PROJECT DELIVERABLE REPORT

Greening the economy in line with the sustainable development goals

D2.9 NAIADES Architecture Mid-term

A holistic water ecosystem for digitisation of urban water sector

SC5-11-2018

Digital solutions for water: linking the physical and digital world for water solutions

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820985"

Document Information

Grant Agreement Number	820985	Acronym	NAIADES	
Full Title	A holistic water ecosystem for digitization of urban water sector			
Topic	SC5-11-2018: Digital solutions for water: linking the physical and digital world for water solutions			
Funding scheme	IA - Innovation Action			
Start Date	1st JUNE 2019	**Duration**	36 months	
Project URL	www.NAIADES-project.eu			
EU Project Officer	Alexandre VACHER			
Project Coordinator	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH			
Deliverable	D2.9 NAIADES Architecture Mid-term			
Work Package	WP2 – SDGs and End-user Requirements and Architecture			
Date of Delivery	**Contractual**	M18	**Actual**	M18
Nature	R - Report	**Dissemination Level**	PU-PUBLIC	
Lead Beneficiary	AIMEN			
Responsible Author	Dr. Juan Manuel Fernández Montenegro	Email	Juan.fernandez@aimen.es	
		Phone	+34 697 99 14 97	
Reviewer(s):	KT, ADSYS			
Keywords	Architecture, FIWARE, Interoperability, Modularity, Security			

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
0.1	21/04/2020	AIMEN	Table of Contents
0.2	03/06/2020	AIMEN, CERTH, UDGA, SIMAVI, ADSYS, IHE, ICCS, KT, JSI	Inclusion of partners' contributions
0.3	11/09/2020	AIMEN, SIMAVI	Architecture Updates
0.4	08/10/2020	AIMEN	Modules Update
0.5	02/11/2020	AIMEN, JSI, SIMAVI	Modules Update
	12/11/2020	ADSYS	Internal review

0.6	16/11/2020	AIMEN	ADSYS review applied
	23/11/2020	KT	Internal review
0.7	26/11/2020	AIMEN	KT review applied
1.0	…		Review and Release
1.1	08/06/2021	AIMEN	PO's comments addressed in Section 6 and 3.5
1.2	06/07/2021	AIMEN	Internal review from (UDGA, DISY, ADSYS)
1.3	08/07/2021	AIMEN	Internal review from (IHE, KT and ADSYS)
2.0	09/07/2021	AIMEN	Review and Release of the Revision (Section 6 and 3.5)

Contents

Figures

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Programming Interface
AWS	Amazon Web Services
CDM	Common Data Management
DCA	Data Collection/Aggregation
DDBB	Databases
DSS	Decision Support System
FTP	File Transfer Protocol
GDPR	General Data Protection Regulation
GEs	Generic Enablers
GUI	Graphical User Interface
HCI	Human Computer Interface
HMI	Human Machine Interface
HW	Hardware
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technology
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IISF	Industrial Internet Security Framework
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KSI	Keyless Signature Infrastructure
lsWTP	Lab scaled Water Treatment Plant
NGSI	Next Generation Service Interface
NGSI-LD	Next Generation Service Interface-Linked Data
OS	Operating System
OSI	Open Systems Interconnection
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point

PKI	Public Key Infrastructure
PRP	Policy Retrieval Point
RAMI4.0	Reference Architecture Model for Industry 4.0
RNN	Recurrent Neural Networks
SDK	Software Development Kit
SGAM	Smart Grid Architecture Model
SG-CG	European Smart Grid Coordination Group
SOP	Standard Operating Procedures
SVR	Support Vector Regression
SW	Software
UC	Use Case
UCAx	Use Case Alicante. x = 1,2,3
UCBx	Use Case Braila. x = 1,2
UCCx	Use Case Carouge. x = 1,2
UClsWTP	Use Case lab scale Water Treatment Plant
UI	User Interface
VM	Virtual Machine
WTP	Water Treatment Plant

Summary

This deliverable summarizes the results from T2.6 as a detailed report about the NAIADES architecture defined until M18. It gathers information regarding the architecture structure, including detailed technical specifications of all the components that integrates it and their interactions; and providing justification of every decision based on results from T2.4 and T2.5, which gather the end-user and stakeholders' requirements, the reference scenarios and pilot operation specifications and KPIs. The first version of this document was provided on M12 so the involved partners had a reference document (that formalizes the already shared information) as specified in MS2. This document was updated so as to provide this final midterm report by M18; which will also be used as starting point of D2.10 (Final architecture report).

1 Introduction

1.1 Context and scope

This report defines and describes NAIADES architecture advances based on the defined end-users, stakeholders' requirements, use cases scenarios and NAIADES technological objectives. It will contain the current architecture definition agreed until M18, to be extended in another report by the end of the project, and all partners that are reflected in the description will implement their components/applications based on it. The report contains information about the main system components and how they interact between them and with external interfaces. It also includes information about all the functional and non-functional requirements and the architectural constraints. The validation of these requirements will be mainly included in the final deliverable (D2.10), making available in this document only the validations performed before M18.

1.2 Chapters organization

The deliverable is organized as follows:

- Section 2 describes common Industry 4.0 reference architectures used by system architects to design ad-hoc architectures; and FIWARE since it was agreed NAIADES platform to be FIWARE compliant.
- Section 3 presents NAIADES strategies and functionalities. It introduces the end-users and stakeholders' requirements and the main solutions/technologies provided by the platform.
- Section 4 defines and describes NAIADES architecture from different perspectives: Business, Layers, Communications, Information, Components, Deployment and Security.
- Section 5 defines and describes NAIADES architecture from the Pilots' use cases point of view.
- Section 6 summarizes the main conclusions.

2 Background Reference Architectures

NAIADES architecture is being designed on the foundations of different reference architectures already validated in industrial and smart cities sectors. These reference architectures aim to smartly interconnect the technologies required to establish a smart framework for processes, operations and business management. Furthermore, focusing on the crescent necessity of interoperable systems capable to be easily adapted by new users and able to easily share information with other platforms, NAIADES platform oriented its design towards modularity and compatibility with FIWARE reference architecture (2.3) due to its acknowledged use by several European Projects.

2.1 RAMI4.0

The Reference Architecture Model for Industry 4.0 (RAMI4.0 [1]) specification provides a first draft of the reference architecture for the Industry 4.0 initiative, trying to group different aspects in a common model and to assure the end-to-end consistency of technical, administrative and commercial data created in the ambit of production across the entire value stream and their accessibility at all times.

RAMI 4.0 is essentially focused on the manufacturing process and production facilities, but it also tries to focus on all essential aspects of Industry 4.0. The participants (a field device, a machine, a system, or a whole factory) can be logically classified in this model and relevant Industry 4.0 concepts described and implemented, thus being possible to be adapted to other processes.

RAMI4.0 is a three-dimensional map showing how to approach Industry 4.0 in a structured manner, ensuring understanding among the involved participants (Figure 1). The model adopts, and adapt to Industry 4.0, the basic ideas of the worldwide accepted Smart Grid Architecture Model (SGAM), which was defined by the European Smart Grid Coordination Group (SG-CG).

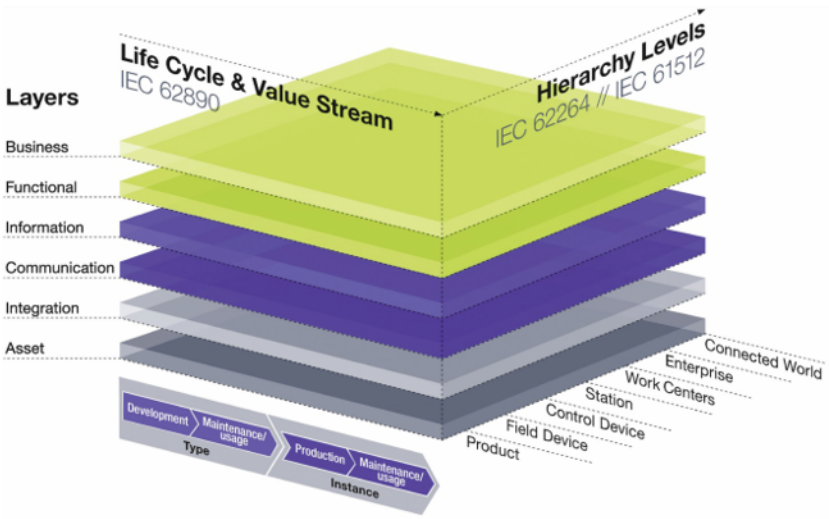

Figure 1. RAMI4.0 reference architecture.

