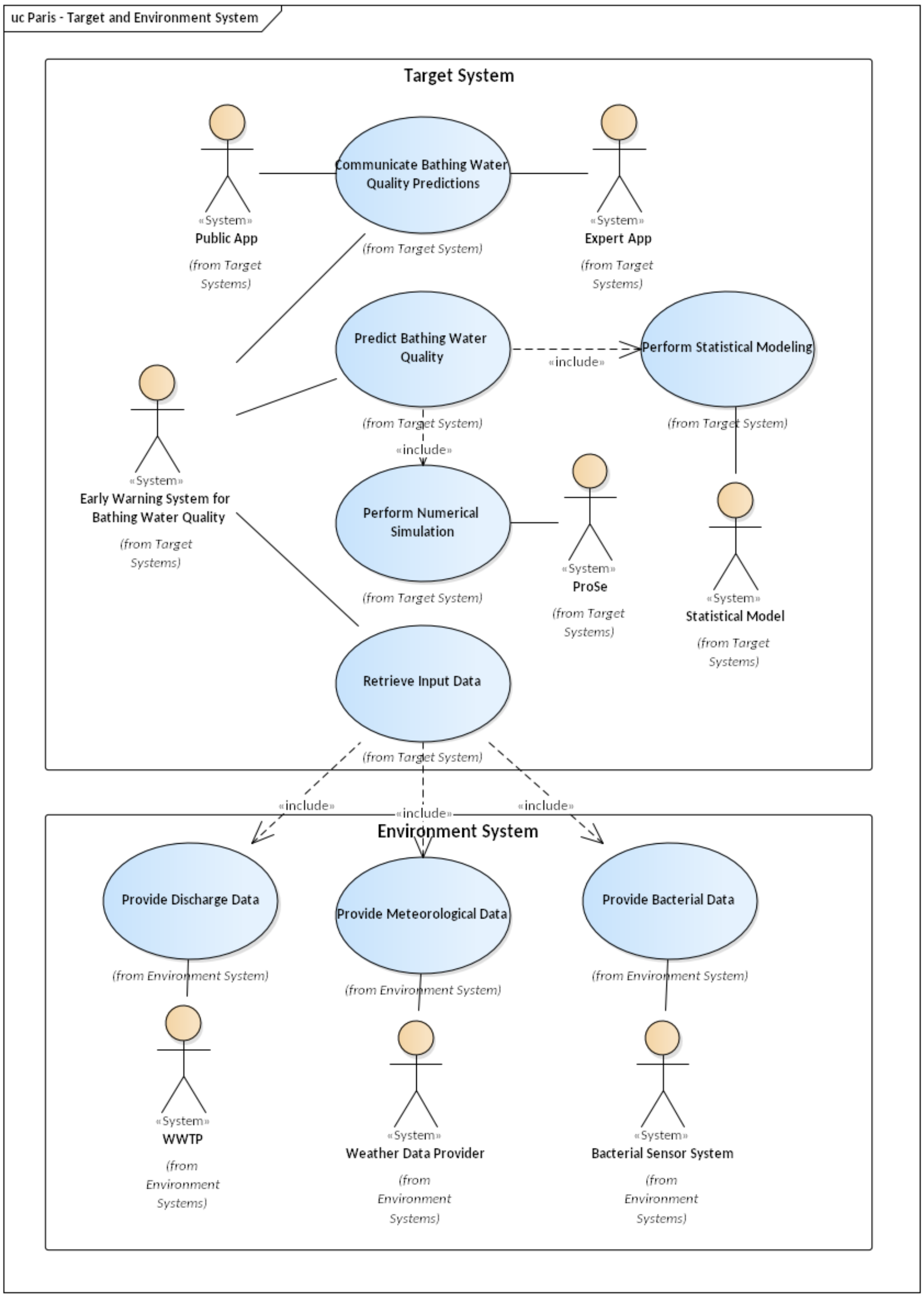


Figure 10. Reference use cases for Paris

System Mapping Model

Figure 11 provides an overview of the use cases that are a part of the target system to be developed and the environment systems.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

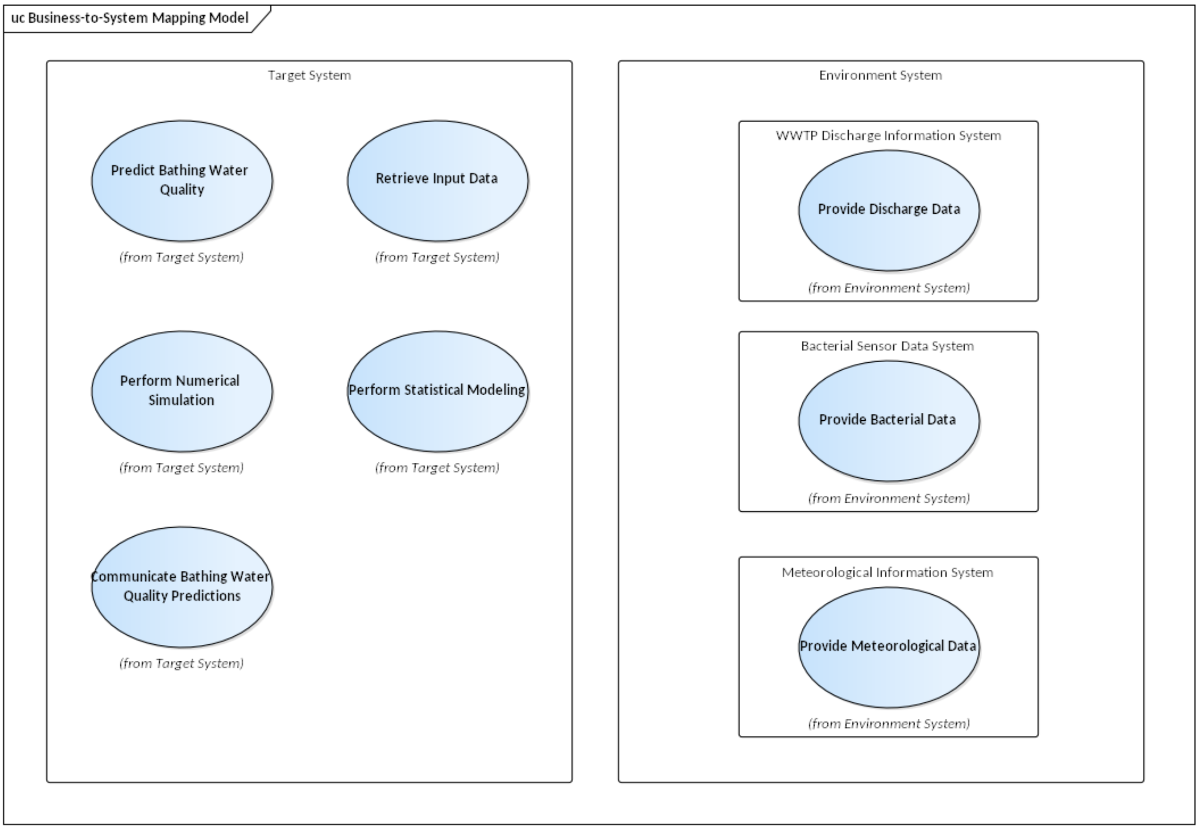


Figure 11. System Mapping Model for Paris

3.2.2. Components View

Based on the description of the context view in the previous section, this section will describe a set of software components needed to realise the implementation in Paris (System Decomposition Model), as well as an initial specification of data to be communicated via the component interfaces.

System Decomposition Model

The target system components to be developed in Paris is illustrated in Figure 12. These system components include the early warning system for bathing water quality (EWS-BWQ), the Expert App that provides processed information about the water quality at the bathing sites to the bathing site managers, and the Public App which receives their decisions concerning status of the bathing sites as well as other relevant data (e.g., water temperature and affluence). The environment systems are represented by the WWTP information system that provides discharge data, the Weather Data System that provides relevant meteorological data, and the Bacterial Sensor System that reports data about the FIB concentration in the river.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

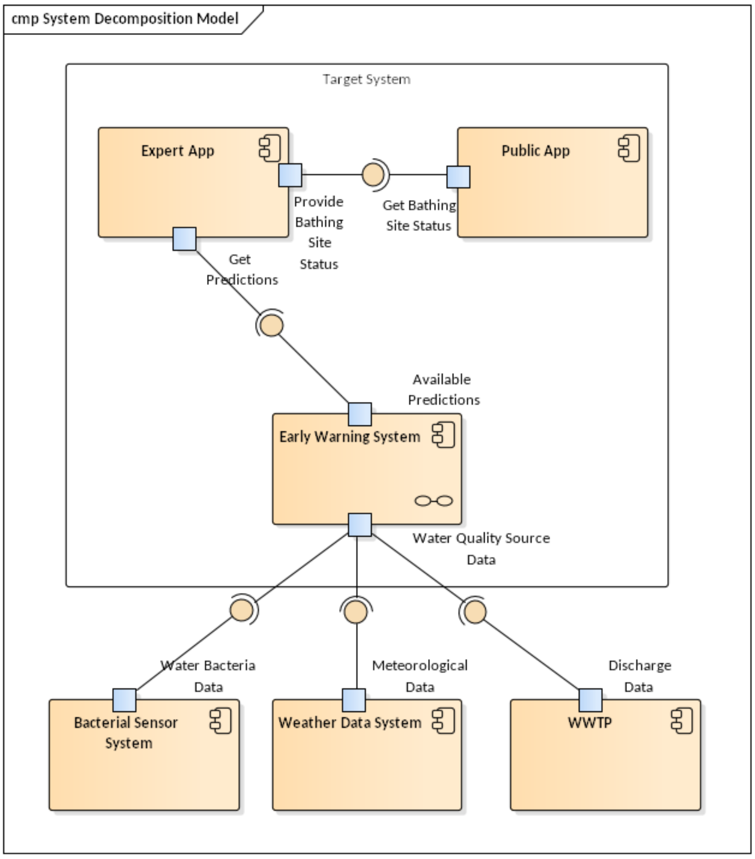


Figure 12. System Decomposition Model for Paris

As illustrated in Figure 13 the EWS-BWQ integrates two sub-components in order to predict the water quality at the bathing sites, namely the ProSe deterministic model and the Statistical Model.

