Ref. Ares(2023)4812826 - 11/07/2023

D3.1 - RTM Specification and System Architecture

WP3- Real-Time Water Efficiency Monitoring Platform for the Process Industry

28/04/2023

Authors: Robert Sanfeliu Prat, Jordi Ricard Onrubia Palacios, Ricard Cervera Enrich, Alex Pujol Garcia (Eurecat)

The AquaSPICE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958396.

Document Information

GRANT AGREEMENT NUMBER	958396	ACRONYM	AquaSPICE
FULL TITLE	Advancing Sustainability of Process Industries through Digital and Circular Water Use Innovations		
START DATE	1st December 2020	**DURATION**	48 months
PROJECT URL	www.aquaspice.eu		
DELIVERABLE	D3.1 : RTM Specification and System Architecture		
WORK PACKAGE	WP3: Real-Time Water Efficiency Monitoring Platform for the Process Industry		
DATE OF DELIVERY	**CONTRACTUAL** 11/2021	**ACTUAL**	04/2023
NATURE	Report	**DISSEMINATION LEVEL**	Public
LEAD BENEFICIARY	EURECAT		
RESPONSIBLE AUTHOR	Robert Sanfeliu Prat		
CONTRIBUTIONS FROM	Jordi Ricard Onrubia Palacios, Ricard Cervera Enrich, Alex Pujol Garcia, Babis Magoutas (ICCS)		
ABSTRACT	This document presents the initial version of the Aqua Spice Real Time Monitoring Platform. The document starts by giving a brief introduction to the concept of industrial monitoring and automation and providing a summary of relevant technologies, protocols and monitoring platforms commonly used to achieve the said automation. Later, the document states the requirements extracted from AquaSPICE Case Studies with respect to the real time monitoring platform. With this requirement at hand, the document describes the proposed RTM Platform (based on FIWARE) architecture and analyses the fulfilment of the identified requirements.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
V1.0	03/11/2021	First version	First version of the document, prior to internal AquaSPICE members review.	Robert Sanfeliu Prat, Jordi Ricard Onrubia Palacios, Ricard Cervera Enrich, Alex Pujol Garcia
V1.1	23/11/2021	Final version	Final version after review	Robert Sanfeliu Prat, Jordi Ricard Onrubia Palacios, Ricard Cervera Enrich, Alex Pujol Garcia
V1.2	30/11/2021	Final version	Submission	
V1.3	28/04/2023	RESUBMISSION	Revised version considering CS#4 as void case. Added missing references. Reviewed citations and references to other works.	Robert Sanfeliu Prat (EURECAT)

Disclaimer

Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

© AquaSPICE Consortium, 2021-2023
This deliverable contains original unpublished work except where indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both. Reproduction is authorised if the source is acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS/ACRONYMS

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CoAP	Constrained Application Protocol
CS	Case Study
CSV	Coma Separated Value
DCS	Distributed Control System
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
ETSI ISG CIM	ETSI Industry Specification Group on cross cutting Context Information Management
FF	Foundation Fieldbus
FMS	Fieldbus Message Specification
GE	Generic Enabler
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HMI	Human-Machine Interfaces
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet Of Things
IP	Internet Protocol
JSON-LD	JSON-Linked Data
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LTE	Long-Term Evolution
M2M	Machine to Machine
MAC	Media Access Control
MES	Manufacturing execution system
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
OPC-UA	OLE for Process Control - Unified Architecture
PA	Process automation
PDP/PAP	Policy Decision Point/Policy Administration Point
PLC	Programable Logic Controller
Pub/Sub	Publish/Subscribe
REST	Representational State Transfer
RFID	Radio frequency identification
RTM	Real Time Monitoring

RTU	Real Time Unit
SASL	Simple Authentication and Security Layer
SCADA	Supervisory Control And Data Acquisition
SME	Small and Medium enterprise
SQL	Structured query language
SQS	Simple Queue Service
SSO	Single Sign-On
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
WPAN	wireless personal area networks
XACML	eXtensible Access Control Markup Language

1. Executive summary

This document presents the initial version of the Aqua Spice Real Time Monitoring Platform. The document starts by giving a brief introduction to the concept of industrial monitoring and automation and providing a summary of relevant technologies, protocols and monitoring platforms commonly used to achieve the said automation.

Later, the document states the requirements extracted from AquaSPICE Case Studies with respect to the real time monitoring platform. With this requirement at hand, the document describes the proposed RTM Platform (based on FIWARE) architecture and analyses the fulfilment of the identified requirements.

Due to the non-existence of the SynDi plant in CS#4, this CS is considered as void case and no work related to this CS is reported in this deliverable.

2. Introduction

AquaSPICE project aims at materializing circular water use in European Process Industries, fostering awareness and solutions. To achieve this generic goal, AquaSPICE stablished the technical goal of "Deliver a monitoring platform, ensuring the continuous and uninterrupted acquisition of reliable, cyber-secure and homogenized dynamic data that will enable the continuously and comprehensive knowledge of water-related processes".

The accomplishment of this goal is to be provided by "WP3: Real-Time Water Efficiency Monitoring Platform for the Process Industry". In that direction, "Task 3.1: RTM System Architecture" is devoted to i) provide an analysis of real time monitoring technology to be adopted at industrial level, ii) analyse Case Study requirements in relation to data collection and iii) propose an RTM system architecture that fulfils the identified requirements.

This document presents the results of work done on WP3 and is structured in the following manner:

- Section 1: Executive summary of the document
- Section 2: Is the introductory chapter, which provides the scope of the deliverable and the main outline of the document.
- Section 3: Provides a brief introduction to industrial monitoring and automation: outlines the key concepts and reviews relevant communication technologies, protocols, and monitoring platforms.
- Section 4: Lists the requirements for CS with respect to the RTM Platform.
- Section 5: Proposes a RTM Platform design to fulfil the identified requirements.

3. Industrial monitoring and automation

Industrial monitoring and automation refer to the use of control systems, such as computers or robots, and information technologies to achieve an automatic operation and control of industrial processes without significant human intervention.

The main benefits of industrial monitoring and automation can be summarized as:

- Productivity increase: Many automated processes can be executed at a much higher speed than the manual counterparts. e.g.: Putting bottle caps by hand or by a robot. Similarly, automation can help eliminating repetitive checks: e.g.: check water level of a tank.
- Quality improvement: Automation aims to reduce human intervention in fabrication processes, thus, reducing the possibility of human error. Moreover, constant fabrication parameter monitoring ensures the delivery of products that comply with the specifications.
- Flexibility: The growing need for product customization is forcing factories to shift from the fabrication of large batches of identic items into a paradigm where production runs are smaller and items are slightly different. In this scenario, industrial monitoring and automation is a key enabler that helps in the rapid re-configuration of production plant.

3.1. The monitoring and automation pyramid

Industrial monitoring and automation systems can be exceptionally intricate, having large number of devices working in synchronism. Typically, automated industrial systems are structured in hierarchical levels. These systems are usually represented as a pyramid called the monitoring and automation pyramid, which divides the layers of production in five levels: Field, Control, Supervisory, Planning and Management from bottom to top. Each layer represents another level of abstraction as explained in the following subsections.

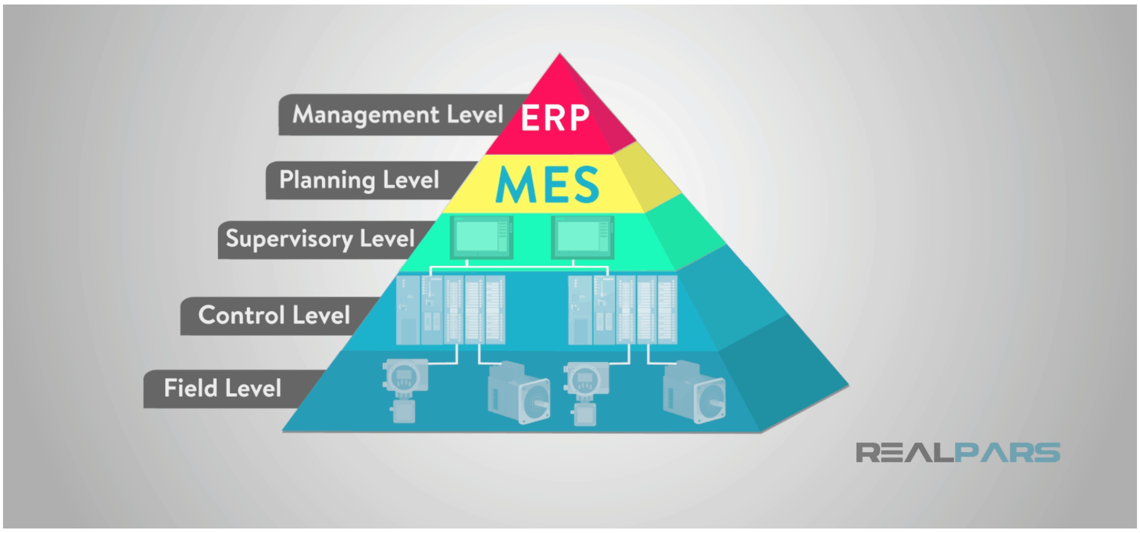

Figure 1: Monitoring and automation Pyramid

(source: https://realpars.com/automation-pyramid/)

3.1.1. Field Level

The lower level of the pyramid consists of field devices such as sensors and actuators installed in the machinery and fabrication environment to sample important process variables (temperature, humidity, speed, etc…) and act on the process to alter it (control air flow inside a blower, start/stop the process, etc…).

3.1.2. Control Level

The role of this level is to control and monitor fabrication processes. This is achieved by analysing the process variables gathered and acting using the installed actuators.

Therefore, this level requires features such as short response time, high-speed transmission, short data lengths, machine synchronization, constant use of critical data, etc.

A wide range of devices belong to this automation level: industrial controllers such as PLCs, distributed control units, and computer systems.

3.1.3. Supervisory Level

The next layer up is where Supervisory Control And Data Acquisition (SCADA) systems as well as Human-Machine Interfaces (HMI) are found. In this layer, process data is monitored through user interfaces, and stored in database. SCADA is typically used to control multiple machines in complex processes, including processes that involve multiple sites.

There are many similarities between the Control level and the Supervisory level, both levels receive input from the shop floor and return outputs to control processes, but one difference is that SCADA in the Supervisory level is often used to allow the human supervisor to adjust or modify decisions taken by the logic programmed at control level.

3.1.4. Planning Level

This level utilizes a computer management system known as MES or manufacturing execution system. MES monitors the entire manufacturing process in a plant or factory from the raw materials to the finished product.

3.1.5. This allows management to see exactly what is happening and allows them to make decisions based on that information. They can adjust raw material orders or shipment plans based on real data received from the systems. It also includes highly advanced tools for tracking important shop floor metrics, such as labor and equipment usage and performance, that can be used to optimize production efficiency and reduce or eliminate waste. [1]Management Level

The top of the pyramid is what is called the management level. This level uses the companies integrated management system, which is known as the ERP or Enterprise Resource Planning.

This is where a company's top management can see and control their operations. ERP is usually a suite of different computer applications that can see everything going on inside a company. It utilizes all the previous levels technology plus some more software to accomplish this level of integration.

This allows the business to be able to monitor all levels of the business from manufacturing to sales, to purchasing, to finance and payroll, plus many others. The integration of the ERP promotes efficiency and transparency within a company by keeping everyone in the same page. [1]

3.2. Industrial communication

Communication requirements between the different layers of the monitoring and automation pyramid vary substantially from layer to layer. Whilst communication between field level and control level must be fast and extremely robust, communications between upper layers of the pyramid require higher bandwidths. Moreover, legacy protocols (often proprietary) are pervasive in industrial environments. For this reason, it is very common to find different types of communication technologies across the automation pyramid, some of which are limited between two of its layers, while others span many layers or even the whole pyramid.