The RAMI 4.0 is a service-oriented architecture that combines all elements and IT components in a layer and life cycle model; and breaks complex processes into simple packages. The 3D model combines:

- Hierarchical Levels (Y Axis): this axis collects the hierarchy levels envisaged by the IEC 62264 international standards on the integration of company computing and control systems.
- Cycle & Value Stream (X Axis): the second axis represents the life cycle of facilities and products. The RAMI4.0 takes the IEC 62890 standard for life cycle management as a reference point to

structure the life cycle. This axis focuses on features able to provide a consistent data model during the whole life cycle of an entity.

- Layers (Z Axis): finally, the vertical axis, represents the various perspectives from the assets (physical world), its digitalization (integration layer), the access to all the necessary data (communication and information layer), the required services/functions of the asset (functional layer), up to the organization and business processes.

2.2 IIRA & IISF

The Industrial Internet Reference Architecture (IIRA [2]) is a standards-based open architecture for IIoT systems, first published in 2015. The IIRA is broadly oriented to drive interoperability among industries, to map applicable technologies, and to guide technology and standard development. The architecture description and representation are generic and at a high level of abstraction allowing Industrial Internet of Things (IIoT) system architects to easily design their systems based on a common framework. The main IIRA categorization is based on the ISO/IEC 42010, introducing the four viewpoints (**Error! Reference source not found.**2): Business, Usage, Functional and Implementation; being the Functional the key viewpoint that groups its guidelines related to interoperability and data exchange.

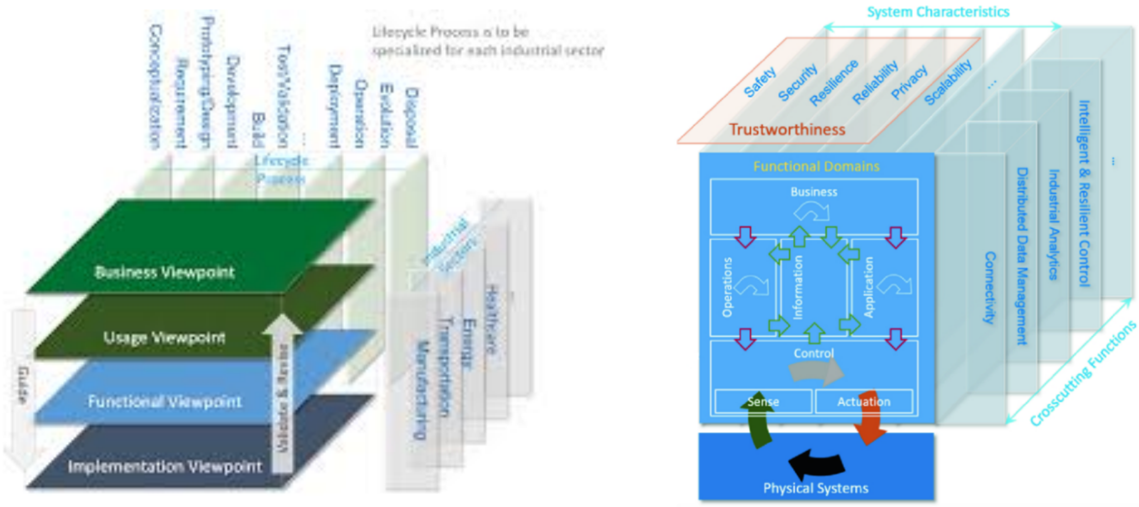

Figure 2. IIRA reference architecture.

A viewpoint comprises conventions framing the description and analysis of specific system concepts.

- **Business Viewpoint**. The business viewpoint focuses on the identification of stakeholders and their business vision, values, and objectives in establishing an IIoT system in its business and regulatory context. It further identifies how the IIoT system achieves the stated objectives through its mapping to fundamental system capabilities.
- **Usage Viewpoint**. The *usage viewpoint* addresses the activities and workflows concerning the usage of an IIoT system to achieve the business objectives. It is typically represented as sequences of activities involving human or logical users (system components), such as: 1) Register new device to the edge gateway; 2) Register the new device in the cloud by querying all the gateways, etc.
- **Functional Viewpoint**. The *functional viewpoint* focuses on the functional components in an IIoT system, their structure and interrelation, the interfaces and interactions between them, and the relation and interactions of the system with external elements in the environment, to support the

usages and activities of the overall system. The Functional Layer is the key part of the IIRA architecture and it can be decomposed into five domains: Control (sensors and actuators), Operations (functions for managing, monitoring and optimization of the control domain), Information (functions to manage and process data), Application (applications logic that realizes specific business functionalities) and Business (functions supporting business processes).

- **Implementation Viewpoint**. The implementation viewpoint is concerned with the technologies and system components required to implement the activities and functions prescribed by the usage and functional viewpoints.

Industrial Internet Security Framework (IISF) is a complement on top of IIRA Functional Viewpoint to provide security mechanisms missing in IIRA reference architecture (Figure 3). They are organized into three layers. The top layer comprises the four core security functions: endpoint protection, communications and connectivity protection, security monitoring and analysis, and security configuration management. These four functions are supported by a data protection layer and a system-wide security model and policy layer.

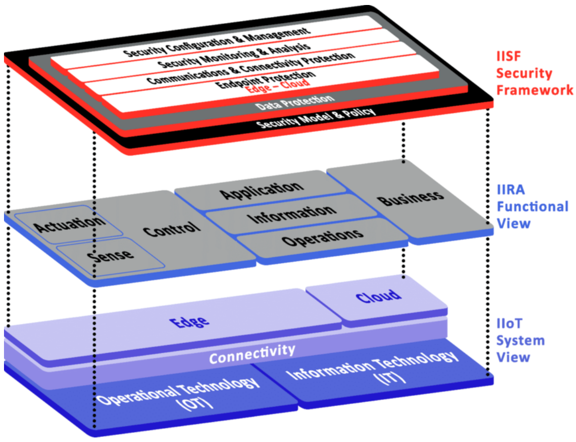

Figure 3. IISF alignment with IIRA Functional Viewpoint

2.3 FIWARE

FIWARE [3] is an open platform that provides smart components, FIWARE Generic Enablers (GEs), for the development of smart solutions. It defines a universal set of standards for context data management. The core of any FIWARE based architecture is the Context Broker (Figure 4). It oversees the management of context information, controlling the updates and access to the current state of the context.