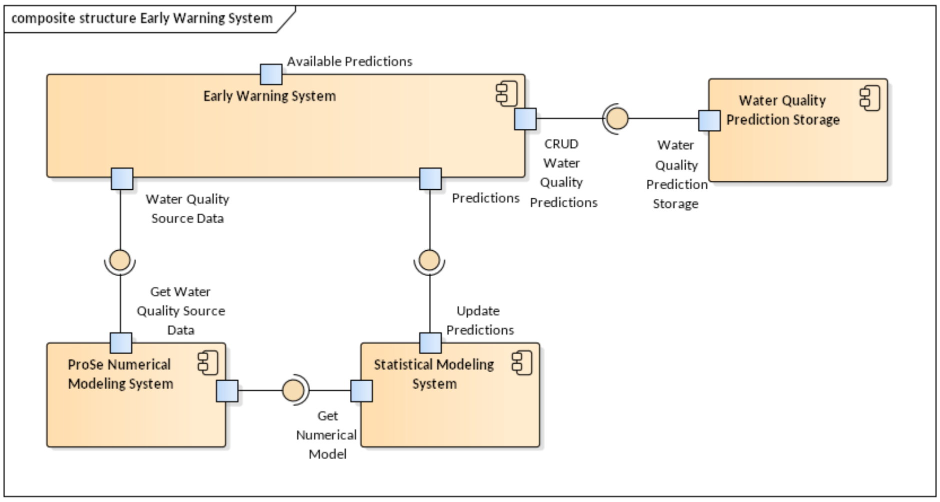


Figure 13. Detailed component view for Early Warning System

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

System Information Model

In this view the initial data requirements from the systems involved in the Paris case are analysed and an aligned system information model is established. Looking at the interfaces of the System Decomposition Model in the previous section, the data requirements sections highlighted in yellow in Figure 14 are relevant and will be described in the following.

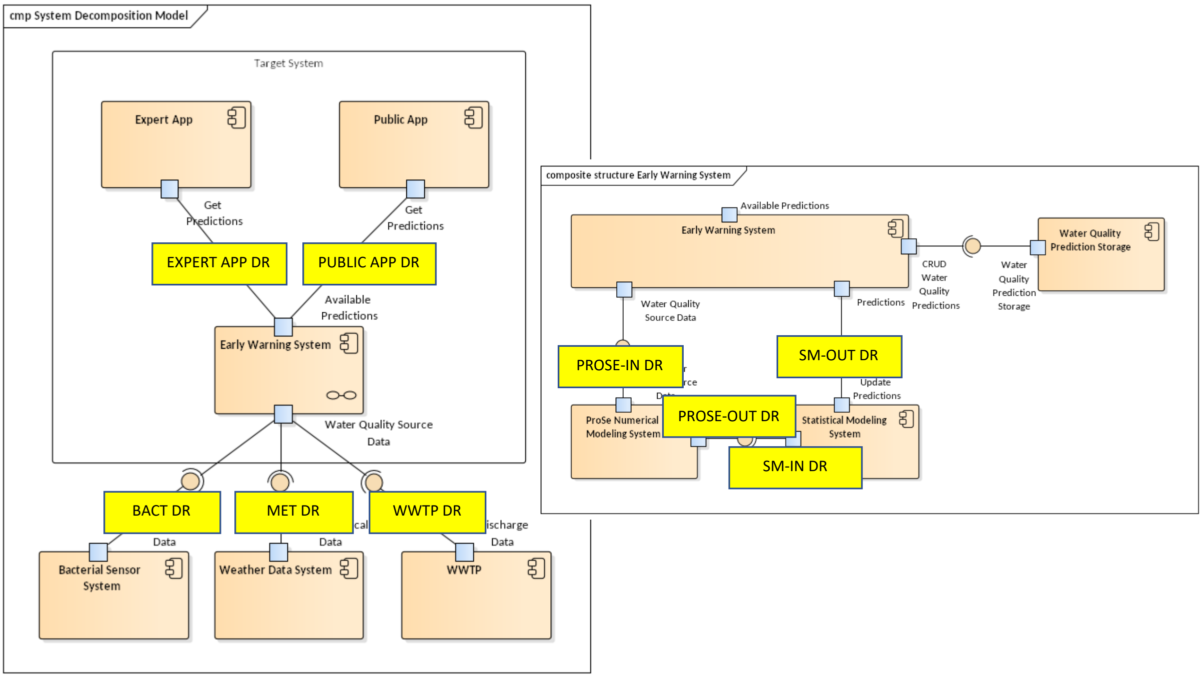


Figure 14. Data Requirements Overview for the Paris case

The tables below describe the data requirements for each of these interfaces.

Table 16. [PROSE-IN] ProSe input data requirements

Data Requirement	Datatype	Definition
River Flow	Water Measurement Entity	Describe flowrates at a localisation.
NH4	Sensor Entity (enum)	Describe NH4 (Ammonium Cation) at a localisation. The enumeration value (enum) should be accompanied by a measurement value expressed as a decimal value.
Free E.Coli	Sensor Entity (enum)	Describe the measurement of free E.Coli at a localisation. The enumeration value (enum) should be accompanied by a measurement value expressed as a decimal value.
Fixed E.Coli	Sensor Entity (enum)	Describe the measurement of fixed E.Coli at a localisation. In this case the E.Coli is fixed on or in other particles. The enumeration value (enum) should be accompanied by a measurement value expressed as a decimal value.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Location	Location Entity	Specification of the geographical location for which a measurement is made. In this case the
Time	Time Entity	Timestamp for when a particular measurement (river flow or bacteria measurement) is made.

Table 17. [PROSE-OUT] ProSe output data requirements

Data Requirement	Datatype	Definition
River Flow	Water Measurement Entity	Describe flowrates at a localisation.
NH4	Sensor Measurement Entity (enum)	Describe NH4 (Ammonium Cation) at a localisation.
Free E.Coli	Sensor Measurement Entity (enum)	Describe the measurement of free E.Coli at a localisation.
Fixed E.Coli	Sensor Measurement Entity (enum)	Describe the measurement of fixed E.Coli at a localisation. In this case the E.Coli is fixed on or in other particles.
Location	Location Entity	Specification of the geographical location for which a measurement is made. In the output the kilometric point is used to designate a location.
Time	Time Entity	Timestamp for when a particular measurement (river flow or bacteria measurement) is made.

Table 18. [SM-IN DR] Statistical Model input data requirements

Data Requirement	Datatype	Definition
Rainfall	Meteorological Data Entity (enum)	Describes the amount of rainfall at a localisation.
Discharge Data from WWTP	**See WWTP DR in** Table 20.	Describes various discharge data provided by the WWTP. This includes river flow, water level and amount of discharged water.
Location	Location Entity	Specification of the geographical location for which a measurement is made.
Time	Time Entity	Timestamp for when a particular measurement (river flow or bacteria measurement) is made. Should include a time zone and UTC offset.

Table 19. [SM-OUT] Statistical Model output data requirements

Data Requirement	Datatype	Definition
Prediction	Prediction Entity	A prediction that states the water quality at a certain location. The prediction values are: ‘excellent’, ‘good’, ‘sufficient’ or ‘poor’. Should include a location and time. Important that different percentiles of prediction interval can be described (2.5, 50, 90, 95, 97.5).

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Table 20. [WWTP DR] WWTP Discharge Data data requirements

Data Requirement	Datatype	Definition
River Flow	Water Measurement Entity	Describe flowrates at a localisation in the river.
Water Level in River	Water Measurement Entity	Describe the water level (height) at a localisation in the river.
Amount of Discharged Water	Water Measurement Entity	Describe the amount of discharged water at a localisation in the river. Should probably include both a timestamp and period.
Device responsible for a measurement	Device Entity	Describe which device or equipment (e.g., sensor or gauging point / meter) responsible for a measurement.
Gauging Point / Meter	Gauging Point / Meter	Describe a gauging point at the WWTP (subclass of Device Entity)

Table 21. [MET DR] Meteorological Data data requirements

Data Requirement	Datatype	Definition
Rainfall	Meteorological Data Entity (enum)	Rainfall should be one of multiple values in an enumeration list.