This section provides a brief introduction to the most common communication technologies to be found in industry. Although it is not intended to serve as a full overview and comparative of those, it will help on the process of understanding current communication infrastructure in AquaSPICE Case Studies.

Industrial communication can be divided into the following i) wired technologies and ii) non-wired technologies.

3.2.1. Wired technology

Wired technologies group all the communication technologies that use electrical-conductive wires (often copper) as the medium for transmitting data between producers and receivers. This category ranges from undocumented proprietary device-to-device communications systems (not covered in this deliverable), to bus and non-bus based technologies and finally, to mother ethernet-compatible technologies.

3.2.1.1 Non-bus-based technologies

These category of communication technologies are categorized by the fact that the receivers used (SCADA's, PLC's, etc…) to centralize the readings, require an isolated interface for each of the transmitters to connect. That is, a receiver for 30 sensors will have 30 different ports. Below is a short list of most relevant non-bus based technologies:

RS232

The RS232 [2] is a serial analogue peer-to-peer communication originated at the mainframe era. Theoretically, RS232 is designed to work at maximum distances of 15m and ran at a speed of 20Kbps, but both limits had been exceeded with acceptable results.

40-20 ma Current Loop

Created at the 50's, the 4 to 20m current loop [3] is a very robust analog sensor signaling standard, ideal for data transmission. When this transmitter (attached to a sensor) senses an analogue input from the sensor, it modulates that signal using a DC data source in the range of 4ma - 20ma. This standard is not susceptible to noise, and it can be transported to long distances.

3.2.1.2 Bus-based-technologies (Field Bus)

Fieldbus is the name of a family of industrial computer networks used for real-time distributed control. Fieldbus works on a network structure with a topology that can range from daisy-chain, star, ring, branch, or tree network. In comparison with the non-bus-based technologies, Fieldbus allows a controller to receive information from hundreds of transmitters using a single communication port. This reduces both the length of cable used and the number of lines.

Fieldbus profiles are standardized by the International Electrotechnical Commission (IEC) as IEC 61784/61158 [4] Below is a short list of most relevant bus based technologies:

RS485

The interface of RS485 [5] is seen as an evolution of RS422. As its predecessor, it offers high transmission speed and a large usage distance. The improvement over the RS422 interface is the upgrade to a bidirectional bus with up to 32 users (expandable to 128). Even though this interface is being replaced for ethernet-based ones it's still a very widespread interface into automation industry.

Given that multiple transmitters work in a common line, it must be guaranteed that only one transmitter is active at the same time and the other one reminds in ultra-ohm state.

CAN / CANOpen

Initially developed by BOSCH for the automotive industry, CAN (Controller Area Network) [6] allows the communication between any devices in the network using a two-wire bus. CAN has become an international standard (ISO 11898) and has lead to the creation of several standardized protocols as CANopen [7] and DeviceNet [8] among others.

Profibus / Profibus PA

The fieldbus Profibus [9] is based on the Fieldbus Message Specification (FMS). After the changes proposed by Siemens on 1994, introducing the concept of Decentralized Periphery, the technology gained good acceptance in the manufacturing industry, becoming in 2016 one of the most installed fieldbuses on the world and approaches 60 million of installed nodes in 2018[1].

Profibus PA (process automation) is used for communication between measuring and process instruments, actuators and process control system or PLC/DCS in process engineering. Profibus PA is a Profibus version with physical layer suitable for process automation, in which several segments (PA segments) with field instruments can be connected to Profibus DP via so-called couplers.

Foundation

Foundation Fieldbus (FF) [10] was designed to address the needs of real-time process automation, it was conceptually engineered as a replacement of the traditional analogue 4-20mA current loop. FF is an digital, serial, two-way communications system designed for in a plant or factory automation environment.

Using FF field transmitters can communicate with each other without a host. The "control" functionality within FF can be in the field device or in the host or partially in both. When control is in the field devices, the host can be disconnected without halting the loop. Additionally FF provides a distributed real time clock on the bus to synchronize devices.

Modbus

Modbus [11] uses in his first version a two-wire cable with RS-485 and UART signals. The protocol itself is very simple withis a master/slave protocol with a maximum of 247 slaves. Modbus comes in several varieties including Serial RTU, Serial ASCII, TCP/IP and UDP/IP. Modbus-TCP version still one of the most used industrial networks mainly in the building automation field [12].

BitBus

Bitbus [13] is the oldest commonly used bus technology. It useRS-485 at the physical layer, with two twisted pairs, one for data and the other for clocking and signals. It

[1] https://www.profibus.com/newsroom/press-news/more-than-20-million-profinet-devices-on-the-market

uses SDLC at the data link layer that permits 250 nodes on one segment with a total distance of 13.2 km. Bitbus does not define routing at the network layer and implements master/slave protocol with one master node and multiple slaves, with slaves only responding to requests from the master.

AS Interface

AS-Interface [14] (Actuator Sensor Interface, ASi) is an industrial networking solution that include a physical layer, data access method and protocol. This solution I commonly used in PLC, DCS and PC-based automation systems. AS-Interface is a networking alternative to the hard wiring of field devices. It can be used as a partner network for higher level fieldbus networks such as Profibus, DeviceNet, Interbus and Industrial Ethernet. The AS-Interface has been an international standard according to IEC 62026-2 since 1999. [15]

Interbus

INTERBUS [16] has been designed as a fast sensor–actuator bus for transmitting process data in industrial environments. Due to its transmission procedure and ring topology, It offers features such as fast, cyclic, and time-equidistant process data transmission, diagnostics to minimize downtime, and easy operation and installation, as well as meets the optimum requirements for fiber-optic technology [17].

It recognizes two cycle types: the identification cycle for system configuration and error management, and a data transfer cycle for the transmission of user data. Both cycle types are based on a summation frame structure

3.2.1.3 Industrial Etherenet

Due to its inherent reliability, performance, and interoperability, Ethernet has extended from office communication to the production floor as the communication protocol of choice for automation and control systems. In the late years, Industrial Ethernet has surpassed the market share of traditional fieldbus protocols that typically require multiple wires.

However, the adoption of ethernet standard in the industry requires some application requirements to be met. In industry, the reliability of communications is a fundamental necessity. Loosing a single packet of information might result in catastrophic events in the production plant. Also, timing constraints are much stricter that in standard Ethernet applications. Moreover, the physical environment where industrial networks are to be deployed (factory floor, with plenty of electromagnetic interferences, dust, humidity, extreme temperatures) is much harsh than the typical office. Finally, for a successful implementation of Ethernet in industry, a broad compatibility with the existing wired communication technologies is mandatory. For these reasons, several Ethernet standard adaptations for industrial usage have appeared:

Ethernet POWERLINK

Ethernet POWERLINK [18] is a free communication system based exclusively on software that complies with all the standard features of Ethernet. POWERLINK is intended for "hard" real-time applications, where signal propagation delays must occur within a precisely defined time interval so that no error message is generated. This makes it particularly suitable for motion control, robotics, and real-time I/O visualization applications.

Modbus TCP/IP

MODBUS TCP [19] is based on MODBUS communication standard, injecting its payload inside the Ethernet application layer. MODBUS TCP uses a mater/slave architecture but is not considered to be a real-time protocol. The combination of Ethernet networks and universal TCP / IP network standards and non-exclusive data representation by MODBUS provides a completely open system for exchanging process data. It offers many advantages, including ease of use on all TCP / IP compliant devices, and provides fast and efficient communication over industrial networks.

EtherCat

The technology, developed by Beckhoff and the EtherCAT Technology Group (ETG), is based on a master/slave architecture and offers real-time data transfer. EtherCAT [20] enables the periodic transmission of input / output data as well as transmission of parameters, metadata, diagnostic data and clock synchronization.

EtherNet/IP

EtherNet/IP [21] is an open industrial network standard of the Open DeviceNet Vendors Association (ODVA). Used to transfer periodic input/output data and non-periodic parametric data. EtherNet/IP, mainly in the US market, is a widely used communication standard for implementing Rockwell control panels. EtherNet/IP In addition to the connection and transmission of process data and parameters, it also supports computer functions such as web servers and e-mail clients.

Profinet

PROFINET [22] is an open industrial Ethernet standard of the PROFIBUS User Organization (PNO) and is extensively promoted by Siemens. Based on design concepts of PROFIBUS DP, PROFINET provides fast data communication over Ethernet networks in combination with industrial computing functions. Given its real-time capabilities and ease of integration, the protocol has become the leading data transfer standard for large-scale plants.

3.2.2. Wireless technologies

Wireless technologies differ from wired technologies on a fundamental fact: the use of wire (often copper) as a communication medium. Instead, wireless technologies use air for the propagation of electro-magnetic waves that transport information. This offers two major benefits:

Although some wired technologies can range up to 10ths of Kms, in some situations, the distances between transmitters and receivers are much greater, or perhaps, the installation of communication wires between the nodes is not viable. In that scenario, wireless communications eliminate the necessity of physically interconnecting the nodes.

Similarly, as semiconductor integration capabilities improved and, consequently, the miniaturization of electronic components, Wireless communications opened the opportunity for moving digital objects.

Below is a short list of most relevant wireless communication technologies for short distance and long distance communication:

3.2.2.1 Short distance

Wifi (IEEE 802.11)

The 802.11 standard is a family of wireless regulations better known as WIFI [23] it is one of the most popular wireless standards in all sectors due to its wide use in homes, offices, and industrial environments. Its main use is the creation of a wireless local area network (WLAN) through MAC protocols and physical layer, which allow multiple devices to be connected to the same network wirelessly to share information between them and have access to the Internet through a router. The most recent regulations allow the use of the frequency bands of 2.4 GHz and 5 GHz, being the second much faster but with less range, thus allowing greater flexibility when configuring the network necessary for a development.

Buetooth

Bluetooth [24] is an industrial specification for wireless personal area networks (WPAN) that enables the transmission of voice and data between different devices using a radio frequency link in the 2.4 GHz ISM band. This specification focuses on the exchange of information between two devices and allows transmission speeds of 32 Mb/s, allowing the constant transmission of moderately heavy data, such as the sending of audio and voice data, which makes it one of the most widely used protocols in the telecommunications and personal computing sectors.

Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a wireless technology derived from the Bluetooth 4.0 standard, which emits data at 2.4 GHz with a range of up to 100 meters. BLE is designed to have a much lighter sending capacity than classic Bluetooth, which provides a great improvement in low power consumption at the cost of losing data transmission speed, these features make it perfect for applications that do not need a constant flow of data i that are powered by batteries, some examples would be: personal electronics devices, IoT sensor nodes, security or home automation among others. Despite deriving from the same standard as classic Bluetooth, these two protocols are not supported.

Radio frequency identification (RFID)

Radio frequency identification (RFID) uses electromagnetic fields to automatically identify, and track tags attached to objects. An RFID system consists of a small radio transponder, a radio receiver, and a transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device the tag transmits digital data, usually an identification inventory number, back to the reader. This number can be used to track inventory goods. There are two types of tags, the passive ones that use the radio frequency power generated by the reader and therefore need it to be at a short distance (10 – 30 cm) and the active tags that work with a battery and, therefore, can be read at a greater range from the RFID reader, up to hundreds of meters. [25]

Near Field Communication (NFC)

Near Field Communication (NFC) [26] is a set of communication protocols between two electronic devices at a distance of 4 cm or less. NFC offers a low-speed connection with a simple setup that can be used to initiate more capable wireless connections. NFC devices can act as electronic identity documents and access cards. They are used in contactless payment systems and allow mobile payment to replace or complement systems such as credit cards. The protocol requires that the electronic device has a specific antenna for NFC communications and by working exclusively at short distances provides greater security in communications. [27]

3.2.2.2 Long distance

Cellular networks 3G/4G/5G

The cellular network or a mobile network is a communication network where the link to and from the end nodes is wireless. The network is distributed over terrestrial areas called "cells," each of which has at least one fixed-location transceiver. These base stations provide the cell with network coverage that can be used for data transmission. Typically, a cell uses a different set of frequencies than neighboring cells, to avoid interference and provide a guaranteed quality of service within each cell. This technology is the one that is used daily for communication with smartphones, both for voice data communication and for internet access.