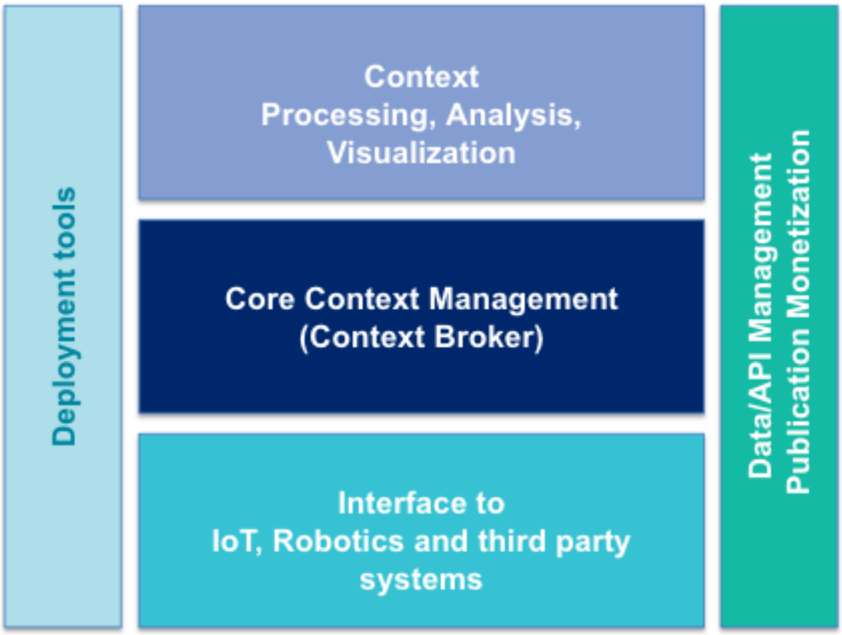

Figure 4. FIWARE scheme to build a solution. Context broker as the core of the solution.

To build a FIWARE compatible architecture it is only required to integrate the Context Broker component. The interfaces for IoT, robotics, human interfaces; and the applications can be built independently, using just third platform components, or combining them with FIWARE Generic Enablers. GEs are general purpose blocks to ease the modular construction of smart solutions. They should provide open interfaces to application developers and to support interoperability with other GEs.

The interactions with the Context Broker are done with the FIWARE Next Generation Service Interface (NGSI) RESTful API. NGSI is intended to manage the lifecycle of the context information (updates, queries, registrations and subscriptions), including linked data (NGSI-LD). NGSI allows the portability of applications between different FIWARE compatible platforms, facilitates the integration of third-party components. Additionally, on modular-based platforms, it facilitates the re-architecting being easier to evolve the architecture as required by the solutions.

FIWARE GEs can be combined to create standalone or cloud-based architectures. There are diverse GEs that provide solutions for each of the blocks of FIWARE reference scheme (Figure 5).

- Core Context Management includes all the GEs in charge of data management. The Orion Context Broker GE is the only mandatory GE and it manages all the current context data; but in order to manage historical data, it is required to include other GEs such as Draco or Cygnus.
- Interface to IoT, Robotics and third-party systems gathers all the GEs that enable the collection of data from and the actuation over the IoT. IDAS GEs provides different IoT Agents GEs to adapt to the different types of communication protocols using on the IoT.
- Context processing, analysis and visualization brings all the GEs that simplify context information analysis and visualization, this is, all the GEs that will provide intelligence to the solution. GEs such as Kurento and Cosmos that enables big data real-time processing.
- Context data/API management, publication and monetization includes the GEs for security (Keyrock and Wilma for authentication of users and devices) and also for the publication and monetization of context data (CKAN extensions and Biz Framework).

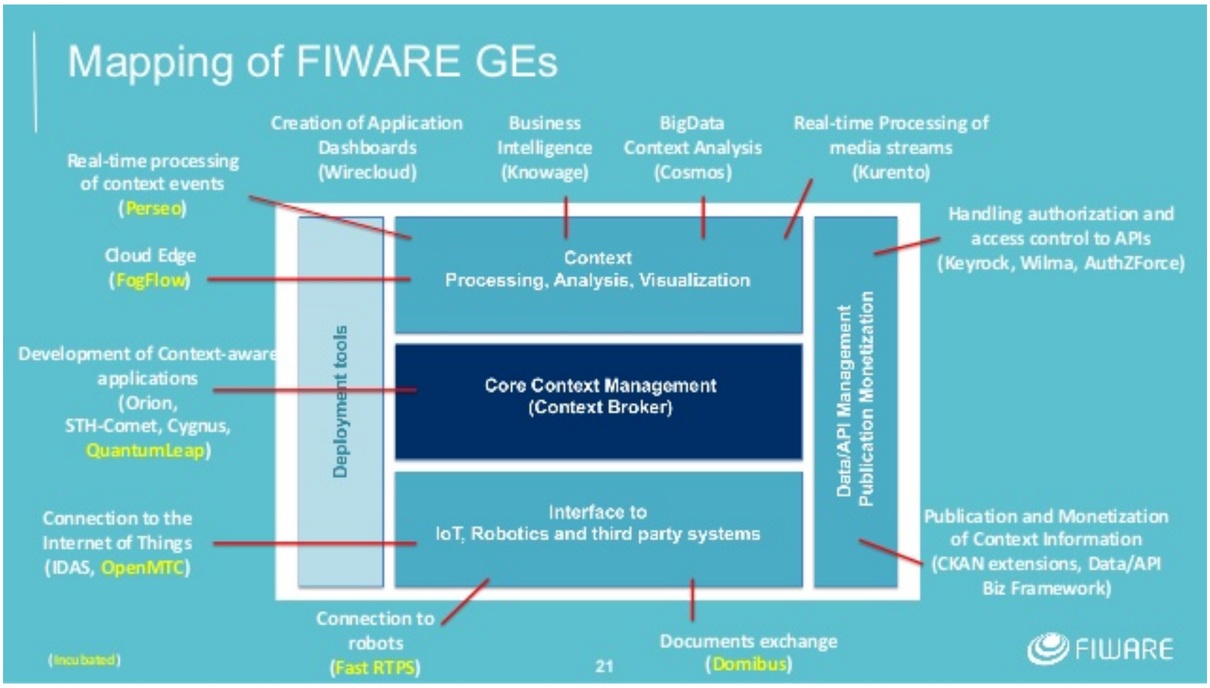

Figure 5. FIWARE GEs mapping.

3 NAIADES main functionalities

3.1 End-users and stakeholders' requirements

NAIADES solution is oriented at solving current issues from three different end-users: two water companies in Alicante and Braila and the council of Carouge. Current architecture was designed to be easily adapted to the requirements presented by the end-users and the capabilities of the partners providing solutions.

3.1.1 Requirements for the proposed Ecosystem:

a. Which management services are the end users missing?

Alicante: In the last twenty years a reduction of 20% in the supplied volume has been observed, even though the number of inhabitants has increased in the same period by about 50,000. Besides, the city has two main wastewater treatment plants, with a combined capacity of 135,000 m^3/day, and a current treated volume daily mean of about 69,000 m^3. One wastewater treatment plant reuses 41.2% of water and the other 35%. Saline water intrusion in the drainage system is of the order of about 12% of the volume that comes to the wastewater treatment plants. AMAEM is missing the following:

1. Further reduction of water demand. To this end, water demand forecast with its associated accuracy, and active awareness campaigns are needed.
2. Reduce the saline water intrusion into the sewer system, by properly accounting for it. This will reduce the current estimated total economic cost of saline intrusion of around 1M Eur per year.

Braila: The city has concerns about water losses in its supply network. Even though the city has a water loss strategy with the purpose of decreasing losses, this problem is still ongoing, and leakage detection is a priority. At the moment, average water losses amount to about 750 l/h/km, and the aim is to reduce it down to 50 l/h/km. CUP is therefore missing the following:

1. Proper monitoring of flows and pressures, in particular in the sector of Radu Negro. They prioritized this area because it is residential and has many building blocks, schools, and government institutions.
2. Water demand monitoring as a way to properly estimate water balance, namely water produced equals water consumed minus water leaked.

Carouge: The user requirements come from the city council of Carouge, not from its water utility. For this reason, the water-related requirements are somewhat different from the other pilots. First, the City Council is in charge of 180 garden boxes on which residents are invited to grow and harvest freely plants and vegetables of their choice, in an ecological way. Second, the City Council is responsible for public health issues that water quality at the city fountains, very popular in summer, could eventually generate. In this regard, their water-related requirements are:

1. Optimise the watering of garden boxes from the point of view of water consumption and of effort of city council staff
2. To improve water quality monitoring at the fountains and provide water quality warnings, related to pH, bacteria and chlorine.

b. End users' ability to cooperate and exchange data regardless of legacy systems technology.