Table 22. [BACT DR] Water Bacteria Data data requirements

Data Requirement	Datatype	Definition
Fixed E.Coli	Sensor Measurement Entity (enum)	Fixed E.Coli should be one of multiple values in an enumeration list.
Free E.Coli	Sensor Measurement Entity (enum)	Free E.Coli should be one of multiple values in an enumeration list.
NH4	Sensor Measurement Entity (enum)	NH4 (Ammonium Cation) should be one of multiple values in an enumeration list.
Sensor characteristics	Sensor Entity	Describes various characteristics associated with a sensor (battery level, network signal, sensor model, etc.). Subclass of Device Entity.

Table 23. [EXPERT APP DR] Expert App data requirements

Data Requirement	Datatype	Definition
To be determined after CoP meetings, probably early 2021…		

Table 24. [PUBLIC APP DR] Public App data requirements

Data Requirement	Datatype	Definition
To be determined after CoP meetings, probably early 2021…		

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

4. Design Requirements for Semantic Model Development

This chapter describes a further elaboration of the data entities identified in the previous chapter for the Paris and Milan cases in DWC. The chapter begins with explaining how the work described in this report fits in to the engineering approach that will be used for the development of a semantic model in DWC. The requirements collection process described in the previous chapter represents the first stage of the semantic model development approach, namely "Requirements Collection and Analysis". The second stage of the semantic model development approach, the "Design and Conceptualisation" stage, will be partly covered in this chapter as we further define the entities identified from Milan and Paris. Here, the data requirements collected from the Milan and Paris cases are mapped to existing data models, primarily the "Smart Data Models"[3].

4.1. Overall approach for Semantic Model Development

One of the objectives of this work is to derive data requirements from which a more complete and more formalised semantic model will be developed in task 4.3.2. Figure 15 illustrates the methodology that will be used for the semantic model development and how the identification of design requirements as described in this report initiates this. A short introduction to semantic models / ontologies along with a description of how they could be applied in the realm of DWC is included in Annex A (for the interested reader).

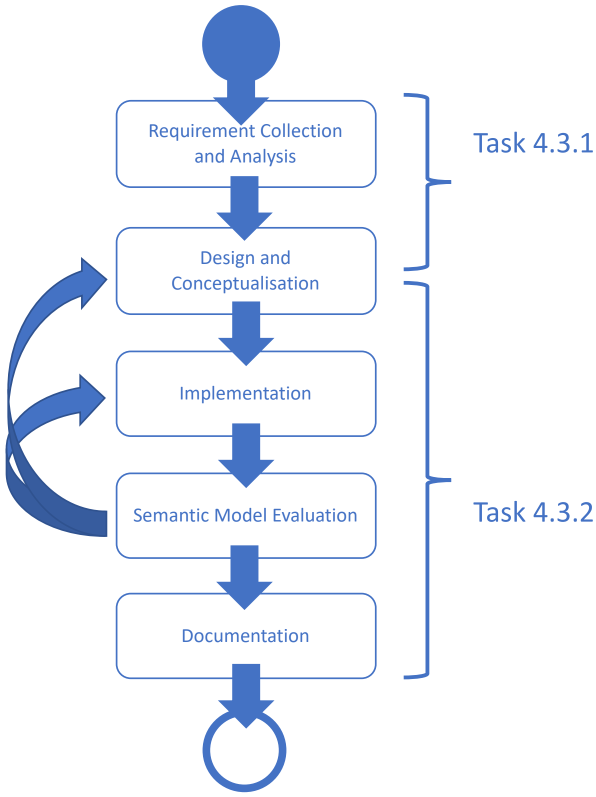


Figure 15. Overall approach for Semantic Model Development in DWC

[3] https://smartdatamodels.org (last accessed 07.12.2020)

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

With respect to the ontology engineering involved when developing a common semantic model for DWC, a methodology for the development of the DWC semantic model[4] is tailored from an analysis of existing ontology engineering methodologies. These existing engineering methodologies are: Methontology [2], NeOn [3], 101 [4], SAMOD [5], UPON [6], UPON Lite [7]. Our analysis has shown that some of these methodologies are too complex and resource-demanding for our task (e.g., Methontology and NeOn), while others are too simplistic (e.g., SAMOD and UPON Lite). Therefore, we have extracted what we consider the most relevant ontology engineering activities considering the scope of the semantic model development in DWC. These stages are described in Table 25. The activities that are included in this deliverable are described in black font while the steps that will be taken as part of task 4.3.2 are described in grey font.

Table 25. Stages in Semantic Model Development Approach

Stage	Description
Requirements Collection and Analysis	This stage begins by defining the domain of interest, the scope and purpose of the ontology to be developed. Furthermore, elicited stakeholder requirements will be described as use case scenarios that will lead to the identification of relevant concepts and properties in the ontology. These use cases scenarios are captured and represented according to the ARCADE Framework specification in Chapter 3.
Design and Conceptualisation	Starting from the Requirements Collection and Analysis in the previous stage as well as the System Information Model (Components View) described in Chapter 3, the "raw" set of entities (concepts) resulting from the previous stage is elaborated. Classes and their properties are defined with a textual description, the subsumption hierarchy of classes is established, any semantic rules (e.g., property restrictions) are defined, and properties are distinguished as object- or data properties (literals). Once the relevant sets of classes and properties have been identified, an analysis of candidate existing ontological resources is performed.
Implementation	In this stage the ontology is formalised in an ontological language (OWL[5]). Relevant existing ontological resources should be re-used whenever possible and common ontology design patterns[6] should be applied. The result from this stage is a complete ontology encompassing all concepts, properties, and semantic rules defined in the preceding stages.
Semantic Model Evaluation	In the evaluation the complete ontology is evaluated manually and automatically. The manual evaluation should involve an examination of the ontology both by the ontology engineers and domain experts to ensure sufficient coverage, relevance, and correctness. The automatic evaluation involves consistency checks through the use of reasoning services and querying the ontology using a set of competency questions. The latter can be performed by issuing SPARQL[7] queries against the ontology, e.g., in the Protégé editor. If the evaluation is unsatisfactory, changes must be made in the Implementation stage and/or the Design and Conceptualisation stage.

[4] In the document we use the terms 'semantic model' and 'ontology' interchangeably.

[5] https://www.w3.org/TR/owl2-overview/

[6] http://ontologydesignpatterns.org

[7] https://www.w3.org/TR/sparql11-overview/

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Documentation	Finally, once the ontology has passed the evaluation, the ontology is made available in a persistent URL along with accompanying documentation. The documentation should in addition to appropriate definitions of the ontology constructs (classes and properties) include relevant details from each stage of the development process.

The scope of this deliverable (D4.4) is to elicit and formulate the design requirements. This encompasses the first and partly the second stage of the methodology: Requirements Collection and Analysis, and Design and Conceptualisation, as indicated in Figure 15. The remaining stages of the methodology will be reported in D4.5 which describes how the design requirements are implemented in a semantic model, and how this semantic model is evaluated and documented.

4.2. Design and conceptualisation of entities in the Milan case

Following from the initial requirements collection described in Chapter 3.1, this stage includes the definition of relevant concepts the semantic model should include.

Relevant concepts and properties

Table 26 describes the set of relevant concepts (entities) derived from the scenarios described in Chapter 3 for the Milan case.

Table 26. Relevant entities derived from scenario description in Milan

Entity	Description
WWTP Quality Measurement Entity	An entity for describing various water quality measurements from sensors at the WWTP.
WWTP Quantity Measurement Entity	Should include parameters for describing water flow and water level at the WWTP.
Soil Sensor Measurement Entity	An entity for describing various measurements from sensors out in the agricultural fields (soil measurements).
Status Entity	Should describe the status of device (electromechanical equipment).
Location Entity	A generic entity for describing the localization of measurements. Should consider topological description of a location (e.g., inside the WWTP).
Time Entity	A generic entity for describing various time parameters (reported time, measured time, validity period) related to measurements.
Alert Entity	An entity to describe alerts sent from the EWS-SWR as well as within the WWTP.
Meteorological Data Entity	An entity for describing meteorological measurements (nowcasts and forecasts) such as rainfall, temperature, wind speed, wind direction, humidity, solar radiation, etc.
Device Entity	An entity for describing physical device used for making measurements. Should likely be represented as a generalisation of Sensor Entity and Meter (Gauge) Entity which may have both equivalent and different attributes/properties.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Sensor Entity	Describes characteristics related to a sensor device, such as battery level, network signal, brand, model, etc.
Meter Entity	Describes characteristics related to a meter device. Synonym term is gauge.

Table 27 maps the data requirements specified above to existing data models.