[28]

GSM / GPRS (2G/3G)

The Global System for Mobile Communications (GSM) is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe protocols for second-generation (2G) digital cellular networks used by mobile devices such as mobile phones and tablets. From this standard various communications protocols used in many countries have been developed, making it a globally recognized standard for mobile communications.

General Packet Radio Service (GPRS) is a packet-oriented mobile data standard in the global system of the 2G and 3G cellular communication network for mobile communications (GSM). GPRS is a best-effort service, which involves variable throughput and latency that depend on the number of other users sharing the service at the same time, unlike circuit switching, where a certain quality of service (QoS) is guaranteed during the connection.

These standards have become obsolete in the world of personal communications where standards with higher connection speed and coverage, mainly LTE, are chosen, but it is still used in some IoT applications with fewer speed requirements due to its lower price.

[29]

LTE (4G)

Long-Term Evolution (LTE) is a standard for wireless broadband communication for mobile devices and data terminals, based on GSM technologies. Increase capacity and speed using a different radio interface along with improvements to the core network. LTE is the next generation upgrade for operators with GSM/GPRS networks. LTE is the most used technology in the world of personal telecommunications since it has an antenna network that offers the best possible coverage in most areas.4G

IoT applications it is frequently used in devices that do not have internet access via Wi-Fi/ethernet and require a constant sending of data. [30]

5G

It is the successor to 4G technology which provides connectivity to most mobile phones today. According to the GSM Association, by 2025, 5G networks are expected to have more than 1.7 billion subscribers worldwide [31]. The most remarkable advantage of this technology is that it will support greater bandwidth which will translate into higher download speeds, which can exceed 10 gigabits per second. Due to the aforementioned increase, it is expected that these networks will not only be used by telephones as is the case with current telephone networks but can also be used for general use on desktop or laptop computers. For this same reason, new applications are expected in areas such as the Internet of Things (IoT) and machine-to-machine.

Low Power Wide Area Network (LPWAN)

LPWAN technologies unlike cellular technologies aim to reduce the energy consumption of communications while maintaining the long sending distance. For this, the main common feature among all of them is the use of a much narrower bandwidth than cellular technologies, with this a lot of sending speed is lost, so these technologies are very focused on the world of the Internet of Things, where in many cases it rewards low consumption compared to the speed of sending data.

LoRa

LoRa enables long-range transmissions with low power consumption. The technology covers the physical layer, while other technologies and protocols such as LoRaWAN [32] Long Range Wide Area Network) cover the upper layers. It can reach data rates between 0.3 kbit/s and 27 kbit/s, depending on the dispersion factor. Since LoRa defines the lower physical layer, the upper layers of the network were missing. LoRaWAN is one of several protocols that were developed to define the upper layers of the network. LoRaWAN is a cloud-based media access control layer (MAC) protocol but acts primarily as a network layer protocol to manage communication between LPWAN gateways and end node devices as a routing protocol, maintained by the LoRa Alliance.

LoRa is an open protocol that allows its use to hardware and software developers, so it provides a great configurability of the desired architecture, and thanks to this a wide variety of devices are on the market compatible with LoRa. [33]

3.2.2.3 *Wireless mesh networks*

A wireless mesh network consists of a group of devices gateways (entry point for clients to the network) and routers (interconnect gateways) geographically distributed in relatively large areas, that allow clints to communicates with any other device in the network through many different routes.

The main advantages of mesh networks over traditional networks are:

- Coverage: Additional gateways can be easily installed to improve coverage of the network.
- Robustness: In a meshed network, the gateways and routers are dense linked (each device has a direct connection with multiple other devices). Thus, if any of the network elements fails, traffic is automatically routed through an alternative path.

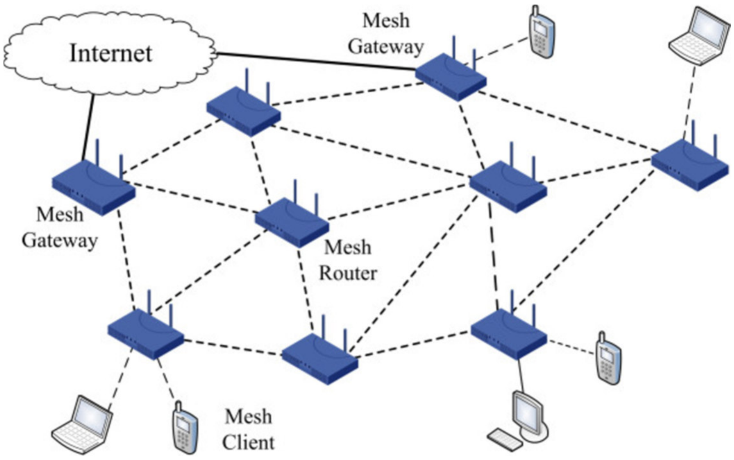

Figure 2: Mesh network

(source: An Overview of Topology Control Mechanisms in Multi-Radio Multi-Channel Wireless Mesh Networks).

Z-wave

Z-Wave [34] is a wireless communications protocol that is primarily used for home automation... It creates a Mesh network that uses low-energy radio waves to communicate from one appliance to another, allowing wireless control of residential appliances and other devices such as lighting control, security systems, thermostats, windows, locks, swimming pools and doors. AZ-Wave system can be controlled via the Internet from a smartphone, tablet, or computer, and locally using a wall-mounted panel with a Z-Wave Gateway. Z-Wave is a closed protocol, and a license is needed to manufacture devices that work with it, thanks to this it ensures that all the devices that use it are compatible with each other, but it increases its cost. [35]

Zigbee

Zigbee [36] is a specification based on the IEEE 802.15.4 standard for a set of high-level communication protocols that are used to create personal area mesh networks with small, low-power digital radios in applications such as home automation, medical device data collection, and other scenarios with low-power sensors, is useful in situations where low bandwidth is available, and is designed for small-scale projects that need wireless connection. Zigbee is an open protocol that allows its use in all types of environments and operating systems, a fact that facilitates the development of devices that use it and allows a high level of configurability but does not ensure that two devices that use Zigbee are compatible. [37]

Thread

Thread [38] is a low-power, IPv6-based mesh network technology for Internet of Things products, designed to be secure and future-proof. The Thread protocol specification is available at no cost; however, this requires agreement and continued adherence to an End User License Agreement (EULA), which states that "Membership in the Thread Group is necessary to implement, practice, and distribute Thread Technology and Thread Group specifications." Like Zigbee, Thread is also based on the IEEE 802.15.4 standard, with the difference that Thread allows IP routing, with access to the cloud and AES encryption. [39]

Sigfox

Sigfox [40] is a French global network operator founded in 2010 that builds wireless networks to connect energy-efficient objects such as electricity meters and smartwatches, which need to be continuously turned on and emitting small amounts of data. Sigfox enables communication using the industrial, scientific, and medical radio band that uses 868 MHz in Europe and 902 MHz in the US. It uses a wide-range signal that passes freely through solid objects, called ultra-Narrow Band, and requires little power. The network is based on a one-hop star topology and requires a mobile operator to transport the generated traffic. The signal can also be used to easily cover large areas and reach underground objects.

Due to the private nature of the protocol and the distribution of antennas, Sigfox is an annual payment protocol, the price of which depends on the number of connected devices, which allows the use of the network and the Sigfox Cloud. [41]

LTE-M

LTE-M [42], short for LTE Cat-M1, is a new Low Power Wide Area Network (LPWAN)technology, developed for IoT applications. It is a protocol that takes advantage of the structure of cellular communications but at a narrow bandwidth that connects to the Internet devices of low complexity that transmit small amounts of data in long periods of time, with low power consumption.

LTE-M is a mixed technology between cellular and LPWAN, so it has a higher bandwidth than the rest of LPWAN technologies, which allows higher data rates, lower latency, and a much more accurate positioning of the device. It also enables voice and mobility delivery, greater coverage and is ideal for machine-to-machine (M2M) communications and solutions using IoT devices. It is the key element to improve the efficient use of resources.

NB-IoT

NB-IoT [43] is a cellular technology, uses cellular communication bands and has been designed to operate in different ways, including the use of the GSM band replacing the current deployment (standalone), using the LTE band, and therefore sharing it (in-band) or even using the spacing between LTE channels to make the most of the communications spectrum (guard-band). Unlike LPWANs, NB-IoT is born conditioned by the LTE architecture and must coexist with this technology without introducing modifications to the structure and architecture of the cellular network. This implies a much higher complexity than its LPWAN competitors. NB-IoT is a half-duplex technology that efficiently enables uplink communication, that is, it allows an establishment of connection to the cellular network, the allocation of network resources to the node (known as User Equipment or UE) and the transmission of data

[44]

3.3. IoT Data Protocols

This section provides a brief introduction to most common IoT data protocols: REST (Representional State Transfer), MQTT (Message Queuing telemetry Transport), AMQP (Advanced Message Queueing Protocol), CoAP (Constrained Application Protocol) and DDs (Data Distribution Service)

3.3.1. REST / HTTP

REST has emerged as the predominant Web API design model. RESTful style architectures conventionally consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource.

REST was initially described in the context of HTTP, but it is not limited to that protocol. RESTful architectures may be based on other Application Layer protocols if they already provide a rich and uniform vocabulary for applications based on the transfer of meaningful representational state. [45]

3.3.2. Message Queuing telemetry Transport (MQTT)

MQTT [46] is a message-centric wire protocol designed for M2M communications that enables the transfer of telemetry-style data in the form of messages from devices, alongin high latency or constrained networks, to a server or small message broker. Devices may range from sensors and actuators, to mobile phones, embedded systems on vehicles, or laptops and full scale computers. It supports publish-and-subscribe style communications and is extremely simple.

MQTT defines methods to indicate the action to be performed on the identified resource. These methods are: *(i)* connect; *(ii)* disconnect; *(iii)* subscribe; *(iv)* unsubscribe and *(v)* publish. MQTT is widely used and there are several projects that implement MQTT such as Microsoft Azure IoT Hub, OpenStack, Amazon Web Services …). [45]

3.3.3. Advanced Message Queueing Protocol (AMQP)

AMQP [47] is a message-centric protocol for sending interoperable messages between two or more clients that emerged from the Financial sector with the aim of freeing users from proprietary and non-interoperable messaging systems. AMQP depicts the behaviour of the messaging provider and client ensuring that implementations from different vendors are truly interoperable.

AMQP isa binary, application layer protocol designed to efficiently support a wide variety of messaging applications and communication patterns. It provides flow controlled, message-oriented communication with message-delivery guarantees such as at-most-once (where each message is delivered once or never), at-least-once (where each message is certain to be delivered, but may do so multiple times) and exactly-once (where the message will always arrive and do so only once,), and authentication and/or encryption based on SASL and/or TLS It assumes an underlying reliable transport layer protocol such as Transmission Control Protocol (TCP). [45]

3.3.4. Constrained Application Protocol (CoAP)

CoAP [48] is a document transfer protocol. Mainly, it was designed to communicate over the Internet very simple electronic devices. CoAP is being standardised by the Internet Engineering Task Force (IETF) Constrained Restful Environments (CoRE) Working Group.

CoAP is focused on providing communication capabilities to small low power sensors, switches, valves and resource constrained internet devices such as Wireless Sensor Networks (WSNs). Moreover, it is designed to easily translate to HTPP for simplified RESTful web integration. CoAP is lightweight, simple and runs over UDP (not TCP) with support for multicast addressing.