From the sensors' viewpoint, different pilots have different availability of data to address their requirements and current challenges.

Alicante can share most of their current measurements but for privacy reasons they cannot share individuals' metering. This makes it challenging to work on the water demand forecasting unless it is done at a sector scale.

Braila does not have sensors at the interest area, Radu Negru, but they can share data of other similar districts that are currently monitored while the sensors are installed. Additionally, Braila end user is also interested in the optimization of water treatment processes on drinking water treatment plants (WTP) to ensure the water quality on the input of the distribution network; nevertheless, due to the impossibility to share water treatments data for privacy issues, it was necessary to create an extra pilot (laboratory based WTP scaled replica) as a showcase for the NAIADES proposed solution.

Carouge, does not have historical data to quantify the water consumption at the plants' boxes and though there are some historical data available about water quality in the fountains, it is not sufficient for AI solutions. Thus, it is necessary the sensors are installed so to be able to develop the solutions.

3.1.2 Requirements for interconnecting heterogeneous IT instances on top of current physical and digital components, concealing the underlying technologies.

Alicante: Only sensors of conductivity and level to measure salinity must be installed and integrated in their current SCADA system. The other required data will be obtained from their system. The process to retrieve the data from Alicante IT systems (SCADA, monitoring tools) and "push" them into NAIADES platform will be supervised by the Data Collection/Aggregation (DCA) module developed by ADSYS for Alicante. This module (described in section 4.5.1) manages to collect periodically all the data identified in the use cases definition process (UC1-2-3), from different IT sources distributed across Alicante infrastructure. This is done through secured FTP connections (Push/Pull) between the DCA module and the IT resources. These raw data are pre-processed by the DCA (cleansed and pre-formatted) before being adapted to NAIADES common data model. Finally, the data are signed "locally" by the DCA, using KSI credentials and authentication methods and "pushed" into the NAIADES platform through a secure connection requiring token authentication/validation form the access control layer leveraging the access to the NAIADES platform.

Braila: Braila pilot required the installation of four pair of sensors (flow and pressure) for Consumption monitoring and forecast; and two pairs (pressure and noise) for Leakages detection; at DMA Radu Negru district. These sensors will be integrated in their SCADA system, sending measure information into the SCADA systems and after internal processes and conversions, CUP Braila will send the SCADA data to a FTP server (part of SIMAVI's Data Collection/Aggregation) as CSV files with an hourly frequency. DCA will access the FTP server, get and deserialize the CSV files and format raw data following the common Data Models structure, get authorization token from NAIADES security mechanism and send modelled data using the authorization to NAIADES Platform.

Carouge: For the Carouge use case, new IoT sensors are going to be deployed in the aforementioned physical sites (fountains and garden boxes); no legacy IoT sensors are expected to be used within the NAIADES project. The new sensors are going to be connected to the NAIADES platform using pre-existing Carouge's LoRa network operated by the city pilot along with Geneva's public services company called SIG [4]. All data generated by these newly deployed IoT sensors is going to be collected, aggregated, re-formatted (following the common Data Models), and finally signed before being pushed to NAIADES platform. Data will be pushed to the Context Management using its API utilising authentication and authorization features. For more details on this integration please refer to section 4.5.1.

3.1.3 Requirements for realizing and deploying the applications or services in real-world IT environments

Requirement of historical data from pilots that sometimes is inexistent thus requiring rapid implementation of data acquisition so the services can be developed. The requirement of data repository also affects to the extent of the architecture since it is necessary to estimate the total amount of data to be stored to set the repository specifications. The HW and SW requirements are addressed per module on section 4.6.

3.1.4 Security Requirements

a. "Security and privacy by design" implementation of the system architecture

Data being pushed by city pilots to the NAIADES platform through the API shall not contain any information which enables identification of data subjects, what makes collection of data trivially compliant with GDPR requirements and other local regulations about users' data privacy rights. The data being generated and then pushed to NAIADES platform shall be data generated and signed by city pilots (and/or data collectors) only. City pilots, along with their data collecting partners, are responsible for enforcing this requirement.

b. Security mechanisms (proactive/ reactive) and privacy requirements of provided IT applications.

Proactive measures are in place for protecting the cloud-based machines and network configurations. Any change in the data entering the platform needs to be signed, monitoring tools will be in place for verifying data has not been tempered. Attacker vectors and moves can be analysed, based on events. Data in transit will have data owner signature, allowing to identify who created it, and if the data has been changed and/or moved between components.

Incoming http-based interactions with platform's API shall be authenticated and authorized. Access control to http resources can be changed only centrally by platform administrators.

For those components running outside of the cloud platform, the component owner is responsible for implementing the best patterns for protecting data before it enters the platform. Tutorials/wiki pages shall be provided to the component owners for this purpose.

3.2 Interoperability

NAIADES' design was rooted on the idea to create a robust, secure, flexible, portable, replicable, and interoperable platform. As the project includes three different pilot sites on top of different technical development and deployment, interoperability is one of the key values to provide in the NAIADES platform in order to provide harmonized smart water services for the pilots. Considering that the pilot sites have different needs and use different technologies on data collection (e.g., legacy, IP based, edge networks to be used, etc.), the focusing of the interoperability inside of the platform is to provide data/semantic interoperability from the collected data to the final services.

NAIADES' interoperability strategy is to adapt NGSI based data models and context APIs. Considering that FIWARE has already started the path towards interoperability in many sectors (smart cities, industry, etc.) based on NGSI data models and APIs and it is currently working on the definition of semantics for water sector in collaboration with European Water Projects (including NAIADES), NAIADES will make use of the well-established FIWARE based NGSI data models and APIs for data communication and interoperability.

Instead of reinventing the wheel, NAIADES will use of existing data models from the domains of "Environment [5]", "Weather [6]" and "Smart environment [7]". These domains define several data models which will be used and potentially extended by NAIADES as shown in the table below. It is noted that these data models are not definite and can evolve, and NAIADES will be examined and may extend the existing data models based on the need of NAIADES use cases. As not all data models needed by NAIADES services exists, the development of non-existent data models (e.g., water distribution, water consumption, water treatments) will be developed with collaboration of other H2020 European Water Projects.

For this mid-term phase, the consortium has focus in aligning all its generated data to the data models of:

Data model	Link to documentation
WeatherForecast	https://fiware-datamodels.readthedocs.io/en/latest/Weather/WeatherForecast/doc/spec/index.html#key-value-pairs-example
WeatherObserved	https://fiware-datamodels.readthedocs.io/en/latest/Weather/WeatherObserved/doc/spec/index.html#key-value-pairs-example
FlowerBed	https://fiware-datamodels.readthedocs.io/en/latest/ParksAndGardens/FlowerBed/doc/spec/index.html#key-value-pairs-example
WaterQualityObserved	https://fiware-datamodels.readthedocs.io/en/latest/Environment/WaterQualityObserved/doc/spec/index.html#key-value-pairs-example

The platform architecture includes modules for translating data from its wild representation collected from homogeneous sources to FIWARE compliant data models. Also, a module for validating compliance to the data models is expected to verify this property for every API request the platform receives.

3.3 Monitoring

Water utilities are currently equipped with a large number of IoT sensors and data-driven technologies along the water network, being the district water meters, pressure gauges, pressure and flow sensors in control points; and water and waste water quality (pH, turbidity) the most commonly installed. Nevertheless, the data is currently only used on a reactive way, this is, when something happens, the collected data is analyzed in order to detect the source of the problem.

Additionally, it was identified that water utilities are not the only entities requiring an appropriate management of water. Industry, Agriculture and Cities have also to manage large amounts of water, where monitoring is essential to reduce consumption, losses, waste and energy; and it is also useful to optimize the management of their infrastructures and staff.