Table 27. Mapping data requirements from Milan to existing data entity models

Entity	Description
WWTP Water Quality Measurement Entity	**Water Quality Model** Description: This model represents different water quality parameters. Link to specification: https://github.com/smart-data-models/dataModel.Environment/tree/master/WaterQualityObserved
WWTP Water Quantity Measurement Entity	**Water Observed Model** Description: This model represents parameters such as flow, level and volume of water observed. Link to specification: https://github.com/smart-data-models/dataModel.Environment/tree/master/WaterObserved
Soil Sensor Measurement Entity	**Agri Parcel Record Model** Description: Contains a harmonised description of the conditions recorded on a parcel of land. Link to specification: https://github.com/smart-data-models/dataModel.Agrifood/blob/master/AgriParcelRecord/README.md
Status Entity	**Device Entity** Description: The status (state) of a device is an attribute (deviceState) in the Device Entity. Link to specification of Device: https://github.com/smart-data-models/dataModel.Device/blob/master/DeviceModel/README.md
Location Entity	Location is described using address (as described in schema.org) and/or geographical coordinates using GeoJSON. GeoJSON is a format for encoding different features of geographic data structures such as point and polygons. TopoJSON is an extension to GeoJSON that encodes topology. However, whether this satisfies the requirements in Milan must be investigated further.
Time Entity	The time specification of FIWARE is specified as an attribute according to the ISO 8601 standard. This means that a dateTime is specified as <date><time><timezone>, where <timezone> is relative to UTC, e.g., 2020-11-20 14:30:30+01.
Alert Entity	**Alert Model** Description: Describes an alert that could be used to send alerts related to traffic jam, accidents, weather conditions, high level of pollutants, etc. Link to specification: https://github.com/smart-data-models/dataModel.Alert/tree/master/Alert

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Meteorological Data Entity	**Weather Observed Model** Description: The weather observed model describes observed weather measurements (nowcasts) and includes parameters such as precipitation (rainfall), temperature, wind direction and windspeed. Link to specification: https://github.com/smart-data-models/dataModel.Weather/tree/master/WeatherObserved
Device Entity Sensor Entity Meter Entity	**Device and Device Model** Description: Describes the static properties of a device. The property category of Device Model allows for specifying whether the device is a sensor, a meter, or another type of device. Link to specification of Device: https://github.com/smart-data-models/dataModel.Device/blob/master/DeviceModel/README.md Link to specification of Device Model: https://github.com/smart-data-models/dataModel.Device/blob/master/DeviceModel/doc/spec.md

4.3. Design and conceptualisation of entities for the Paris case

Following from the initial requirements collection described in Chapter 3.2, this stage includes defining relevant concepts the semantic model should include.

Relevant concepts and properties

Table 28 describes the set of relevant concepts (entities) derived from the scenarios described in Chapter 3 for the Paris case.

Table 28. Relevant entities derived from scenario description in Paris

Entity	Description
Water Quality Measurement Entity	Should include parameters for describing various bacteria measurements, such as NH4, Free E.Coli and Fixed E.Coli.
Water Measurement Entity	Should include parameters for describing river flow, river water level (height), volume/amount of discharged water and stormwater overflows.
Device Entity	An entity for describing physical device used for making measurements. Should likely be represented as a generalisation of Sensor Entity and Meter (Gauge) Entity which may have both equivalent and different attributes/properties.
Sensor Entity	Describes characteristics related to a sensor device, such as battery level, network signal, brand, model, etc.
Gauge Entity	Describes characteristics related to a gauge device. Synonym term is meter.
Location Entity	A generic entity for describing the localization of measurements. Should include the possibility of specifying the kilometric point of a location.
Time Entity	A generic entity for describing various time parameters (reported time, measured time, validity period) related to measurements. Should include internationalization attributes (UTC offset and time zone).

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

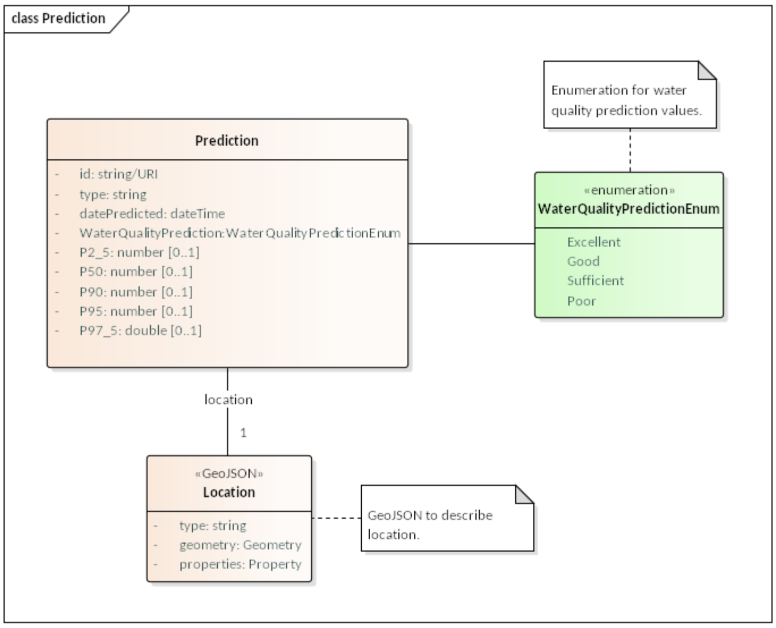
Meteorological Data Entity	Describes meteorological / weather measurements, primarily rainfall in the Paris case.
Prediction Entity	Describes the predicted water quality at bathing sites as predicted by the EWS-BWQ. Support for describing different percentiles of prediction interval should be included (2.5, 50, 90, 95, 97.5).

Table 29 describes a mapping between data requirements from the Paris case and existing data models.

Table 29. Mapping data requirements from Paris to existing data models

Entity	Existing Data Model
Water Quality Measurement Entity	**Water Quality Model** Description: This model represents different water quality parameters. Link to specification: https://github.com/smart-data-models/dataModel.Environment/tree/master/WaterQualityObserved
Water Measurement Entity	**Water Observed Model** Description: This model represents parameters such as flow, level and volume of water observed. Link to specification: https://github.com/smart-data-models/dataModel.Environment/tree/master/WaterObserved
Device Entity Sensor Entity Gauge Entity	**Device and Device Model** Description: Describes the static properties of a device. The property category of Device Model allows for specifying whether the device is a sensor, a meter, or another type of device. Link to specification of Device: https://github.com/smart-data-models/dataModel.Device/blob/master/DeviceModel/README.md Link to specification of Device Model: https://github.com/smart-data-models/dataModel.Device/blob/master/DeviceModel/doc/spec.md
Location Entity	Location is described using address (as described in schema.org) and/or geographical coordinates using GeoJSON. GeoJSON is a format for encoding different types of geographic data structures such as points and polygons. A kilometric point specifies the distance in kilometers from a specified point. Possibly the distance in kilometers can be specified as a property associated with a point (expressed as latitude-longitude coordinates).
Time Entity	The time specification of FIWARE is specified as an attribute according to the ISO 8601 standard. This means that a dateTime is specified as <date><time><timezone>, where <timezone> is relative to UTC, e.g., 2020-11-20 14:30:30+01.
Meteorological Data Entity	**Weather Observed Model** Description: The weather observed model describes observed weather measurements (nowcasts) and includes parameters such as precipitation (rainfall), temperature, wind direction and windspeed.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

	Link to specification: https://github.com/smart-data-models/dataModel.Weather/tree/master/WeatherObserved
Prediction Entity	The prediction will likely require a new entity, such as the following: 

digital-water.city
digitalwater_eu
digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

5. Design Requirements for Interoperability Middleware Architecture

Based on interaction with the consortia from Milan and Paris this section sketches the system architectures along with inherent technical components necessary for realising the technical implementations in Milan and Paris. The DWC project is collaborating with other water related EU projects on reusing data models and re-usable components of the FIWARE framework[8]. This chapter therefore begins with a brief introduction to the most relevant concepts from the FIWARE framework before architectural sketches on how FIWARE can be applied in the setting of the Milan and Paris case are presented.

5.1. Introduction to FIWARE

FIWARE is a framework of open-sourced software components targeted towards digitalisation and smart application of data across multiple application domains. The focal point of FIWARE is interoperable solutions for context management. This includes the ability to source data from measurement devices (e.g., sensors), represent these source data in a wider context representation, and provide the means for accessing these context data by end-user applications. The five architectural perspectives of the framework are illustrated in Figure 16. In the following we will focus on the three in the middle, namely Core Context Management, Interface with IoT, and Context Processing, Analysis and Visualisation.

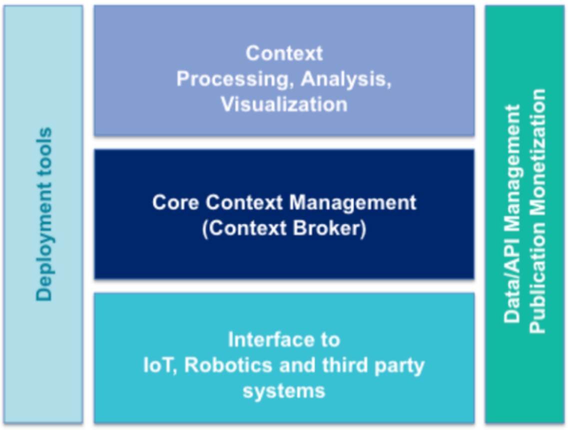


Figure 16. FIWARE Framework[9]

5.1.1. Core Context Management

The Core Context Management part of FIWARE represent the ability to produce, gather, publish and consume context data and turn this into actionable information to be applied by end user applications[10].

The context data are represented through values assigned to attributes that contribute to define the entities and are managed by a *Context Broker*. A Context Broker is a core and mandatory component

[8] https://www.fiware.org/

[9] Illustration taken from https://www.fiware.org/developers/

[10] https://fiwaretourguide.readthedocs.io/en/latest/core/introduction/).