CoAP is based on RESTful architecture and hence, it supports client/server programming model where the resources are server controlled abstractions made available by an application process and identified by Universal Resource Identifiers (URIs). Clients initiate requests to resources using HTTP request methods such as GET, PUT, POST and DELETE. It is important to note that CoAP supports resource discovery.[49]

3.3.5. Data Distribution Service (DDS)

The DDS standard [50] i data-centric publish-and-subscribe technology that emerged from the Aerospace and Defence community to address the data distribution requirements of mission-critical systems. It enables scalable, real-time, reliable, high performance and interoperable data exchanges between publishers and subscribers. Moreover, it is both language and OS independent. DDS is used on business-critical applications like financial trading, air traffic control, smart grid management, and other big data applications. Also, it is used in a wide range of Industrial Internet applications.

The DDS specification defines: *(i)* a Data Centric Publish Subscribe (DCPS) layer; *(ii)* a DDS Interoperability Wire Protocol (DDSI); and *(iii)* an Extensible and Dynamic Topic Types for DDS standard.

DCPS provides a set of APIs that present a set of standardised “profiles” targeting real-time information-availability for any domain. Moreover, these APIs have been implemented in a range of different programming languages (Ada, C, C++, C#, Java, JavaScript, CofeeScript, Scala, Lua, and Ruby) and helps to ensure that DDS applications can be ported easily between different vendor’s implementations.

The DDS Interoperability Wire Protocol (DDSI) is a wire-level protocol refers to the mechanism for transmitting data from point-to-point which is needed if more than one application has to interoperate. The protocol also supports automatic “Discovery” that allows DDS participants to declare the information that they can provide or what data they would like to receive, in terms of topic, type and QoS.

Extensible and Dynamic Topic Types defines how Topic data types can be extended dynamically while ensuring application portability and interoperability. [51]

3.4. Monitoring platforms

The number of industrial monitoring platforms (sometimes called Industrial Internet of Things platforms) is growing rapidly. With Gartner listing 78 offerings[2] among which one can find big IT companies such as Microsoft, Amazon, Oracle, etc… This section provides an overview of existing open-source monitoring solutions to be used on AquaSPICE project.

3.4.1. KAA

Kaa [52] is a highly flexible, multi-purpose middleware platform for implementing complete end-to-end IoT solutions, connected applications, and smart products. The platform comes in two licensing options: Open-source (up to v.0.10, now discontinued) and proprietary (since v.0.10)[3].

As described in the project documentation [53], the open-source version of the platform features include:

Device management

Kaa provides a register of digital twins, which represent things, devices, and other entities managed by the platform. Kaa also allows to store device attributes. To connect to the platform, a device has to present valid credentials.

Data collection

Kaa uses a proprietary protocol for collecting structured data from connected devices. This protocol ensures reliable data delivery with response. To minimize network usage and improve the data throughput, the protocol supports batching.

Data processing and analytics

Kaa platform features data collection adapters that allow sending data to various databases or data analytics systems. Raw, unstructured data can also be transformed into well-structured time series, convenient for analytics, pattern analysis, visualization, charting, etc. Out of the box, Kaa is integrated with Open Distro for Elasticsearch, which provides advanced analytics, visualization, and alerting features for Kaa users.

Time series consumers can listen to new data points and trigger particular actions, for example, send mobile push notifications.

Data visualization

Kaa can be connected to other visualization tools thanks to its open APIs.

[2] https://www.gartner.com/reviews/market/industrial-iot-platforms

[3] https://www.kaaiot.com/kaa-open-source

Alerts, triggers, and notifications

Kaa is pre-integrated out of the box with the Open Distro for Elasticsearch, a powerful open-source analytics platform. It allows to create alerts based on analytics that can be then notified by various communication channels, such as email, Slack, SMS, push notifications, etc. It is also possible to set up specific actions based on preconfigured data thresholds, thus enabling basic rule engine functionality.

Configuration management

The Kaa platform offers the capability of managing device configurations (variables, set points, etc…). To ensure reliability, the configuration delivery is based on configuration application confirmations and result codes. The configuration requests can be delivered to the devices using both the push (a device may subscribe to be notified) and pull (device periodically checks for changes) modes.

Configuration management can be performed individually (one device at a time) or in groups of devices based on their individual characteristics.

Multitenancy

The Kaa platform s.upports advanced multitenancy, which means that a single Kaa instance may serve multiple independent tenants and each tenant's data is completely isolated from other tenants

In its open-source version, Kaa features a monolithic architecture and the scalability must be managed manually. Kaa is mainly developed in Java.

3.4.2. ThingsBoard

ThingsBoard [54] is an open-source IoT platform for data collection, processing, visualization, and device management.

As described in the project documentation [55], the main attributes of the platform are:

scalable: horizontally scalable platform, build using leading open-source technologies.

fault-tolerant: no single-point-of-failure, every node in the cluster is identical.

robust and efficient: single server node can handle tens or even hundreds of thousands of devices depending on use-case. ThingsBoard cluster can handle millions of devices.

durable: never lose your data. ThingsBoard supports various queue implementations to provide extremely high message durability.

customizable: adding new functionality is easy with customizable widgets and rule engine nodes.

The architecture of ThingsBoard platform (Illustration 1: ThingsBoard architecture) is composed by the following modules:

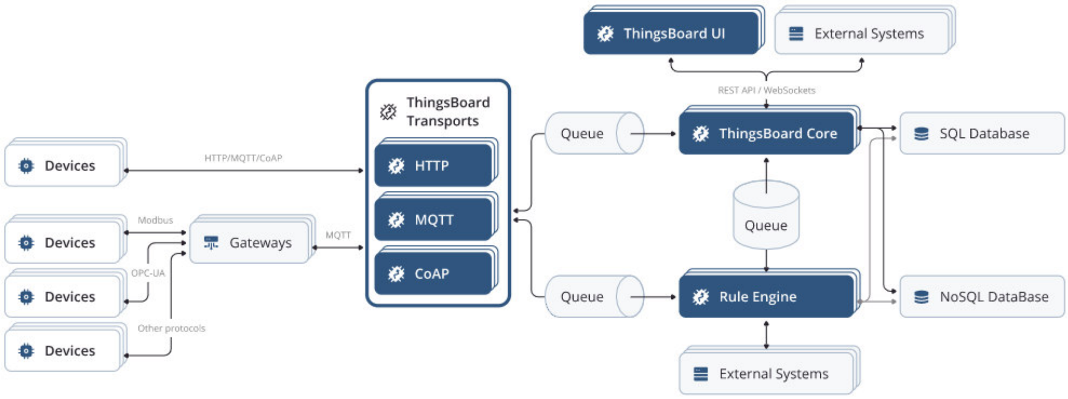

Illustration 1: ThingsBoard architecture

(source: https://thingsboard.io/docs/)

ThingsBoard Transports: ThingsBoard provides MQTT, HTTP, CoAP and LwM2M)based APIs that allow the integration of external devices. Each of the protocol APIs are provided by a separate server component and is part of ThingsBoard "Transport Layer". MQTT Transport also provides Gateway APIs to be used by gateways that represent multiple connected devices and/or sensors.

Once the Transport receives the message from device, it is parsed and pushed to durable Message Queue. The message delivery is acknowledged to device only after corresponding message is acknowledged by the message queue.

ThingsBoard Core: Is responsible for handling REST API calls and WebSocket subscriptions. It is also responsible for storing up to date information about active device sessions and monitoring device connectivity state.

ThingsBoard Rule Engine: ThingsBoard Rule Engine is a highly customizable and configurable system for complex event processing. It allows to filter, enrich and transform incoming messages originated by IoT devices and related assets, and trigger actions (e.g: notifications or communication with external systems). The rule engine is programmed with a graphical programming interface, connecting functional blocks that implement pre-existing logic or custom JS functions.

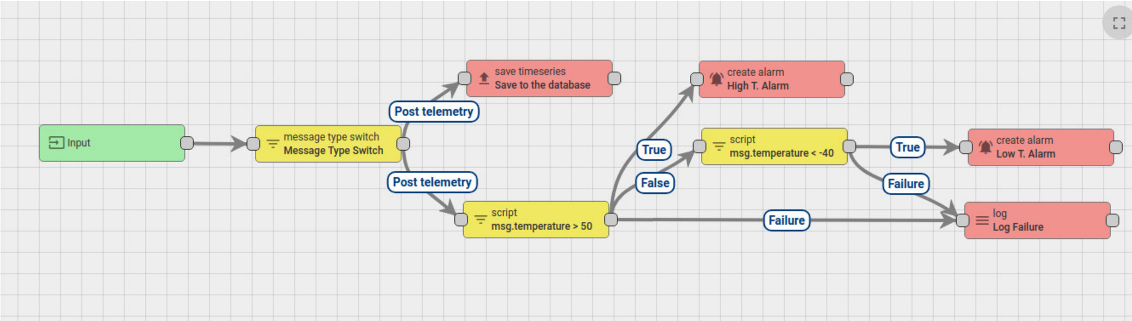

Figure 3: ThingsBoard Rule Engine panel

(source: https://thingsboard.io/docs/)

ThingsBoard Web UI: ThingsBoard provides a lightweight component written using Express.js framework to host static web ui content. Those components are completely stateless and no much configuration available. The static web UI contains application bundle. Once it is loaded, the application starts using the REST API and WebSockets API provided by ThingsBoard Core.

ThingsBoard supports multiple message queue implementations: Kafka, RabbitMQ, AWS SQS, Azure Service Bus and Google Pub/Sub. Using durable and scalable queues allow ThingsBoard to implement back-pressure and load balancing. Back-pressure is extremely important in case of peak loads.

It is possible to run the platform as a monolithic application or as a set of microservices. Monolithic mode requires minimum support efforts, knowledge and hardware resources to do the setup and low maintenance efforts. Microservices approach offers the capability to offer high availability and scale to millions of devices.

ThingsBoard uses database to store entities (devices, assets, customers, dashboards, etc) and telemetry data (attributes, timeseries sensor readings, statistics, events). Platform supports three database:

SQL - Stores all entities and telemetry in SQL database

Hybrid (PostgreSQL + Cassandra) - Stores all entities in PostgreSQL database and timeseries data in Cassandra database.

Hybrid (PostgreSQL + TimescaleDB) - Stores all entities in PostgreSQL database and timeseries data in Timescale database.

ThingsBoard back-end is written in Java, but also has some micro-services based on Node.js. ThingsBoard front-end is a SPA based on Angular 9 framework.

3.4.3. Fiware

FIWARE [56] is an open-source framework initiative defining a set of standards for context data management. As described in the project documentation [57], FIWARE offers the possibility to access and manage heterogeneous context information through open API. FIWARE is built around the concept of Generic Enablers (GE). A GE it's an independent module aimed at solving a particular need (data visualization, long term persistence, security management, data integration, etc…). A FIWARE compatible architecture is composed by a set of GEs connected to a core Context Broker. The Context Broker is responsible for managing the lifecycle of context information. This context information consists of entities (e.g.: a pump) and their attributes (e.g. max flow rate, current flow rate, work mode, etc..). Using the APIs provided by the context broker, one can create context elements, manage them though updates, update their attributes, perform queries to retrieve their status, and subscribe to context changes. On Its heart, the context broker utilizes a database to store the last version of the context information.

The open standard nature of FIWARE NGSI offers programmers the ability to port their applications across different "Powered by FIWARE" platforms and a stable framework for future development. Additional functionality can be easily added to a Smart Solution simply by using additional FIWARE or third-party components for which the integration with the FIWARE Context Broker component is solved. This integration is simplified since all components comply to the FIWARE NGSI standard interface, which eliminates vendor lock in. The component-based nature of a FIWARE based solution allows for re-architecting as the solution evolves according to business needs.