NAIADES's objectives for monitoring are focused on five activities:

- Provide expert support about new monitoring technologies available to help with unresolved issues, such as best practices for data collection, new water quality sensors and ad-hoc solutions.
- Monitor all gathered data and combine measurement from different sectors (traffic, weather, socioeconomic, etc.) so to create new reactive alerts.
- Generate big data related to critical events produced by internal or external incidents to the water system. Current urban water models will be used to generate unusual data, not usually part of the historical data, which will be essential for training AI algorithms for critical situations.
- Provide critical water consumption monitoring, this is, building an analytical model by analyzing consumption; pressure and flow sensors time-series and relevant alarms to understand critical water consumption states. The model developed will enable assessment of risk for water system to transform into an alarm state (i.e. risk of high consumption due to drought, etc.). The water system operator will therefore be able to visualize the state of the system, detect the problematic states and design interventions based on the information provided.
- Generate the best treatment parameters (dosages, retention times, etc.) from the quality measurements at the inlet of the Water Treatment Plants so to assure the best quality at the outlet.

3.4 Data analysis/prediction

The real profit for water management is taken from forecasting analysis. These analyses can accurately predict future measurements and events thus giving the advantage to be able to plan any reaction in advance, preventing or reducing consequences from future issues and optimizing water related operations and management.

NAIADES will utilize state-of-the-art AI technologies to produce future predictions that will be useful for expert staff and also to feed the system that will suggest optimal actions (Decision support system) to support their decisions. The main predictions that NAIADES aims to provide are:

- **Weather predictions**. Due to climate change, extreme weather events are less predictable, thus it is very important that water utilities are not just reactive to weather and they are able to plan in advance the best actions to severe weather situations. Furthermore, being able to obtain more accurate weather predictions will improve forecast provided by other services such as demand or quality forecast.
- **Water demand predictions**. Water demand predictions are closely related to water utility's costs. Currently water utilities are using very inaccurate and simple mechanisms for future predictions, which may be often erroneous. For example, in Alicante, water demand estimates for the next day are (approximately) done by taking water consumption on the same day of the last year, corrected

by the factor of local consumption average changes. In case consumption exceeds prediction, water utility needs to re-fill water tanks during the day, which increases the costs of water production as electricity costs are significantly higher during the day.

- **Failures and leakages**. One of the important tasks of water utilities is to minimize water losses through their pipeline system. The leakage is a normal phenomenon in every pipeline system, however, as the leakages increase over time, water losses can become a significant cost-related problem. For illustration: in Malta, an important potable water source is water desalination, a relatively expensive process. As water losses in Malta pipeline system are comparable to desalination water production, it means, that by reducing water losses, water utilities in Malta could significantly reduce their water production costs. Identifying failure and leakages within the pipeline system would help water utilities to identify problematic parts of the pipeline system more accurately and faster, which would have important cost-related consequences.
- **Water quality predictions**. Water quality management is critical to guarantee adequate levels of water quality for their specific uses (drinking, bathing, etc.). To not control these levels constantly will imply costly treatments and infrastructure maintenance. Being able to detect beforehand the degradation of water quality allows the end-user to plan efficiently the best actions to reduce time, costs and environmental impact and to guarantee best water quality at any moment.

3.5 Decision support

As mentioned above, water utility infrastructures are largely equipped with IoT sensors and data-driven technologies. The correlation and analysis of the data generated within the infrastructure might provide a better understanding and enhance the overview of the involved processes and operations, strengthening the user's situational awareness by providing alerts for significant changes and occurrences; and support decision making.

The NAIADES solution aspires to provide the said better understanding of the involved processes and operations, contributing to each individual link of the value chain such as i) the effect of process monitoring in water resources like quality and consumption; ii) the facilitation of communication and knowledge sharing between actors and stakeholders with different and roles and responsibilities; iii) the early detection of daily details and flaws; and iv) the maintenance procedures and requirements.

The NAIADES Decision Support System (DSS) shall consist of three major modules: Model Management, Knowledge Management and User Interface (UI) Management.

The **Model Management** module is a key element in most decision-making processes. It consists of both the DSS models and the DSS model management system. DSS helps in various decision-making situations by utilising models that allow the user to analyse information in many ways. The model management system stores and maintains the DSS models. Its function of managing models is similar to that of a database management system. The models to be used in DSS depend on the decisions of the user and the kind of analysis required. Once the initial incident parameters have been established, the event analysis begins.

The **User Interface Management** module shall facilitate communication with the DSS and will be set to perform the following tasks:

- **Model management**: DSS needs models to analyse the provided information. The models create new information that decision-makers need in order to detect system failures and manage water resources. For example, in cases where sensors are malfunctioning or where there is a leakage or overuse of water in certain areas within the system, the DSS will aid in identifying these problematic areas or the causes of overuse.

- **User interface management**: A user interface (UI) enables decision-makers to access information and to specify the models they want to use to create the information they need.

The **Knowledge Management** module shall provide information about the relationship among data that is too complex for a database to represent. It shall consist of rules that can constrain possible solutions as well as alternative solutions and methods for evaluating them. In this context, an example of Alicante's Use case can be given. Combining data from Sensors, relevant AI Toolkits, the results of the Consumption & Awareness module, past or present data and short-term weather forecasts and Calendar information (National Holidays and certain dates of interest to the city, mainly) the DSS shall be able to provide significant support in the decision-making of the end-users from the Low level (operators) to the High level (municipality), guiding them on their decisions for future actions and identifying early any major discrepancies between consumption points and time periods. For example, on a Short Term Analysis Level if the Daily consumption increase by 50% compared with the last week's average, the DSS will automatically inform the user and suggest a course of action relevant to the consumption point. In the case of a School, it will recommend analysing the night flow and to look for internal leaks. In the Case of a Green area, it will recommend looking for internal leaks and afterwards to analyse the hourly consumption pattern and compare with the irrigation schedule. A detailed description of all the DSS functionalities will be provided in the second iteration of the DSS Deliverable (D7.2).

3.6 User awareness and behavioural change support

NAIADES aims at providing a set of innovative services that will enhance public awareness on water consumption and usage savings, promote user engagement and enhance user participation in water conservation activities. These include awareness services that present detailed information about water consumption and behavioural change support services that leverage the power of persuasive strategies and features such as social proof, tailoring, self-monitoring, goal setting/ commitment and rewards. The goal is to change the attitudes and practices towards water conservation by enhancing public awareness on water consumption and usage savings, promoting user engagement and enhancing user participation in water conservation activities. These services will make use of the data middleware and AI services which are part of the NAIADES solution.

More specifically the NAIADES awareness services and behavioural change support services focus on the following set of use cases:

- Water consumption awareness for water management companies and public officials. This use case will show how to make use of the NAIADES data aggregation middleware and optimally water consumption data to interested stakeholders in order to monitor and understand how water is consumed in a specific area or consumption point, while considering contextual parameters such as the time of consumption and weather conditions. Decisions on water consumption mitigation measures can rely on such information and such measures can be monitored after their implementation.
- Behavioural change support for inducing sustainable water use behaviours among water consumers. This use case will show how ICT solutions can be used in order to change consumers' perceptions and actions towards water conservation. In NAIADES, we focus on school students, a group of consumers that can provide a channel for generating great impact as students i) will evolve to the responsible citizens of tomorrow and ii) can transfer the knowledge, attitudes and behaviour they shape to their families, leading to a cascading effect of the NAIADES impact.
- Water consumption awareness for public employees. This use case will show how to make use of watering data and AI services for watering predictions in order to inform public employees decisions related to plants watering tasks. Plants watering is performed in nearly every municipality and public building, and commonly leads to unnecessary water consumption and waste of human

resources. The NAIADES approach aims to inform watering decisions for timely watering actions and water savings.

4 NAIADES architecture

In this section, NAIADES' architecture design is introduced from different perspectives/overviews that facilitate the presentation of the guidelines related to business interconnections, system interoperability, big data management, communication, and security.

4.1 Business Overview

The general concept of the NAIADES architecture is based on some basic principles that interlink roles between the partners, based on what they will do during the project. Interconnections between data providers, data owners, service provider, data consumers and marketplace users are presented in this paragraph at a high level. These interconnections create a process flow of data/information/services that lead up to the final NAIADES solution.