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

of the FIWARE framework that allows for storing, updating and subscribing to the entities representing the context via a standardised REST API (NGSIv2 or NGSI-LD as described below). Orion[11] is a Context Broker that has been released by FIWARE. Orion provides the NGSI v2 API. In addition, the following Context Brokers providing the NGSI-LD API are under incubation: Orion-LD Context Broker[12], the Scorpio Broker[13], and the Stellio Context Broker[14].

In addition to the Context Broker the Core Context Management also include Generic Enablers[15] that enable to store context data persistently, such as STH-Comet[16], Cygnus[17], Draco[18] and QuantumLeap[19].

The context itself is defined by means of the FIWARE NGSI[20] API. NGSI defines a data model for describing context information; a context data interface for exchanging information via queries, subscriptions and updates; and a context availability interface for exchanging information on how to obtain context information. There are basically two NGSI versions that are relevant, NGSIv2 and NGSI-LD. The main elements in the NGSIv2 model are entities, attributes and metadata, as shown in Figure 17. An entity represents a physical or logical object (e.g., a sensor, a person, an issue in a ticketing system) defined by an identifier and a type definition, an attribute represents some property of the entity (e.g., a measurement value), while the metadata describes additional "data about the data", such as the accuracy of the measurement value.

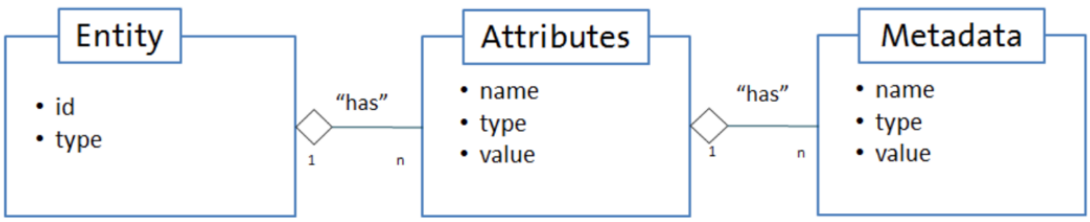


Figure 17. NGSIv2[21]

The other version of NGSI, NGSI-LD, where LD stands for Linked Data, has a different underlying data model than NGSIv2. Here, data are described in triples in a subject-predicate-object pattern resulting in a graph representation of the context data. Furthermore, in NGSI-LD the ID of an entity should be a Uniform Resource Identifier (URI) ensuring a consistent representation of the identifier of an entity.

---

11 https://fiware-orion.readthedocs.io/en/master/#welcome-to-orion-context-broker
12 https://github.com/FIWARE/context.Orion-LD
13 https://github.com/ScorpioBroker/ScorpioBroker
14 https://github.com/stellio-hub/stellio-context-broker
15 A Generic Enabler is a component that is considered general purpose and independent of any particular usage area.
16 https://github.com/telefonicaid/fiware-sth-comet
17 https://fiware-cygnus.readthedocs.io/en/latest/
18 https://github.com/ging/fiware-draco
19 https://github.com/smartsdk/ngsi-timeseries-api
20 https://fiware.github.io/specifications/ngsiv2/stable/
21 Illustration taken from https://fiware.github.io/specifications/ngsiv2/stable/

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Figure 18 shows the underlying model of NGSI-LD. As the figure shows, there is no metadata element, and the Attributes element in NGSIv2 now refers to either Property or Relationship where the former represents literal values (strings, decimals, etc.) while the latter represent relationships between different entities.

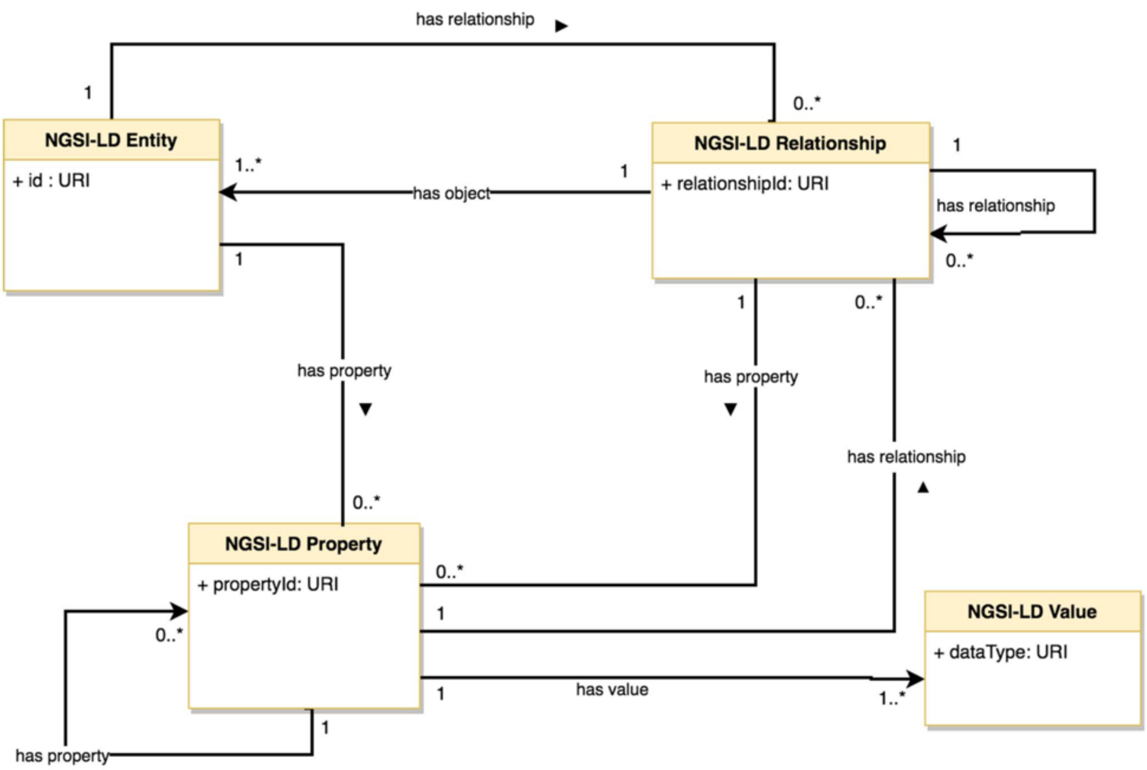


Figure 18. NGSI-LD[22]

Figure 19 and Figure 20 illustrate how data are formatted using NGSI v2[23] and NGSI-LD[24]. These JSON snippets, which both represent an extract of the WeatherObserved data model[25], shows how temperature and precipitation is represented along with a timestamp and information about the weather station providing the measurements.

As these examples show, there are some notable differences in how data are represented. First of all, NGSIv2 is represented as basic JSON format[26], while NGSI-LD is represented using JSON-LD[27]. Further, in NGSI-LD the ID shall be represented using a URI, not a simple string value as in NGSIv2. Each attribute in NGSI-LD shall contain two fields, a property and a value, whereas in NGSIv2 an attribute can be

[22] Illustration taken from https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf

[23] https://fiware.github.io/specifications/ngsiv2/stable/

[24] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf

[25] See https://github.com/smart-data-models/dataModel.Weather/blob/master/WeatherObserved/README.md

[26] https://www.json.org/json-en.html

[27] https://json-ld.org/

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

represented by just a value. In NGSI-LD a context element is added to provide fully qualified names (URIs) associated to terms. This is similar to how namespaces are used in XML.


```
{
  "id": "Spain-WeatherObserved-Valladolid-123",
  "type": "WeatherObserved",
  "dateObserved": {
    "type": "DateTime",
    "value": "2016-11-30T07:00:00.00Z"
  },
  "temperature": {
    "value": 3.3
  },
  "precipitation": {
    "value": 0
  },
  "source": {
    "value": "http://www.aemet.es"
  },
  "location": {
    "type": "geo:json",
    "value": {
      "type": "Point",
      "coordinates": [-4.754444444, 41.640833333]
    }
  },
  "stationCode": {
    "value": "2422"
  }
}
```