Following, a listing of the different GE categories and available components in each category is made:

3.4.3.1 Context Broker

There exist several implementations of Context Broker in the FIWARE ecosystem. The most prominent are:

- Orion-LD[4]: Orion-LD is a NGSI-LD Broker, which supports the NGSI-LD and the NGSIv2 APIs. Currently in Beta 3 version, plan is to merge it with the main branch of Orion. Uses MongoDB as storage.

- Stellio[5]: Stellio is an NGSI-LD compliant context broker developed by EGM. NGSI-LD is an Open API and Datamodel specification for context management. Its main components are a Kafka streaming engine that decouples communication inside

[4] https://github.com/FIWARE/context.Orion-LD
[5] https://github.com/stellio-hub/stellio-context-broker

the broker and a TimescaleDB to handle temporal and geospatial queries as well as for managing subscriptions and subsequent notifications.

- Scorpio[6]: Scorpio is an NGSI-LD compliant context broker developed by NEC Laboratories Europe and NEC Technologies India. It implements the full NGSI-LD API as specified by the ETSI Industry Specification Group on cross cutting Context Information Management (ETSI ISG CIM). Its main components are, a Kafka streaming engine that decouples communication inside the broker, a PostgresDB as storage and PostGIS, spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location queries to be run in SQL.

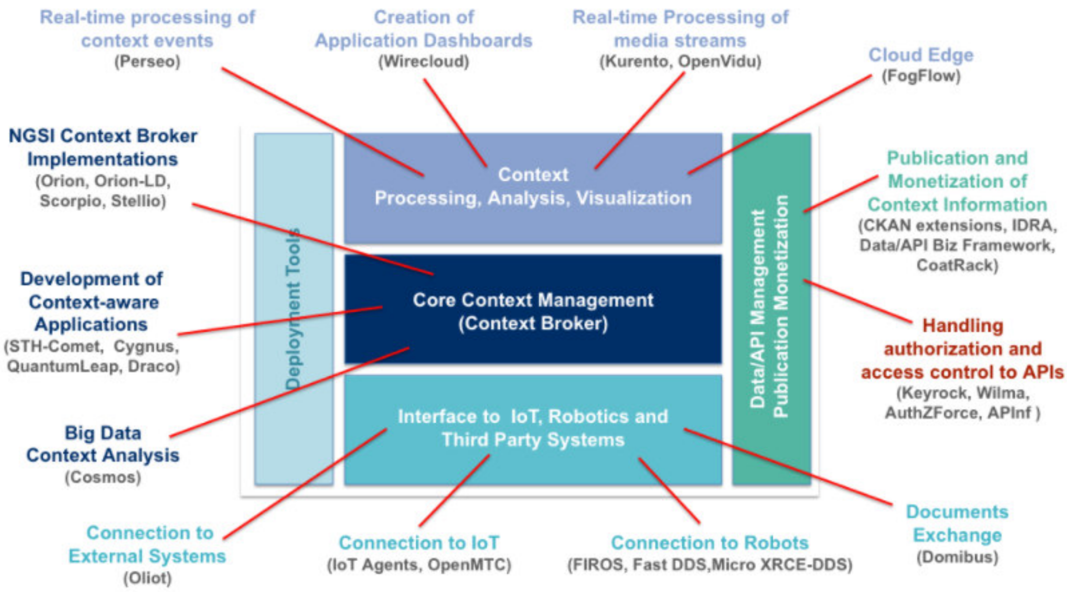

Figure 4: Fiware modules

(source: https://github.com/FIWARE/catalogue)

3.4.3.2 History Context Manager

The History Context Manager captures updates on context information managed by the Context Broker and produces a stream of context data which can then be stored into a specific persistent data sink.

The available History Context Managers are:

- STH Comet[7]: Generic Enabler brings the means for storing a short-term history of context data (typically months) on MongoDB.

[6] https://github.com/ScorpioBroker/ScorpioBroker
[7] https://fiware-sth-comet.readthedocs.io/en/latest/

- Cygnus[8]: Generic Enabler brings the means for managing the history of context that is created as a stream of data which can be injected into multiple data sinks, including some popular databases like PostgreSQL, MySQL, MongoDB or AWS DynamoDB as well as BigData platforms like Hadoop, Storm, Spark or Flink.
- Draco[9]: Generic Enabler is an alternative data persistence mechanism for managing the history of context. It supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic and also offers an intuitive graphical interface.
- Cosmos[10]: Generic Enabler enables simpler Big Data analysis over context integrated with popular Big Data platforms (Spark and Flink).
- QuantumLeap[11]: Generic Enabler supports the storage of context data into a time series database (CrateDB and Timescale).

3.4.3.3 Interfaces

To interface with IoT, robotics, and other third-party systems a set of GE exists that can stablish a bi-directional communication with the external system and the platform: gathering valuable context information and trigger external system actuations in response to context updates. The most relevant GEs are:

- IDAS: Generic Enabler offers you a wide range of IoT Agents making it easier to interface with devices using the most widely used IoT protocols (LWM2M over CoaP, JSON or UltraLight over HTTP/MQTT, OPC-UA, Sigfox or LoRaWAN)
- OpenMTC[12]: Incubated Generic Enabler brings an open source implementation of the OneM2M standard. A northbound interface with the Orion Context Broker is implemented as part of the product.
- Fast DDS[13]: Incubated Generic Enabler has been adopted as default middleware in ROS2, the widely known Robot Operating System, therefore it helps to interface with ROS2-based robotics systems.
- Micro XRCE-DDS: Incubated Generic Enabler is a lite version of the DDS middleware, adapted to run in extremely constrained resource devices. (e.g. micro-controllers).
- FIROS[14]: Incubated Generic Enabler works as a translator between the robotics domain and the cloud, transforming ROS messages into NGSI v2 and vice versa.

[8] https://fiware-cygnus.rtfd.io/
[9] https://fiware-draco.readthedocs.io/en/latest/
[10] https://github.com/ging/fiware-cosmos
[11] https://github.com/orchestracities/ngsi-timeseries-api
[12] https://fiware-openmtc.readthedocs.io/en/latest/
[13] https://github.com/eProsima/Fast-DDS
[14] https://github.com/iml130/firos

- Domibus[15]: Incubated Generic Enabler helps users to exchange electronic data and documents with one another in a reliable and trusted way.
- Oliot[16]: Incubated Generic Enabler is a mediation gGateway which translates information from NGSI based platforms to EPCIS based IoT platforms.

3.4.3.4 Visualization

FIWARE provides 2 different solutions for visualization:

- WireCloud[17]: Brings a powerful web mashup platform making it easier to develop operational dashboards which are highly customizable by end users.

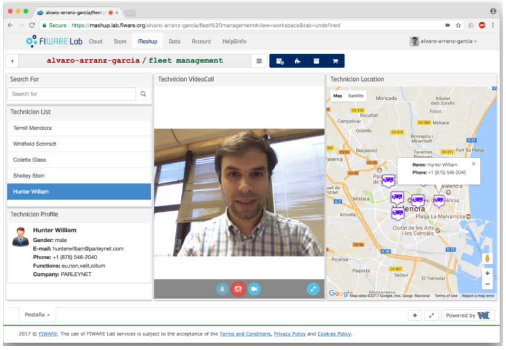

Illustration 2: WireCloud dashboard

(source: fiwaretourguide)

- Knowage[18]: is a powerful Business Intelligence platform enabling to perform business analytics over traditional sources and big data systems.

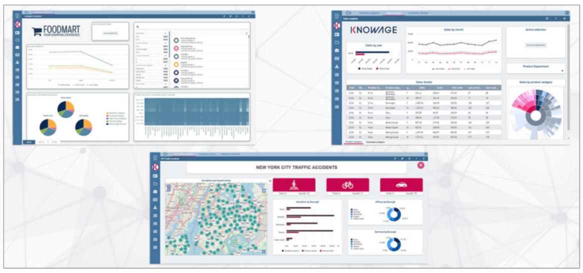

Illustration 3: Knowage dashboard

(source: fiwaretourguide)

[15] https://github.com/cefedelivery/domibus
[16] https://github.com/yalewkidane/FIWARE_EPCIS_Mediation_Gateway
[17] https://wirecloud.readthedocs.io/en/stable/
[18] https://knowage.readthedocs.io/en/latest/

3.4.3.5 *Security*

FIWARE offers many GE to grant security in the application:

- The Keyrock[19] Identity Management Generic Enabler brings support to secure and private OAuth2-based authentication of users and devices, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) and Identity Federation across multiple administration domains.
- The Wilma[20] proxy Generic Enabler brings support of proxy functions within OAuth2-based authentication schemas. It also implements PEP functions within an XACML-based access control schema.
- The AuthZForce PDP/PAP[21] Generic Enabler brings support toPDP/PAP functions within an access control schema based on the XACML standard.

3.4.3.6 *Data models*

The FIWARE Foundation, TM Forum, and IUDX are leading a joint collaboration initiative to support the adoption of a reference architecture and compatible common data models[22]. that underpin a digital market of interoperable and replicable smart solutions in multiple sectors, starting with Smart Cities.

A smart data model includes three elements: The schema, or technical representation of the model defining the technical data types and structure, the specification of a written document for human readers, and the examples of the payloads for NGSIv2 and NGSI-LD versions.

All data models are public and of royalty-free nature.

Relevant data models for AquaSPICE project are: Water Network Management EPANET[23], Water Quality[24], OpenChannelManagement[25], WasteWater[26]

3.4.4. MainFlux

Mainflux [58] is modern, scalable, secure open source and patent-free IoT cloud platform written in Go.

As described in the project documentation [59], it accepts user and thing connections over various network protocols (i.e. HTTP, MQTT, WebSocket, CoAP), thus making a seamless bridge between them. It is used as the IoT middleware for building complex IoT solutions.

---

[19] https://fiware-idm.readthedocs.io/en/latest/
[20] https://fiware-pep-proxy.readthedocs.io/en/latest/
[21] https://authzforce-ce-fiware.rtfd.io/
[22] https://www.fiware.org/smart-data-models/
[23] https://github.com/smart-data-models/dataModel.WaterDistributionManagementEPANET/tree/master
[24] https://github.com/smart-data-models/dataModel.WaterQuality/tree/master
[25] https://github.com/smart-data-models/dataModel.OpenChannelManagement/tree/master
[26] https://github.com/smart-data-models/dataModel.WasteWater/tree/master

- Protocol bridging (i.e. HTTP, MQTT, WebSocket, CoAP)
- Device management and provisioning
- Fine-grained access control
- Platform logging and instrumentation support
- Container-based deployment using Docker

The platform is built around 3 main entities: users, things and channels.

- User represents the real (human) user of the system. It is represented via its e-mail and password, which he uses as platform access credentials to obtain an access token. Once logged into the system, user can manage his resources (i.e. things and channels) in CRUD fashion and define access control policies by connecting them.
- Thing represents devices (or applications) connected to Mainflux that uses the platform for message exchange with other "things".
- Channel represents a communication channel. It serves as message topic that can be consumed by all of the things connected to it.

Mainflux platform can be run on the edge as well. Deploying Mainflux on a gateway makes it able to collect, store and analyze data, organize and authenticate devices.

4. AquaSPICE RTM Platform Requirements

To guarantee that the designed RTM platform will be suitable for all case studies, Task 1 has been devoted to the recollection of requirements. This recollection has been carried out through several activities:

- CS Questionarie: As reported in D1.2 – User Analysis, Use Cases Requirements and Quality Criteria a questionnaire intended to collect input from each CS about current situation, to be situation and constraints was circulated. This questionarie included several questions regarding existing and envisioned digital infrastructure as well as needs for digitalization.
- CS workshops: As part of WP4, several rounds of technical workshops have been conducted for each case study. These workshops, where technical representatives of the CS and partners of AquaSPICE attended, have been centered at discussing the water treatment processes to be implemented during the project and analyzing the necessities with respect to the digital sphere of the project (monitoring, data analytics, optimization, etc..).
- Bi-lateral meetings: Apart from the questionnaire and individual CS workshops, several bi-lateral meetings between WP3 members and CS leaders have been conducted to clarify the necessities regarding data management.