The first level of this process flow is the data owners, meaning the partners that own the raw data that are produced during the implementation of the system. Since the system will rely on physical systems where sensors are already (and more will be) installed; on the locations of cities where the water companies operate, these data will belong to the end-users themselves, i.e. the water operating companies. However, this project is heavily dependent on AI services, therefore large amount of data is required for the development of these services, creating the demand to retrieve data from other sources. These online sources are the original owners of the data that will be used for the system.

Second level is the data providers, i.e. the partners that are responsible for the provision of data within the system. As such, IBATEC will provide physical sensors required for Carouge and Alicante use case solutions that are not existent in those end users; CUP will acquire the required sensors for their pilot (Braila); and AIMEN will develop the WTP scaled lab pilot with the required sensors. For the next step, ADSYS, UDGA, SIMAVI and AIMEN will collect and aggregate all the data from the end users, Alicante, Carouge, Braila and WTP scaled lab, and other sources (weather providers) and they will send the data to NAIADES platform following common data models and the security mechanisms. DISY is in charge of providing a tool for applying the appropriate FIWARE-NAIADES data models to the data from all sources so they follow the same format. Since the system's database will also exchange (provide and receive) data with all components, UDGA can also be considered a data provided since they are responsible for the development of the core of the platform (context broker), as well as SIMAVI which is responsible for the Cloud platform and its communications.

Third level is considered the data consumers, in which the partners directly use the data that is stored in the database, without any processing (raw data) or from a service. In this category are included KT, responsible for the User Interface (UI) of the system and all visualizations, as well as ICCS, responsible for the development of the dashboards that are related to their consumers' recommendation systems.

Fourth level is the service providers, the largest of all levels. Here, the partners are developing services which will add value to the overall system by offering integration of databases (UDGA), integration of cloud services and the development of the marketplace (SIMAVI), the overall security mechanisms that will run throughout the system (UDGA and GT), the data fusion/refinement (ADSYS), the development of urban models (IHE), to the recommender systems (ICCS). Regarding the AI services, KT and JSI will develop the water demand predictions, CERTH the weather prediction module, JSI the water consumption analysis, failure and leakage prediction and the consumer confidence solution, and finally AIMEN will develop the water quality monitoring, and the dynamic treatment and quality prediction module.

Fifth level regards the service consumers, partners which use the outcomes of the services either directly or indirectly. The end users can be considered as partners who use the outcomes of the services via the DSS therefore directly from a service, and as such are considered service consumers. Specifically, for the visualizations of the outcomes of the predictions from the AI services, KT and ICCS are again considered as service consumers since they will be displaying the predictions via the UI and the recommender dashboard. Additionally, some service providers can be also considered service consumers since they will use the outcomes from other services, like weather predictions as input for their services.

Finally, on the sixth level, as marketplace users are considered the end users that will be using the system and will be able to access any kind of service they desire. However, target of the NAIADES system is to attract external water utility companies that will be interested in the functionalities of the system and will approach the consortium as potential clients.

A visual representation of the data flow, the levels and the main partners that are involved in each one, is shown in Figure 6.

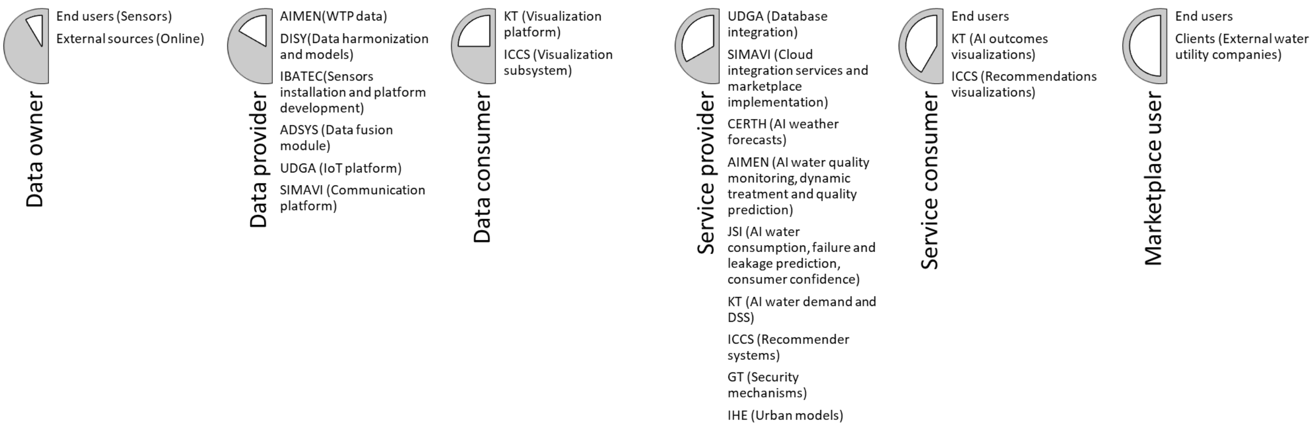

Figure 6. Business Overview: Interconnections of roles for each partner within the project.

4.2 Layers Overview

The layered structure from the Open System Interconnection model (OSI model [8]), which was developed to simplify network complexity by describing what the layers must do, was taken into consideration, as well as the n-tier architecture [9] [10], which has the processing, data management and presentation functions separated so it is easier to create flexible and reusable applications. Using both approaches as a guide, NAIADES platform design resulted in an eight layers structure (Figure 7). Six layers are linked to the stages the data must go through from their raw state on source to the refined version presented to the users. These stages have different purposes, such as transformation, processing, and management, and they are related to different technologies. The other two are transversal layers (management and security) that will affect the data during the whole journey in order to guarantee its consistency, privacy and assure it will circulate securely in the platform.

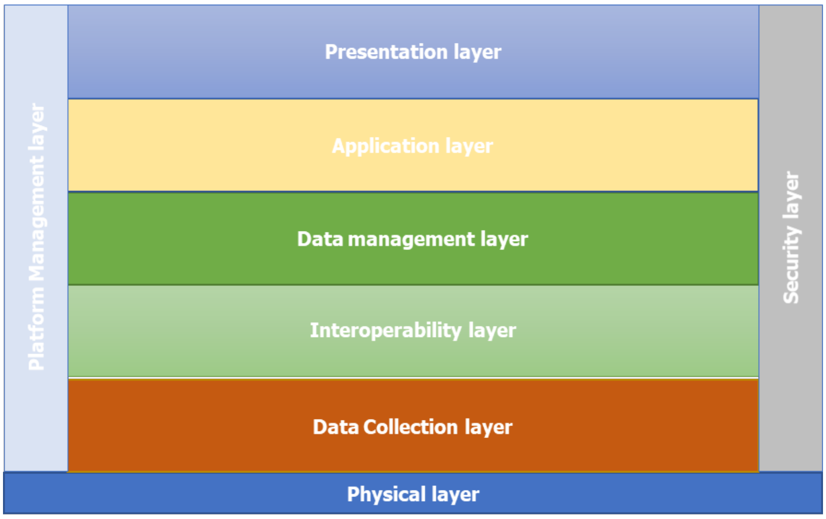

Figure 7. NAIADES architecture layers overview.