Figure 19. NGSIv2 example


```
{
    "id": "urn:ngsi-ld:WeatherObserved:Spain-WeatherObserved-Valladolid-123",
    "type": "WeatherObserved",
    "dateObserved": {
        "type": "Property",
        "value": {
            "@type": "DateTime",
            "@value": "2016-11-30T07:00:00.00Z"
        }
    },
    "temperature": {
        "type": "Property",
        "value": 3.3
    },
    "precipitation": {
        "type": "Property",
        "value": 0
    },
    "source": {
        "type": "Property",
        "value": "http://www.aemet.es"
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                -4.754444444,
                41.640833333
            ]
        }
    },
    "stationCode": {
        "type": "Property",
        "value": "2422"
    },
    "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 20. NGSI-LD example

5.1.2. Interface with IoT

In order to interface with devices and systems providing context data to the Context Broker and its NGSI API, the IDAS Generic Enabler provides a set of IoT Agents supporting different IoT protocols. Currently IoT Agents for the following protocols are provided: JSON (over HTTP/MQTT), Lightweight M2M (LWM2M), Ultralight, LoRaWAN, OPC-UA and Sigfox.

5.1.3. Context Processing, Analysis and Visualisation

This architectural layer provides Generic Enablers that aim to enable processing, analysis and visualisation of context information. Examples of Generic Enablers in this area are: Wirecloud[28] for

[28] https://wirecloud.readthedocs.io/en/stable/

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

visualisation of integrated data, Knowage[29] for business analytics and Kurento[30] for real-time processing of media streams.

5.2. Interoperability Middleware Architecture for Milan

5.2.1. Realisation View

Figure 21 sketches the overall system architecture in the Milan case. The colour coding indicates digital solutions to be developed in DWC (target systems) in blue, environment systems in yellow, reusable components from FIWARE in beige, custom developments required in pink and other generic software components such as databases in grey.

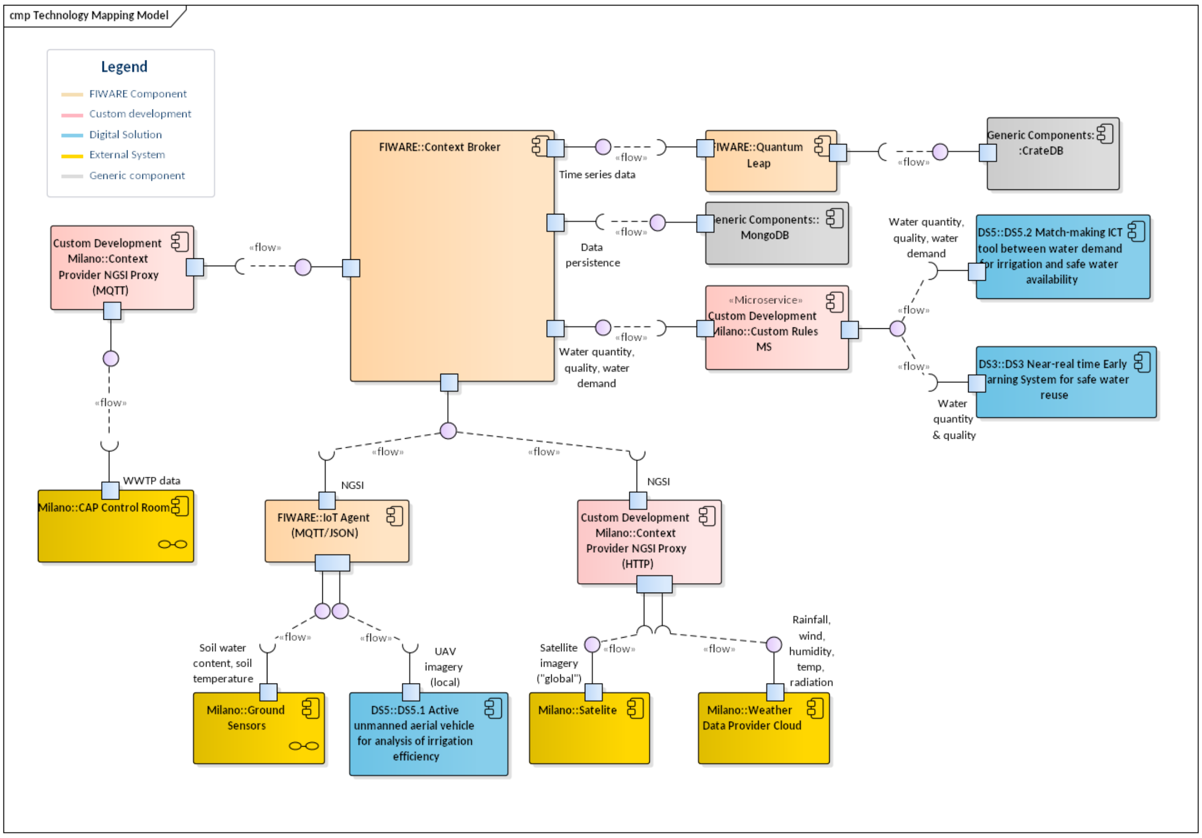


Figure 21. Middleware Architecture in Milan

Figure 21 shows the internal components structure at the CAP Control Room and how this system interacts with various sensors at the WWTP (illustrated in Figure 23) and the Alert system specifically.

[29] https://knowage.readthedocs.io/en/latest/
[30] https://kurento.readthedocs.io/en/stable/

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

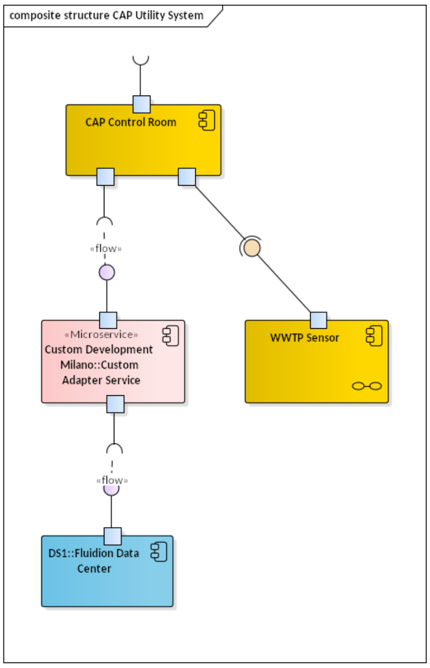


Figure 22. Component structure internally at CAP Control Room

Figure 23 illustrates a decomposition of different sensors at the WWTP.

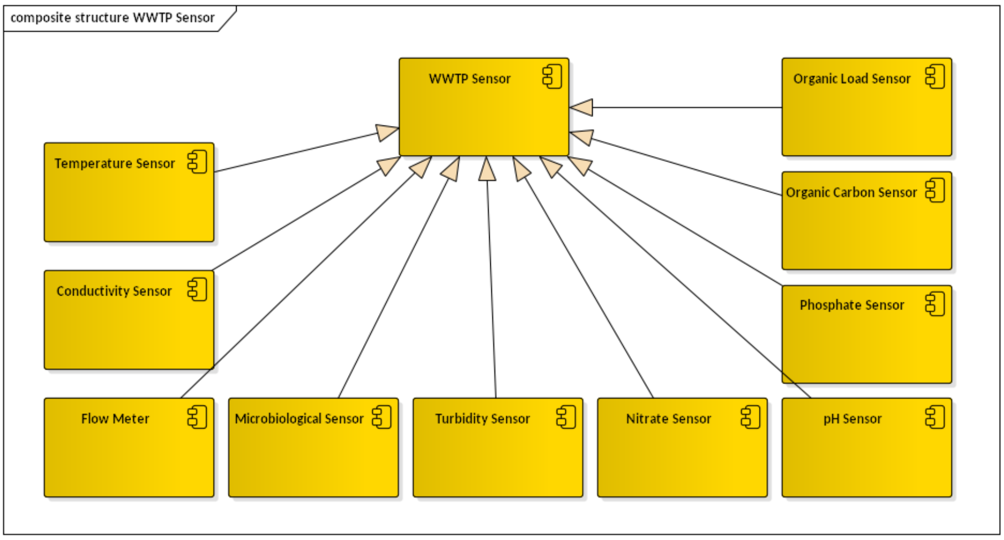


Figure 23. Sensors at the WWTP

5.2.2. Summary of required software components in Milan

Table 30 describes relevant FIWARE components and custom developments required for the Milan case.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Table 30. Software components to be realised in Milan

Software Component	Component name and description
FIWARE Components	
Context Broker	• Orion Context Broker (initially)
NGSI Version	• NGSI v2 (initially)
Type of data storage / persistence solution	• CrateDB with QuantumLeap as Generic Enabler
IoT Agents	• IoTAgent (MQTT) for interaction with AG sensors and UAV (DS6)
Custom Developed Components	
Custom development required	• NGSI Proxy (MQTT) for interaction with the CAP Control Room • NGSI Proxy between Satellite Data Provider and the Context Broker. • NGSI Proxy between Weather Data Provider and the Context Broker. • Adapter service between Fluidion´s Alert System and CAP Control Room • Microservice with custom rules between the Context Broker and the Matchmaking System + the EWS-SWR.

5.3. Interoperability Middleware Architecture for Paris

5.3.1. Realisation View

Figure 24 sketches the overall system architecture in the Paris case. The colour coding indicates digital solutions to be developed in DWC (target systems) in blue, environment systems in yellow, reusable components from FIWARE in beige, custom developments required in pink and other generic software components such as databases in grey. Note that while the more conceptual system decomposition view in Section 3.2.2 included an interface between the Expert App and the Public App (DS18), this realisation view suggests that status data from the Expert App are rather stored in the Context Broker and retrieved by the Public App.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

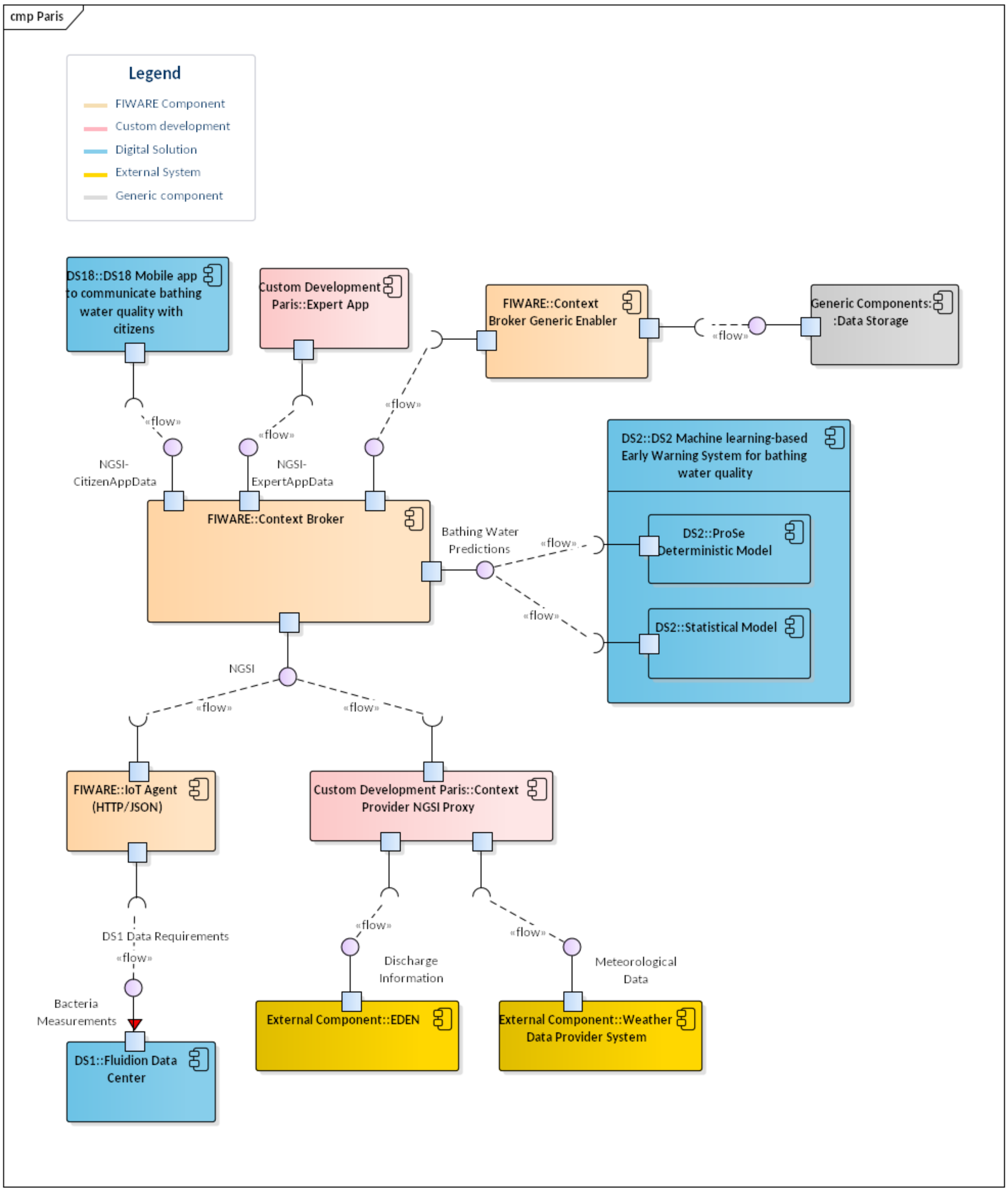


Figure 24. Middleware architecture in Paris

5.3.2. Summary of FIWARE components in Paris

Table 31. Software components to be realised in Paris

Software Component	Component name and description
FIWARE Components	
Context Broker	• Orion-LD Context Broker
NGSI Version	• NGSI-LD

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Type of data storage / persistence solution	• MongoDB
IoT Agents	• IoTAgent (JSON over HTTPS) for interaction with the Alert System.
Custom Software Components	
Custom development required	• NGSI Proxy for interaction with the WWTP (EDEN/MAGES) system (protocol to be defined). • NGSI Proxy for interaction with external weather data providers (protocol to be defined).

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

6. Conclusions and Further Work