As a result of the said activities, the current section gives a detailed description of the collected requirements.

Ref.	Title
R_001	Device management
Requirement description	
The platform must offer the possibility to create/list/modify/delete devices. A device represents a physical/logical component of a production process, or an element related to it. A device must have a unique identifier associated to it. This identifier will be then used for updating information and retrieving it. A device must have a categorization associated to it. This categorization will help data producers and consumers know what attributes can be associated to a device. It must be able to associate a variable number of attributes to a device. Each attribute with its own type.	

Ref.	Title
R_002	Data types
Requirement description	
The platform must be able to handle information structured in arbitrary ways. Basic data types such as character arrays, numbers, booleans nested in arbitrary complex structures should be supported.	

Ref.	Title
R_003	Import static data sources
Requirement description	
For connectivity and/or security reasons, some CS won't be able to provide real time data to the RTM Platform. In contrast, they will record relevant information into permanent media (CSV, Excel, database, etc…). RTM Platform must thus, offer the necessary mechanisms to manually import these datasets. This information will be then mapped to devices and their attributes.	

Ref.	Title
R_004	Ingest real time data
Requirement description	

The platform must facilitate the integration of real time data coming from different sources (sensors, weather prediction services, etc…). This information will be then mapped to devices and their attributes.

Ref.	Title
R_005	Data updates subscription mechanisms
Requirement description	
Clients of the RTM Platform must be able to subscribe to information updates. Once subscribed, the platform should handle the notification of such changes when they occur.	

Ref.	Title
R_006	Data query mechanisms
Requirement description	
Clients of the RTM Platform must be able to query the data ingested by the platform. The data query mechanism must provide sufficient filter capabilities so that, clients can request data based on the device they belong to and the moment the data were generated.	

Ref.	Title
R_007	Visualization of data
Requirement description	
The platform must provide a tool to allow users to interact with the stored data and create visualizations of it using appropriate graphical elements (graph bars, tables, pie charts, etc…).	

Ref.	Title
R_008	Data persistence
Requirement description	

Data must be persisted for at least 10 years for online queries. After this period, data must be stored perpetually for offline query using static files or similar technologies. The offline data will require a manual load process before being able to query it.

Ref.	Title
R_009	Scalability
Requirement description	
Given the data retention requirement, it is foreseen that the storage mechanism can potentially grow past the terabytes point. For this reason, it is required that the storage technology can handle this amount of data. To guarantee that this requirement is meet, the persistence system must offer the possibility to scale horizontally.	

Ref.	Title
R_010	Deployment
Requirement description	
The designed architecture should allow its deployment in any domestic servers as well as public or private cloud infrastructure.	

Ref.	Title
R_011	Licensing
Requirement description	
Software used for the platform must come free of charge forever.	

Ref.	Title
R_012	Standard protocols and formats
Requirement description	
Standard data models linked with the AquaSPICE ontology have to used for structuring the information.	

Ref.	Title
R_013	Access control
Requirement description	
The platform must enforce access control for all the interactions with the clients: device management, historical data query and notification subscriptions. Access control mechanism will allow manager to define a context-based set of rules that will allow related stakeholders to determine consent provision aspects and preferences to be applied in either automatic or manual manner.	

Ref.	Title
R_014	Monitoring and management of the platform
Requirement description	
The platform will have the necessary tools and services for monitoring, both of the underlying infrastructure, as well as of the distributed tools, as well as of the different operational services, in order to ensure the correct Operation of this and the ability to detect early possible failures.	

Ref.	Title
R_015	Reliability
Requirement description	
The platform must be consistent, with good control and management of errors.	

Ref.	Title
R_016	Resilience
Requirement description	
The platform must have the capability to recover from a fault.	

5. AquaSPICE Real Time Monitoring Platform Design

This section is devoted at presenting the proposed architecture for the RTM Platform. First, a brief introduction of the main components and technologies is presented. Next, the reasoning behind the selection of different technologies for the platform is presented. Later, a deep analysis on how all the identified requirements will be met by the platform is given.

5.1. Architecture

Once having analysed different platform alternatives (Monitoring platforms) and the requirements for AquaSPICE RTM Platform (AquaSPICE RTM Platform Requirements) it has been decided to use FIWARE project. This decision is because FIWARE is the only option that exhibits compliance with NGSI-LD standards (as demanded by requirement R_012). Apart from the requirement fulfilment, FIWARE is considered to have a collaborative and mature ecosystem of developers (more than 8K), innovation Hubs (21), accelerators, cities, top members like Atos, Engineering, Red Hat, NEC, Telefónica and Trigyn Technologies among and more than 1000 SMEs and startups.[27]

Illustration 4: RTM Platform architecture diagram depicts the basic structure of the RTM Platform and its integration within the AquaSPICE ecosystem. The platform is primarily composed of five components (i) Context broker, (ii) Historical data storage, (iii) security provider (iv) data model server and (v) Dashboarding tool.

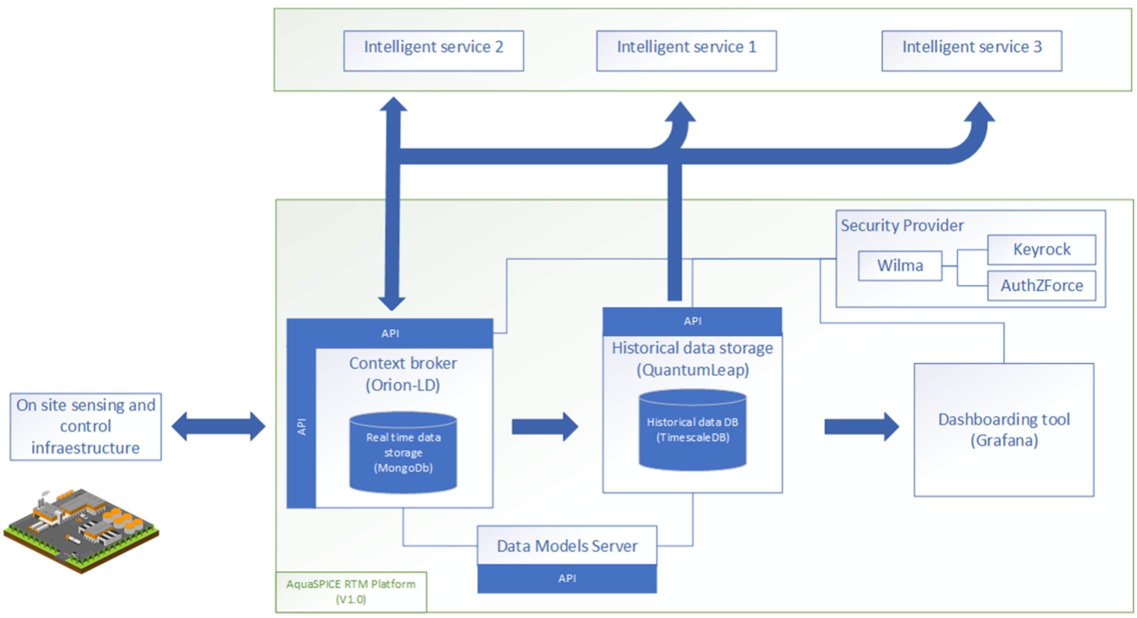

Illustration 4: RTM Platform architecture diagram

[27] https://www.fiware.org/wp-content/uploads/FIWARE_PressKit_English.pdf

The context broker (Orion-LD) is responsible for managing the lifecycle of context information: entities and their attributes. Using the APIs provided by the context broker, one can create context elements, manage them though updates, update their attributes, perform queries to retrieve their status, and subscribe to context changes. On Its heart, the context broker utilizes a database (MongoDB) to store the last version of the context information. Operations against this API are first validated with the security provider to provide authentication (identify the user performing the request) and authorization (validate that the user has appropriate permissions to perform the request).

On the one hand, it is expected that this API will be utilized by the on-site control infrastructure to i) provide sensor readings and ii) subscribe to actuation orders generated by the intelligent services (e.g: to start/stop a water pump). For each data source and protocol that is required to be supported, an adapter can be developed (through the specific development of this or the use of tools/libraries/ external clients), for the publication of data directly on the platform.

On the other hand, the intelligent services layer (optimization algorithms, life cycle analysis models, etc…) will have access to the API for querying current context information, subscribing to new events, and publishing data (e.g: virtual sensor data, on site infrastructure action requests, etc..).

The Historical data storage module is responsible for storing, in a persistent manner, all the changes occurred to the context (changes in entities and associated attributes). As described before, the context broker only holds the current status of the context, thus, no historical data is managed by this context broker. To provide this long-term storage, the historical data storage module (QuantumLeap), subscribes to context changes in the context broker and persists them in a dedicated database (Timescale). The historical data storage module will offer a securitized REST API to access the historical data. It is expected that intelligent services will use this API to retrieve data for its batch analysis.

The security provider is composed by three separated components:

- PEP-Proxy Wilma: Responsible for intercepting API requests and communicate with Keyrock for authentication and AuthZForce for authorization.
- Keyrock: Responsible for identity management, Auth2-based authentication of users and devices, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) and Identity Federation across multiple administration domains.
- AuthZForce: Responsible for enforcing security policies expressed using the XACML standard.

Once a request to any exposed REST services is intercepted by PEP-Proxy Wilma the authentication information provided by the client (user credentials, session token, etc..) will be extracted and diverteded to Keyrock. Keyrock will validate the provided credentials and return the requester information. Then, a call from Wilma to AuthZForce will trigger the authorization of the request against security policies defined using XACML notation. If the process succeeds, the original request will be proxied to the destination API. If not, an authentication/authorization HTTP error will be returned to the originator of the request.

All the information exchanged between the platform and the external systems (on site infrastructure and intelligent services) will be modeled using JSON-LD format. To be able to represent all the concepts of the different CS, appropriate data models aligned with the AquaSPICE onthology will be integrated or developed into the platform. The Data Model Server (Nginx) will act as a repository of these models, offering a public unrestricted API for retrieving them.

The Dashboarding tool (Grafana) allows the platform users to create visualizations of the gathered data. These visualizations are ordered in dashboards composed of many panels displaying certain aspects of the data in a variety of representations (tables, plots, maps, etc..).

5.2. Technology selection

Below, a brief motivation for the selection of the different technologies that compose the proposed architecture is given:

Context broker

Among different alternatives offered by FIWARE, Orion-LD has been chosen because i) it supports NGSI-LD standard and ii) is the most mature context broker. The selection of Orion-LD implies the usage of MongoDB as the short-term internal persistence mechanism.

Historical data storage module

Given the fact that the data collected from pilot plants will be primarily time-series oriented (sensor readings, events, etc…), it has been determined that a time-series database is appropriate for the application. Among other different open-source time-series databases available (Timescale[28], Apache Druid[29], InfluxDB[30], OpenTSDB[31], Graphite[32], etc…) it has been concluded that Timescale is a good choice given that it will be able to cope with the expected volume of information and requires little administration efforts in comparison with other technologies.

28 https://www.timescale.com/
29 https://druid.apache.org/
30 https://www.influxdata.com/products/influxdb-overview/
31 http://opentsdb.net/
32 https://graphiteapp.org/

Timescale is an open-source time-series database engine based on PostgreSQL. Starting from vanilla PostgresSQL engine, Timescale adds several improvements to optimize the storage of time series data on a massive scale (up to tens of TB uncompressed[33]).

QuantumLeap has been selected among other alternative context broker-historical data storage links offered by FIWARE (STH Comet, Cygnus, Draco, etc..) because is the only one designed with timescale compatibility (although any other technology with relational database support would also suffice, as Timescale is, at its heart, a PostgresSQL).