- **Physical layer**. This layer represents all the elements that are going to be measured, monitored and that will provide any relevant information about the processes to be managed. The main element is the water; its properties (quality and quantity) can be measured in deposits, pipes, reservoirs, rivers, etc. The other elements to be measured will depend on the management process, from water infrastructure components to other sectors such as weather, traffic, or socioeconomic info.
- **Data collection layer**. This layer can be divided in two sublayers: sensors/devices and gateways/aggregators. Sensors/devices sublayer represents all the HW devices in charge of telemetry, including other platforms providing information, such as weather stations and smart cities platforms. Gateway/aggregators sublayer includes the HW devices that connect the different devices and their data. Furthermore, this sublayer is in charge of the pre-processing of the data; preparing it to be sent to the main platform. It correlates data, making data from different devices, which likely implies different type of measurements, has a common denominator. Also, it homogenizes data and makes it compatible with other platforms by using functionalities from the interoperability layer and it will authenticate the data before sending it to the platform by using security layer components for data integrity.
- **Interoperability layer**. This layer is in charge of the translation of data collected from water utilities and services with the objective of creating common data models. The incoming data from different encodings and from different sources is translated to a common vocabulary for the creation of a single data dictionary. The functionalities of this layer are divided in two main actions: translation and validation. The translation transforms large volumes of data (high-frequency, high-volume) to common vocabularies and enriches them with metadata. The validation avoids any data with the incorrect data model format to reach the data management layer. This action is essential to guarantee that all the data inserted in the platform is correct and consistent.
  Currently, all the developments on this layer are taking into account upcoming standards while collaborating in current initiatives with cutting-edge semantic and big-data frameworks (like FIWARE).
- **Data Management layer**. This layer is directly linked to FIWARE data/context management chapter. The core of FIWARE, the context broker, will be the main component of this layer. The purpose of data management is to ease the development of applications that exploits data by being in charge of processing large amounts of data in an aggregated way, manage changes in context information and store all the context information related to defined context entities in a repository. Due to the use of FIWARE on this layer, all the connections to receive and share context data with

other layers are done through NGSI or NGSI-ld, to receive and generate notifications of context events.

- **Application layer**. This layer contains all the functional business logic. It is represented by a set of individual components and; applications/services; that work as a black box for the users, able to represent a business activity. In NAIADES, it includes all the technologies that analyse and process data, from statistical methods, time series analysis to machine learning techniques, so to create new information. Two main categories of services can be identified in NAIADES: internal and external applications. Internal services are components intrinsic to NAIADES; that will generate new data that will be stored in the platform for other services to generate more data or the NAIADES presentation layer to display them conveniently. External services are components that can be added to NAIADES, having access to agreed data, so to produce additional ad hoc outcomes. Those external services should provide their own presentation technology.
- **Presentation layer**. This layer focuses on the Human Machine Interfaces (HMI). It includes all the frontend applications used to present data to the users in the most efficient way, this is, data representations (graph, charts, tables, images) that solve the users' requirements. This layer translates data from the application layer to a format easy to understand by users. The most common ways to provide the information is by using web technologies or by developing dedicated applications that can be accessed from different devices: PCs, phones, and tablets.
- **Security layer**. This is a transversal layer affecting all the other layers of the architecture. Its main purpose is focused on data, users and system protection. It ensures secure exchange of information, guarantying end-to-end confidentiality and integrity by implementing authentication, access control and transport security. Well established authentication mechanisms will be combined with a multi-stakeholder attribute based access control mechanism that will provide, based on a security token included within the submitted request and the evaluation of the security policies, fine-grained access control to data in the Cloud guaranteeing that only the specify group of receivers indicated by the data owner has access to the data and for a concrete purpose. Furthermore, blockchain will be used to provide an audit trail of water consumers' data, enabling both product data traceability and secure access for stakeholders.
- **Platform Management layer**. In charge of the platform health, it uses logs and monitors the platform infrastructure to detect any problem that will hinder its correct operation. It will also use functionalities from the security layer to add security info to the management of the platform and be able to trigger alarms when any type of problem arises.

4.3 Communication Overview

This overview presents the different communications protocols and guidelines utilized amongst the components.

4.3.1 Communication between SCADA systems and Data Collection Aggregation

File Transfer Protocol (FTP) is used to establish the communication between pilot SCADA system and Data Collection Aggregation module. FTP is a communication protocol that is used to transfer files (measurements data) over a TCP-based network, such as the internet. The SCADA system connects to the FTP server of Data Collection Aggregation using the server address and it authenticates using a designed username and password. A backend process layer of Data Collection Aggregation will act as an FTP client and it will connect to the FTP server via FTP protocol, extracting needed files based on address, user and password.

4.3.2 Data Collection Aggregation pilots' modules

Data Collection Aggregation modules are external applications (per pilot) that will run in separate server machines, external from NAIADES platform.

Data Collection Aggregation can run on any server, host and port is needed, as long as it can send HTTP

REST request to the NAIADES cloud platform. It will have to set up in the properties file the URL of the NAIADES cloud platform and the credentials for authentication.

Data Collection Aggregation will communicate with its own FTP server via FTP protocol to get raw measurements data. Data Collection Aggregation will integrate common Data Model tools (from Interoperability layer), integration consisting on adding a structure of jar (Java Archive) files in first phase of the integration. Data Model tool will use IO API writing locally the modeled data. Data Collection Aggregation will get the input of the modeled data and will send it to the cloud as:

1. Call NAIADES Identity Manager using HTTP protocol via web service REST request to get authorization token. This authentication process will be done at the beginning when the server is started. After that, Data Collection Aggregation will use the token – refresh token, to sign each request sent after the authentication.
2. Send the modeled data to the Cloud using HTTP protocol via web service POST REST request as payload data and authorization as request header. As a safe mechanism, in situation in which sending measurements to the cloud generates 401 Unauthorized Error, automatically Data Collection Aggregation can do login again, get a new token and resend the data with the new token.

In the case some custom use cases needed NAIADES application to call Data Collection Aggregation, the communication will be done using the same Http protocol and web services exposed by Data Collection Aggregation.

All communication between pilots and NAIADES cloud platform are synchronous. A mechanism of Queue Messaging could be implemented, to add the message from the pilot in a queue, and NAIADES application to consume the message from the queue, like NATS, or ActiveMQ, RabbitMQ, etc., but since all communication between pilots and NAIADES are synchronous, http web services mechanism is more suitable for the present scenario.

4.3.3 Communication between internal users HMI and cloud platform IoT

The HMI frontend and backend microservices will be hosted inside the cloud platform.

HMI module is a web-based frontend application of the NAIADES ecosystem, dedicated to the pilots' users of Carouge, Braila and Alicante. Pilots' users will view their pilot measurements, notifications and statistics in custom windows depending on their use cases. The data that users will see in the interface will come from internal cloud repository and processing services, as a result of processed measurements sent from their Data Collection Aggregation.

NGINX proxy will be configured to expose outside the platform HMI based on the URL. If the user types the main URL of the platform e. g. "NAIADES-application-eu:" or "e.g. NAIADES-application-eu/login" NGINX will redirect the request to the internal application that loads the HMI frontend.

The requests from frontend HMI UI to IoT platform will be authorized by the Access Control authorization security mechanism.

The credentials for internal users will exist in the Identity Management module and will be provided for each pilot. HMI modules cannot access directly the Identity Manager's Database, it must be done using Identity Manager API, for login/logout methods.

As a first step, at login time, row credentials will be sent from HMI frontend to the Identity Management, for an authorization token, using REST HTTP requests. The token will be saved locally in browser and will authorize each request that HMI will be sent to the NAIADES services. Authorization layer will authorize or reject the requests based on user roles saved in Identity Manager. If users send requests with bad token the NAIADES application will respond with 401 Unauthorized HTTP Error.