This report described a requirements collection process for the Milan and Paris cases in the DWC project. From scenario descriptions design requirements with respect to which data will be communicated between which system components have been elicited and defined. The requirements collection process has been supported by an architectural description framework named ARCADE. This approach guides a generic description of requirements that can support reuse of the specified material in other settings, both within and outside the DWC project. Furthermore, the report describes a mapping of data requirements to existing data models in the water domain and beyond. The report also describes a mapping between the generic software components identified as relevant for the Milan and Paris cases and FIWARE.

The work described will be continued in task 4.3.2 where a common semantic model for DWC will be developed, and in task 4.3.3 where a common semantic interoperability middleware architecture will be developed.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

7. References

[1] H. Schwarzmüller, A. Vennesland, P. Halland Haro, and G. Bour, 'D4.1 Interoperable and secure flow of information: Cyber-physical sphere and interoperability aspects in the utilities regarding the DWC solutions', Digital-Water.City, Jun. 2020.

[2] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, 'Methontology: from ontological art towards ontological engineering', 1997.

[3] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, 'The NeOn methodology for ontology engineering', in *Ontology engineering in a networked world*, Springer, 2012, pp. 9–34.

[4] N. F. Noy, D. L. McGuinness, and others, *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and …, 2001.

[5] S. Peroni, 'A simplified agile methodology for ontology development', in *OWL: Experiences and Directions–Reasoner Evaluation*, Springer, 2016, pp. 55–69.

[6] A. De Nicola, M. Missikoff, and R. Navigli, 'A proposal for a unified process for ontology building: UPON', in *International Conference on Database and Expert Systems Applications*, 2005, pp. 655–664.

[7] A. De Nicola and M. Missikoff, 'A lightweight methodology for rapid ontology engineering', *Commun. ACM*, vol. 59, no. 3, pp. 79–86, 2016.

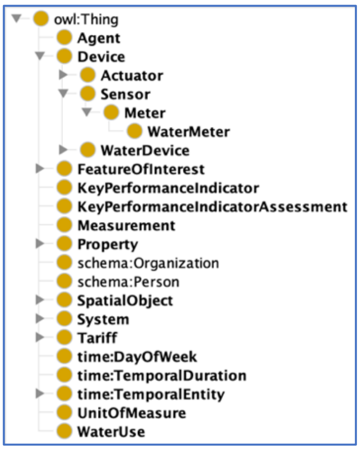
[8] S. Staab and R. Studer, *Handbook on ontologies*. Springer Science & Business Media, 2010.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Annex A: Introduction to semantic models / ontologies

This section provides a minimal and practical description of some key aspects related to ontologies to prepare for the remainder of this report. For a more detailed explanation of ontologies and their application, the reader is referred to e.g., the book Handbook on Ontologies [8] and the W3C Recommendation on the OWL 2 Web Ontology Language[31].

An ontology is a formal definition of the concepts, properties and interrelationships of the entities that exist in some domain of discourse. It provides a shared vocabulary that can be used to describe the domain, classifying and categorising the elements contained within it. Typically, an ontology is formalised using the Web Ontology Language (OWL). OWL is a part of the W3C suite of Semantic Web standards[32], which includes among others Resource Description Format (RDF)[33], a framework for representing web data using subject-predicate-object triples, and the Resource Description Format Schema (RDFS)[34] which provides a data-modelling vocabulary for RDF data. While both OWL and RDFS offer a vocabulary for describing RDF data, OWL allows for greater expressibility than RDFS.



In an ontology, classes represent sets of individuals (also called instances or objects) with similar characteristics and are organised in a specialisation hierarchy (this hierarchy is also called a *subsumption hierarchy).* This is illustrated in the figure to the right which depicts the specialisation hierarchy of classes in the SAREF4WATR ontology[35] as represented in the ontology editor Protégé. Here, a `WaterMeter` is a subclass of (specialisation of) `Meter`, `Meter` is a subclass of `Sensor`, and `Sensor` is a subclass of `Device`. This also means that the individuals associated with a particular class are specialisations of those individuals belonging to classes higher in the specialisation hierarchy.

In addition to classes and individuals, ontologies also describe properties, of which there are two fundamental types: object properties and data properties. Object properties define relationships between individuals whereas data properties define literal values associated with individuals. For example, the object property `hasMeasurement` is a relationship that allows for stating various types of measurements of a particular water sample. In the example shown in Figure 25 a sample of water (here, ex:DTSample335632 is an individual of the class `Water`) has a certain concentration of cadmium and e.Coli. The object property `relatesToProperty` allows for defining different types of measurements. The data properties `hasTimestamp` and `hasValue` allows for defining the actual time of measurement and concentrations of cadmium and e.Coli in the water sample respectively, while the object property `isMeasuredIn` enables a definition of which unit of measurement is applied.

---

[31] https://www.w3.org/TR/owl2-overview/

[32] https://www.w3.org/standards/semanticweb/

[33] https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[34] https://www.w3.org/TR/rdf-schema/

[35] https://saref.etsi.org/extensions.html#SAREF4WATR

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

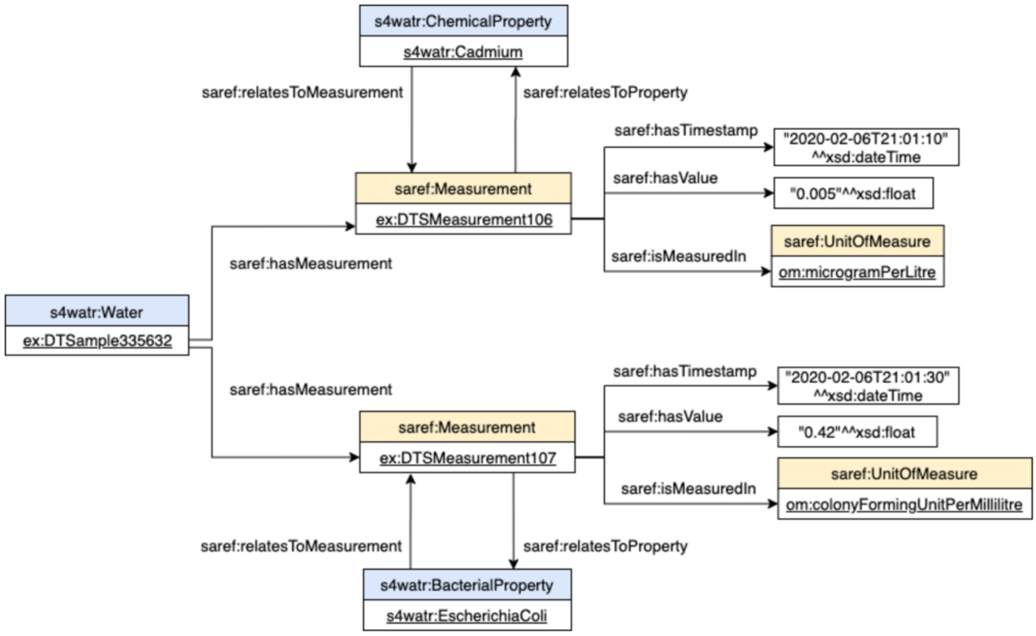


Figure 25. Example from the SAREF4WATR ontology[36]

As revealed by the figure there are two ontologies involved in this example, SAREF4WATER (using the s4watr prefix) and SAREF (using the saref prefix). SAREF is a suite of ontologies[37] where SAREF itself is the core ontology, while there are many extensions (modules) for different application domains. SAREF4WATR is the extension of SAREF for the water management domain. Using SAREF together with one or more of the extension ontologies allows for extending the scope, possibly integrating data from multiple domains into a single knowledge base.