Security provider

The triad of components (Wilma, Keyrock and AuthZForce) is the de-facto standard for the development of authentication/authorization mechanisms in FIWARE architectures.

Model Server

Nginx[34] is an open-source web server with capabilities of reverse proxy, HTTP cache, and load balancer. Nginx is built to offer low memory usage and high concurrency. Developed since 2004, Nginx is gaining traction in comparison with traditional web servers like Apache[35]. The motivation for the selection resides principally on the simplicity of its configuration.

Dashboarding tool

Although FIWARE offers two alternatives for the visualization of stored data (WireCloud and Knowage), it has been decided to opt for Grafana[36] as a dashboarding tool. Grafana is an open-source dashboarding tool capable of connecting a plethora of different storage backends (relational databases, graphite, OpenTSDB, Elasticsearch, etc…). In comparison with the other tools, Grafana is a much mature product with a strong community of enthusiasts developing custom visualization components. Moreover, Grafana is capable to interact with an external OAuth server for users authentication. This makes it possible to link it with Keyrock.

5.3. Requirements fulfilment analysis

Next, a detailed description of how the identified requirements are going to be met by the proposed architecture is given:

Ref.	Title
R_001	Device management

[33] https://docs.timescale.com/timescaledb/latest/overview/#relational-and-time-series-together
[34] https://www.nginx.com/
[35] https://kinsta.com/blog/nginx-vs-apache/
[36] https://grafana.com/

Requirement description
The platform must offer the possibility to create/list/modify/delete devices for the monitoring of these. A device represents a physical/logical component of a production process, or an element related to it. A device must have a unique identifier associated to it. This identifier will be then used for updating information and retrieving it. A device must have a categorization associated to it. This categorization will help data producers and consumers know what attributes can be associated to a device. It must be able to associate a variable number of attributes to a device. Each attribute with its own type.
Requirement fulfilment justification
Orion-LD context broker uses NGSI-LD as a representation standard. NGSI-LD modeling is centered around entities and attributes associated to these entities. Entities have an unique ID and can have a type associated. The selected context broker (Orion-LD) offers a REST API to create/list/modify/delete these entities and associated attributes. The type of an entity can be then linked to a data model[37].

Ref.	Title
R_002	Data types
Requirement description	
The platform must be able to handle information structured in arbitrary ways. Basic data types such as character arrays, numbers, booleans nested in arbitrary complex structures.	
Requirement fulfilment justification	
Orion-LD context broker uses NGSI-LD, an entity can have an arbitrary set of properties, each with its own type. Attribute nesting is also permitted[2].	

Ref.	Title
R_003	Import static data sources
Requirement description	
For connectivity and/or security reasons, some CS won't be able to provide real time data to the RTM Platform. In contrast, they will record relevant information into	

[37] https://github.com/FIWARE/context.Orion-LD/blob/develop/doc/manuals-ld/entities-and-attributes.md

permanent media (CSV, Excel, database, etc…). RTM Platform must thus, offer the necessary mechanisms to manually import these datasets. This information will be then mapped to devices and their attributes. The requirements and architecture for an underlying semantic framework to allow normalisation and integration of in situ sensor data with other data sources in the network will also be provided.
Requirement fulfilment justification
Custom scripts will be developed as needed to integrate information coming from static sources. These scripts will use the REST API provided by the context broker.

Ref.	Title
R_004	Ingest real time data
Requirement description	
The platform must facilitate the integration of real time data coming from different sources (sensors, weather prediction services, etc…). This information will be then mapped to devices and their attributes.	
Requirement fulfilment justification	
The proposed RTM Platform offers a REST API that allows any authorized client to interact with the context (create entities, update readings, etc..). This API will be the standard for ingesting real time data. Using it, on-site existing sensing and control infrastructure will publish the collected data from the monitored process. If, due to any technical reason, it is not possible to use this mechanism for a CS. A custom interface will be developed to match the CS needs. In this situation, the utilization of existing FIWARE generic enablers for data acquisition will be studied.	

Ref.	Title
R_005	Data updates subscription mechanisms
Requirement description	
Clients of the RTM Platform must be able to subscribe to information updates. Once subscribed, the platform should handle the notification of such changes when they occur.	
Requirement fulfilment justification	
The proposed RTM Platform REST API that allows any authorized client to create subscriptions on elements and receive notifications when changes to this element	

occurs. To do so, client must indicate the data it is interested in (specific entities, entity types, attributes, etc..) and a callback URL where notifications must be received. Once a subscription is stablished, the context broker watches for changes in the subscribed entities and sends REST PUSH commands to the subscriber with the details[38]. Apart from this context-broker callback-based subscription mechanism, FIWARE, and thus, AquaSPICE architecture, offers the possibility to publish all data streams into a PUB-SUB broker (Kafka). Using this mechanisms, platform clients (presumably from the intelligent services layer), will be able to subscribe to data changes. This mechanism will be primarly used in situations where high data throughput is necessary or when a REST callback endpoint cannot be provided by the client[39].

Ref.	Title
R_006	Data query mechanisms
Requirement description	
Clients of the RTM Platform must be able to query the data ingested by the platform. The data query mechanism must provide sufficient filter capabilities so that, clients can request data based on the device they belong to and the moment the data were generated.	
Requirement fulfilment justification	
QuantumLeap component from FIWARE architecture implements an API[40] that allows clients to query historical data using REST standard. The queries can be filtered by device id, device type, attributes, and time range. Moreover, aggregation functions (max, min, avg, etc..) can be applied to the retrieved data.	

Ref.	Title
R_007	Visualization of data
Requirement description	

[38] https://documenter.getpostman.com/view/513743/SzfDvjN5

[39] https://fiware-cygnus.readthedocs.io/en/latest/cygnus-ngsi/integration/orion_cygnus_kafka/index.html

[40] https://app.swaggerhub.com/apis/smartsdk/ngsi-tsdb/0.8.2

The platform must provide a tool to allow users to interact with the stored data and create visualizations of it using appropriate graphical elements (graph bars, tables, pie charts, etc…).
Requirement fulfilment justification
Grafana will allow users to create dashboards form information stored in the historical persistence module (Timescale).

Ref.	Title
R_008	Data persistence
Requirement description	
Data must be persisted for at least 10 years for online queries. After this period, data must be stored perpetually for offline query using static files or similar technologies. The offline data will require a manual load process before being able to query it.	
Requirement fulfilment justification	
Whilst context-broker only keeps the last version of each entity, in the proposed architecture, QuantumLeap will manage the long term persistence of data into Timescale. Timescale offers the possibility of storing tens of TBs of data uncompressed or thousands if compressed in a single server. Moreover, it also offers the possibility to distribute the data across several servers. Finally, once the 10 years period has been completed, data can be exported to CSV files using PostgreSQL tools.	

Ref.	Title
R_009	Scalability
Requirement description	
Given the data retention requirement, it is foreseen that the storage mechanism can potentially grow past the terabytes point. For this reason, it is required that the storage technology can handle this amount of data. To guarantee that this requirement is meet, the persistence system must offer the possibility to scale horizontally.	
Requirement fulfilment justification	
As described for R_008: Data persistence, Timescale offers the possibility of storing tens of TBs of data uncompressed or thousands if compressed in a single server. Moreover, it also offers the possibility to distribute the data across several servers.	

Ref.	Title
R_010	Deployment
Requirement description	
The designed architecture should allow its deployment in any domestic servers as well as public or private cloud infrastructure.	
Requirement fulfilment justification	
The designed infrastructure can be deployed in any hardware capable of executing a mainstream Linux distribution (Ubuntu, Centos, Debian, etc…) or execute Docker virtualization engine.	

Ref.	Title
R_011	Licensing
Requirement description	
Software used for the platform must come free of charge forever.	
Requirement fulfilment justification	
Timescale is licensed on Apache 2.0 license (for the opensource version) and Tiemescale License (for the full free version). The most restrictive license (the former one) grants free usage "as long as you are not offering TimescaleDB as a hosted Database-as-a-Service"[41] Orion-LD and Grafana are distributed under AGPLv3 license[42] which only requires making publicly available the source code (if modified).[43] QuantumLeap, Keyrock and Wilma are distributed under MIT License[44] Nginx is distributed under 2-clause BSD-like license[45]	

Ref.	Title
R_012	Standard protocols and formats

[41] https://www.timescale.com/legal/licenses
[42] https://www.gnu.org/licenses/agpl-3.0.en.html
[43] https://grafana.com/licensing/
[44] https://opensource.org/licenses/MIT
[45] http://nginx.org/LICENSE

Requirement description
Standard data models linked with the AquaSPICE ontology have to used for structuring the information.
Requirement fulfilment justification
Orion-LD context broker uses NGSI-LD data models. NGSI-LD is an an extended subset of JSON-LD for use with context management systems, its payloads are encoded as linked data using JSON. In AquaSPICE project, data models used for each CS will be aligned with the ontology defined in WP1.

Ref.	Title
R_013	Access control
Requirement description	
The platform must enforce access control for all the interactions with the clients: device management, historical data query and notification subscriptions. Access control mechanism will allow manager to define a context-based set of rules that will allow related stakeholders to determine consent provision aspects and preferences to be applied in either automatic or manual manner.	
Requirement fulfilment justification	
The security provider is composed by three separated components: • PEP-Proxy Wilma: Responsible for intercepting API requests and communicate with Keyrock for authentication and AuthZForce for authorization. • Keyrock: Responsible for identity management, Auth2-based authentication of users and devices, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) and Identity Federation across multiple administration domains. • AuthZForce: Responsible for enforcing security policies expressed using the XACML standard. Once a request to any exposed REST services is intercepted by PEP-Proxy Wilma the authentication information provided by the client (user credentials, session token, etc..) will be extracted and diverteded to Keyrock. Keyrock will validate the provided credentials and return the requester information. Then, a call from Wilma to AuthZForce will trigger the authorization of the request against security policies defined using XACML notation. If the process succeeds, the original request will be	

proxied to the destination API. If not, an authentication/authorization HTTP error will be returned to the originator of the request.

Ref.	Title
R_014	Monitoring and management of the platform
Requirement description	
The platform will have the necessary tools and services for monitoring, both of the underlying infrastructure, as well as of the distributed tools, as well as of the different operational services, in order to ensure the correct Operation of this and the ability to detect early possible failures.	
Requirement fulfilment justification	
Each of the integrated technologies offers self-monitoring capabilities. Moreover, custom made monitoring process to assure that the whole system is working correctly can be developed. This monitoring process can simulate normal operations of the different APIs and watch for expected results. If any anomaly occurs, it can be logged and published to the Timescale database for visualization in Grafana.	

Ref.	Title
R_015	Reliability
Requirement description	
The platform must be consistent, with good control and management of errors.	
Requirement fulfilment justification	
The integrated components have undergone extensive testing both programmatically and in real usage scenarios. This guarantees a good reliability. For any code developed ad-hoc (e.g: data connectors), unit tests and integration tests will be developed to assure software quality.	

Ref.	Title
R_016	Resilience
Requirement description	
The platform must have the capability to recover from a fault.	
Requirement fulfilment justification	

The integrated components have undergone extensive testing both programmatically and in real usage scenarios. This guarantees a good reliability. For any code developed ad-hoc (e.g: data connectors), unit tests and integration tests will be developed to assure software quality.

6. Conclusion and future work

After providing a brief review on the topic of industrial monitoring and automation and analysing different communication technologies and protocols as well as existing monitoring platforms, an initial design of the real time monitoring platform for AquaSPICE project has been given. As described, the proposed RTM platform is based on the open-source initiative FIWARE. Thanks to the modules provided by the FIWARE ecosystem, we can define a modular architecture capable of adapting to the different specificities of each CS, whilst sharing the core concepts and modules. Finally, to validate the proposed architecture, requirements for the platform emanating from the different CS have been collected and a description of their fulfilment by the proposed platform has been given.