An ontology can be used to uniformly define classes (types), properties (relationships and attributes) and axioms (semantic rules and assertions) of data entities in a knowledge base (aka triple store or knowledge graph). Here, data are described in the triple format (subject-predicate-object) such that according to the example in Figure 25 you would have the following three linked triples stating the measurement of cadmium in a water sample:

Subject	Predicate	Object
DTSample335632 (type Water)	hasMeasurement	DTSMeasurement106 (type Measurement)
DTSMeasurement106 (type Measurement)	relatesToProperty	Cadmium (type ChemicalProperty)
DTSMeasurement106 (type Measurement)	hasValue	0.005 (datatype float)

Provided that NGSI-LD is used as format for expressing entities in the context broker (e.g., Orion-LD) and associated data storage, quite powerful queries as well as learning techniques can exploit both the explicit (as in the example above) and latent semantics expressed in the ontology.

---

36 Illustration taken from the technical specification of SAREF4WATR, available from: https://www.etsi.org/deliver/etsi_ts/103400_103499/10341010/01.01.01_60/ts_10341010v010101p.pdf

37 An overview of the SAREF suite of ontologies is available at: https://saref.etsi.org/index.html

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

One example of using latent semantics from knowledge bases is knowledge graph embedding. In the works of Myklebust el al. (2019)[38] knowledge graph embedding techniques are used to model eco-toxological effects of various compounds in the water environment. The idea is that based on the known eco-toxological effects declared in the knowledge graph, the knowledge embedding model will compute/learn the probability of unknown eco-toxological effects. This is also known as link prediction. For example, the knowledge graph states that compound X affects (e.g., has a lethal effect) on species Y. How the compound X affects species Z is not known. But based on the learned vector positions of X, Y and Z, and the quantified effect (relationship) X has on Y, the model also predicts that X affects Z. The quantified effect (relationship) can for example be represented by some computed distance/offset between X and Y. The vector space representation of all entities in the knowledge graph is generated by a so-called knowledge graph embedding model (e.g., based on neural networks). The objective of these models is to learn an optimal vector representation for each entity in the knowledge graph and the intuition is that these vectors capture some latent (unexpressed) semantics from the context of each entity in the knowledge graph. Here, context is represented by for example the structural characteristics of the knowledge graph (e.g., which entities are neighbors to entity E in the graph) or ontological definitions (e.g., entity E is a member of the class Arsenic).

[38] Myklebust, Erik B., et al. "Knowledge graph embedding for ecotoxicological effect prediction." *International Semantic Web Conference*. Springer, Cham, 2019.

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Annex B: UML notation used in this report

UML Use Case Diagram

Use case diagrams specify *what* is the expected functionality, and there is little focus on *how* this functionality should be accomplished. The notation used in this report is illustrated in Figure 26 and described as follows:

- Use case: A function/behavior that the system should perform.
- Actor / Stakeholder: Someone (human, organisation, system) that interacts in some way with a given use case. The fact that an actor/stakeholder has an interaction with a use case is indicated by an *association*.
- Include relationship: Sometimes the functionality of a use case uses the functionality of another use case an *include* relationship indicates this.
- Generalisation: Specifies a relationship between more general / less general actors / stakeholders, e.g., that a Sensor is decomposed into a Temperature Sensor and a Soil Moisture Sensor.
- Multiple levels / composite level: A symbol resembling infinity indicates that a use case (or another UML entity) contains sub-structures that are revealed in another diagram.

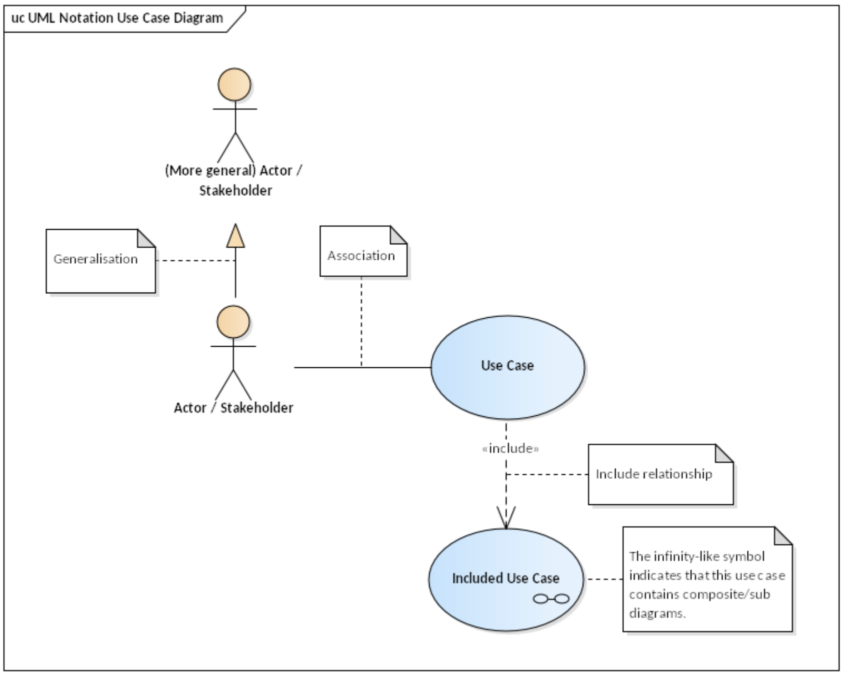


Figure 26. Use Case Diagram

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

UML Component Diagram

A component diagram is used to break down a system into its components and to visualise how these components interact. The notation used in this report is illustrated in Figure 27 and described as follows:

- Component: Represents a system or a modular part of a system.
- Port: A port specifies an interaction point with which a component can interact with its environment. Exposes provided or required interfaces of a component.
- Provided interface: Indicates an interface that the “hosting” component provides.
- Required interface: Indicates an interface that the “hosting” component requires (offered by the linked provided interface).
- Information Flow: Indicates that some data / information flows between two components. This association may link to a data model defining the data/information flow.
- Multiple levels / composite level: as with the use case, a component may also contain sub-structures that are revealed in another diagram.

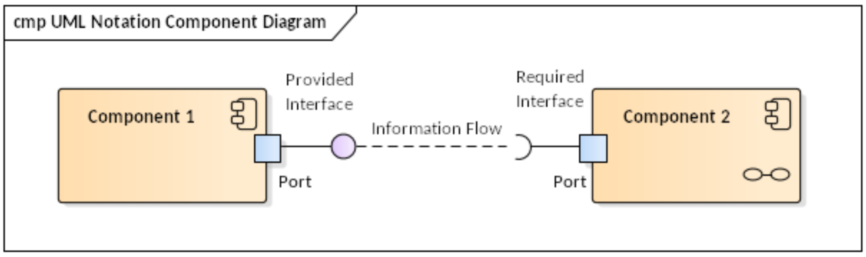


Figure 27. Component Diagram

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.

Digital Water .City

Leading urban water management to its digital future

digital-water.city
digitalwater_eu

digital-water.city has received funding from the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 820954.