As a result of the work done during Task 3.1: RTM System Architecture, the blueprints of the AquaSPICE RTM Platform have been laid out. Continuing with the efforts done so far, "TASK 3.2: Smart Network with Multi-Source Data Acquisition, Harmonisation and Processing Framework" will be centred at providing of specific connectors to heterogenous data sources needed for each CS, specially focusing on the normalization and validation aspects of the process. Moreover, "Task 3.2" will contribute to further refine the specific long term storage approach finally used by the architecture and integrate it. "TASK 3.3: Data Fusion Middleware" will focus on implementing the Context-broker and all the necessary APIs to interact with it. "TASK 3.4: Context-Aware Access Control, Data Integrity and Security Assurance Mechanisms" will undertake the development of a policy model for formally defining context-based access control and consent rulesets for the authentication/authorization layer. "TASK 3.5: Real-Time Water Monitoring Platform" will coordinate the integration of all developed modules and finally "TASK 3.6: Selection, Design, Configuration and Deployment of RTM Platform for Case Study Application" will support the deployment of the resulting RTM platform on each CS.

7. References

[1] R. Kumar, "What is the five layer automation pyramid," [Online]. Available: https://medium.com/world-of-iot/92-what-is-the-five-layer-automation-pyramid-d0ccc1b903c3.

[2] Maxim integrated, "FUNDAMENTALS OF RS-232 SERIAL COMMUNICATIONS," [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html.

[3] National Instruments, "Fundamentals, System Design, and Setup for the 4 to 20 mA Current Loop," 2023. [Online]. Available: https://www.ni.com/es-es/innovations/white-papers/08/fundamentals--system-design--and-setup-for-the-4-to-20-ma-curren.html.

[4] UNE, "UNE-EN IEC 61158-1:2019," [Online]. Available: https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0061941..

[5] Texas Instruments, "The RS-485 Design Guide," [Online]. Available: https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1635841939827&ref_url=https%253A%252F%252Fwww.google.com%252F.

[6] Texas Instruments, "Introduction to the Controller Area Network (CAN)," [Online]. Available: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1635837678726.

[7] CiA, "CANopen – The standardized embedded network," [Online]. Available: https://www.can-cia.org/canopen/.

[8] CiA, "DeviceNet – designed for factory automation," [Online]. Available: https://www.can-cia.org/can-knowledge/hlp/devicenet/.

[9] PROFIBUS, "PROFIBUS," [Online]. Available: https://www.profibus.com/technology/profibus.

[10] FieldComm Group, "FOUNDATION FIELDBUS," [Online]. Available: https://www.fieldcommgroup.org/technologies/foundation-fieldbus.

[11] Modbus, "MODBUS PROTOCOL," [Online]. Available: https://modbus.org/specs.php.

[12] T. Carlson, "HMS," 03 May 2022. [Online]. Available: https://www.hms-networks.com/news-and-insights/news-from-hms/2022/05/02/industrial-networks-keep-growing-despite-challenging-times.

[13] Intel, "The BITBUS interconnect serial control bus specification," [Online]. Available: http://www.bitbus.org/dnl/BITBUS%20Interconnect%20Specification.PDF.

[14] Siemens, "AS-Interface - Introduction and Basics," [Online]. Available: https://cache.industry.siemens.com/dl/files/856/1171856/att_961903/v1/SYH_asi-grundlagen_76_201805231008141616.pdf.

[15] Wikipedia, "AS-Interface," [Online]. Available: https://en.wikipedia.org/wiki/AS-Interface.

[16] InterBus, "InterBus," [Online]. Available: http://www.interbusclub.com/index_en.html.

[17] J. Jasperneite, "INTERBUS Means Speed, Connectivity, Safety," in *The Industrial Communication Technology Handbook*, CRC-Press, 2005, pp. 10-20.

[18] ETHERNET POWERLINK, "ETHERNET POWERLINK," [Online]. Available: https://www.ethernet-powerlink.org/.

[19] Modbus, “MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE,” [Online]. Available: https://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

[20] EtherCAT Technology Group, “EtherCAT,” [Online]. Available: https://www.ethercat.org/default.htm.

[21] ODVA, “Ethernet-ip,” [Online]. Available: https://www.odva.org/technology-standards/key-technologies/ethernet-ip/.

[22] PI, “PROFINET TECHNOLOGY,” [Online]. Available: https://us.profinet.com/technology/profinet/.

[23] IEEE 802.11TM, “IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS,” [Online]. Available: https://www.ieee802.org/11/.

[24] Bluetooth, “Bluetooth® Wireless Technology,” [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[25] Wikipedia, “Radio-frequency identification,” [Online]. Available: https://en.wikipedia.org/wiki/Radio-frequency_identification.

[26] NFC Forum, “What Is NFC,” [Online]. Available: https://nfc-forum.org/what-is-nfc/.

[27] Wikipedia, “Near-field communication,” [Online]. Available: https://en.wikipedia.org/wiki/Near-field_communication.

[28] Wikipedia, “Cellular network,” [Online]. Available: https://en.wikipedia.org/wiki/Cellular_network.

[29] Wikipedia, “GSM,” [Online]. Available: https://en.wikipedia.org/wiki/GSM.

[30] Wikipedia, “LTE,” [Online]. Available: https://en.wikipedia.org/wiki/LTE_(telecommunication).

[31] GSMA, “New GSMA Study: 5G to Account for 15% of Global Mobile Industry by 2025 as 5G Network Launches Accelerate,” [Online]. Available: https://www.gsma.com/newsroom/press-release/new-gsma-study-5g-to-account-for-15-of-global-mobile-industry-by-2025/.

[32] LoRa Alliance, “What is LoRaWAN® Specification,” [Online]. Available: https://lora-alliance.org/about-lorawan.

[33] Wikipedia, “LoRa,” [Online]. Available: https://en.wikipedia.org/wiki/LoRa.

[34] Z-Wave Alliance, “Z-Wave Specifications,” [Online]. Available: https://z-wavealliance.org/development-resources-overview/specification-for-developers/.

[35] Wikipedia, “Z-Wave,” [Online]. Available: https://en.wikipedia.org/wiki/Z-Wave.

[36] Connectivity standards alliance, “Zigbee,” [Online]. Available: https://zigbeealliance.org/solution/zigbee/.

[37] Wikipedia, “Zigbee,” [Online]. Available: https://en.wikipedia.org/wiki/Zigbee.

[38] Thread Group, “WHAT IS THREAD?,” [Online]. Available: https://www.threadgroup.org/What-is-Thread/Overview.

[39] Wikipedia, “Thread,” [Online]. Available: https://en.wikipedia.org/wiki/Thread_(network_protocol).

[40] SigFox, “SigFox,” [Online]. Available: https://www.sigfox.com/en.

[41] Wikipedia, “Sigfox,” [Online]. Available: https://en.wikipedia.org/wiki/Sigfox.

[42] GSMA, “LTE-M Deployment Guide to Basic Feature Set Requirements,” [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf.

[43] Ericsson, “A Primer on 3GPP Narrowband Internet of Things (NB-IoT),” [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1606/1606.04171.pdf.

[44] Alcatel Lucent, “The LTE Network Architecture”.

[45] A. Foster, “Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper,” PrismTech Corp, 2014.

[46] IBM, “MQTT V3.1 Protocol Specification,” [Online]. Available: https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.

[47] OASIS, “Advanced Message Queuing Protocol (AMQP) Version 1.0 Specification,” [Online]. Available: http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html.

[48] IETF, “The Constrained Application Protocol (CoAP),” [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7252.

[49] Wikipedia, “Constrained Application Protocol,” [Online]. Available: https://en.wikipedia.org/wiki/Constrained_Application_Protocol.

[50] OMG, “Data Distribution Service for Real-Time Systems Version 1.2,” [Online]. Available: https://www.omg.org/spec/DDS/1.2/PDF/.

[51] DDS Foundation, “OMG: Data Distribution Service (DDS),” [Online]. Available: https://www.omgwiki.org/ddsf/doku.php?id=ddsf:public:guidebook:06_append:01_family_of_standards:01_core:data_distribution_service_dds.

[52] Kaa, “Kaa documentation,” [Online]. Available: https://kaaproject.github.io/kaa/docs/v0.10.0/Welcome/.

[53] Kaa, “Kaa IoT Platform Features for Enterprise IoT Projects,” [Online]. Available: https://www.kaaiot.com/products/overview.

[54] Thingsboard, “Thingsboard documentation,” [Online]. Available: https://github.com/thingsboard/thingsboard.

[55] Thingsboard, “Thingsboard architecture,” [Online]. Available: https://thingsboard.io/docs/reference/.

[56] Fiware, “Fiware,” [Online]. Available: https://www.fiware.org/.

[57] FIWARE Foundation, e.V., “FIWARE components,” [Online]. Available: https://www.fiware.org/catalogue/.

[58] Mainflux, “Mainflux project page,” [Online]. Available: https://github.com/mainflux/mainflux.

[59] Mainflux, “Mainflux Components,” [Online]. Available: https://mainflux.readthedocs.io/en/stable/architecture/.

[60] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, P.-A. Champin and N. Lindström, “JSON-LD 1.1: A JSON-based Serialization for Linked Data,” [Online]. Available: https://www.w3.org/TR/json-ld11/. [Accessed 1 2 2023].

[61] T. Salman and R. Jain, “A Survey of Protocols and Standards for Internet of Things,” *Advanced Computing and Communications,* vol. 1, 2017.

[62] V. a. L. P. F. a. N. M. a. R. G. R. Cardellini, “Runtime Adaptation of Data Stream Processing Systems: The State of the Art,” *Association for Computing Machinery,* vol. 54, no. 11s, 2022.

[63] D. Bees, L. Frost, M. Bauer, M. Fisher and W. Li, “NGSI-LD API: for Context Information Management,” ETSI, 2019.

[64] G. Association, “IoT Big Data Framework Architecture v2.0,” 2018.

[65] Thread Group, “Thread Network Fundamentals,” 2020.

[66] Microsoft Azure, “IoT Technologies and protocols,” [Online]. Available: https://azure.microsoft.com/en-us/solutions/iot/iot-technology-protocols/.

[67] FIWARE Foundation, “FIWARE,” [Online]. Available: https://www.fiware.org/.

[68] Solid IT gmbh, “DB-Engines Ranking of Time Series DBMS,” 2 2023. [Online]. Available: https://db-engines.com/en/ranking/time+series+dbms. [Accessed 1 2 2023].

[69] BHV automation, “BHV automation,” 2022. [Online]. Available: https://www.bhv-automation.de/systems/bus-and-interface-systems/?L=1.

[70] G. Proehl, “An Introduction to Near Field Communications,” [Online]. Available: https://www.mouser.es/applications/rfid-nfc-introduction/?_gl=1*q0vf6x*_ga*MTc0OTMyMjU5NS4xNjg1OTYyMzY3*_ga_15W4STQT4T*MTY4NTk2MjM2Ni4xLjAuMTY4NTk2MjM2Ni4wLjAuMA..*_ga_1KQLCYKRX3*MTY4NTk2MjM2Ni4xLjAuMTY4NTk2MjM2Ni42MC4wLjA..

[71] E. Odunlade, “ Introduction to LoRa and LoRaWAN: What is LoRa and How Does It Work,” [Online]. Available: https://circuitdigest.com/article/introduction-to-lora-and-lorawan-what-is-lora-and-how-does-it-work.

[72] E. B. F. C. F. M. Kais Mekki, “A comparative study of LPWAN technologies for large-scale IoT deployment,” *ICT Express,* vol. 5, no. 1, pp. 1-7, 2019.

[73] Wikipedia, “Fieldbus,” [Online]. Available: https://en.wikipedia.org/wiki/Fieldbus.