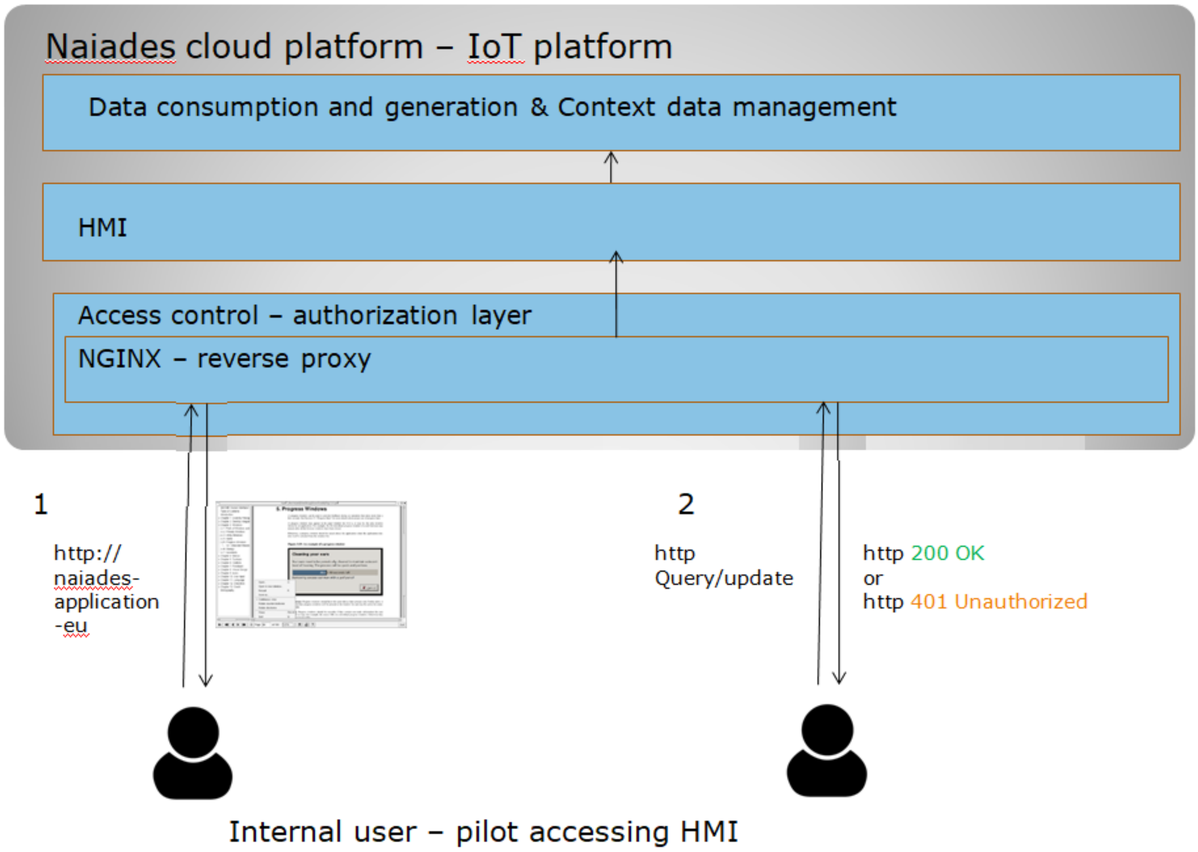

Figure 8. NAIADES communication - Access control

4.3.4 Communication between external users and cloud platform – Marketplace

The Marketplace frontend and backend microservices will be hosted inside the cloud platform.

NGINX proxy will be configured to expose the platform HMI based on the URL. If the user types the main URL of the platform, e. g. "NAIADES-application-eu/marketplace", NGINX will redirect the request to the internal application that loads the Marketplace UI.

The requests from the frontend Marketplace UI to the Marketplace backend (inside platform) will be authorized by the Access Control authorization layer.

The request from the external users or external applications that wants to integrate NAIADES application, based on marketplace API described in the interface, will be done using a communication between external-users/applications – marketplace backend using HTTP REST web service protocol. All requests will be authorized by Access Control authorization layer and will be handled using the NGINX proxy.

All requests from Marketplace backend to IoT platform process layer will be done internally and they are also authorized.

Marketplace frontend allows external users to create their own account. External users' accounts will be stored in the Identity Management module. The communication between Marketplace frontend and Identity Management will be done the same as the communication between HMI and Identity Management, using HTTP protocol REST requests. Marketplace Frontend modules cannot directly access Identity Manager Database, it will be done using the Identity Manager API, for login, logout and edit the user.

External users can access dedicated NAIADES services via Marketplace backend, they do not have direct access to the main processing services, neither when the user is authorized. Custom web services will be exposed to the Marketplace API in order to let others' external application to use it.

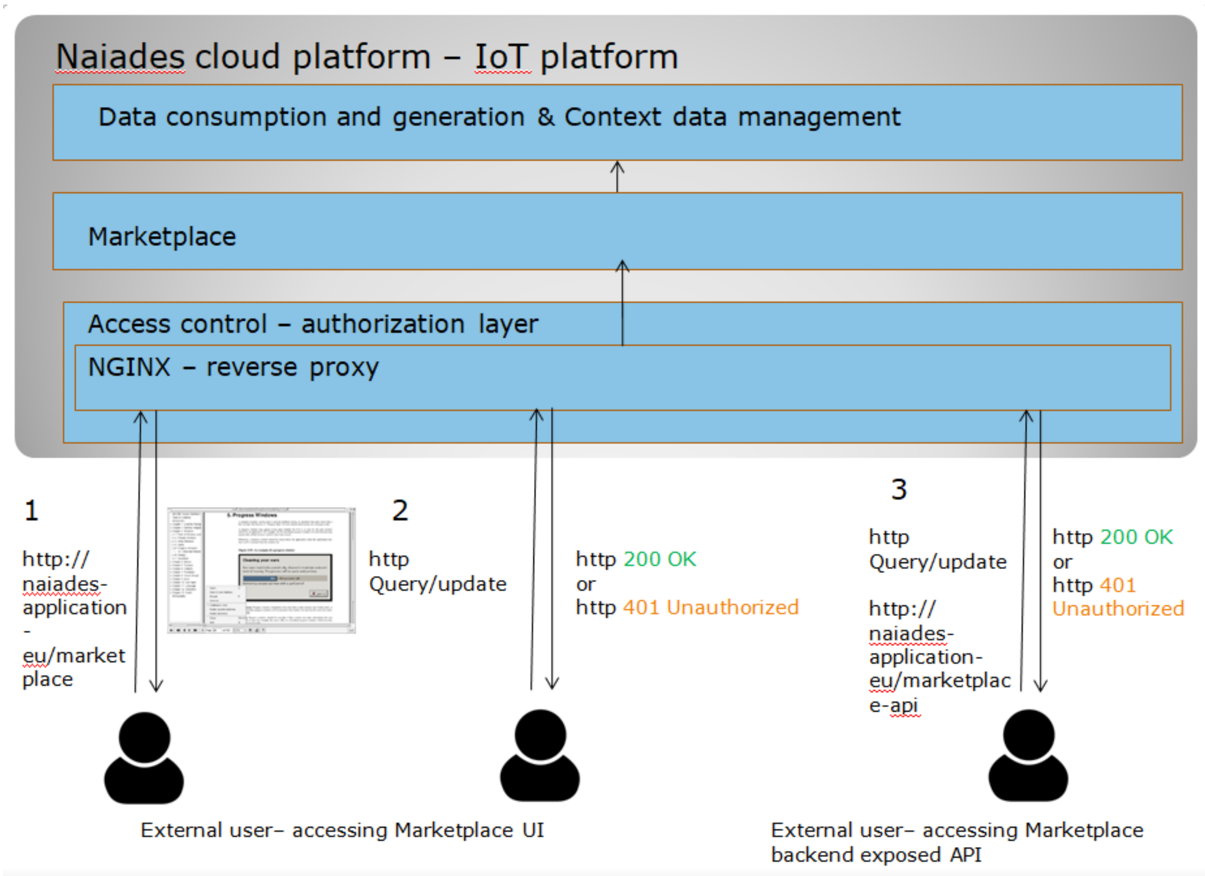

Figure 9. NAIADES communications - External App/users' access.

4.3.5 Communication between between the internal services and the data manager

Data management layer and most of the application layer will be deployed in the same platform, server instance machine. All microservices will be in individual docker containers that will run on the same host but using the docker instance name as identifier, and on unique and different ports. The ports are private and open internally, they cannot be visible and accessible from outside the platform. The communication between Data Management and the rest of the services will be done via HTTP RESTS requests using ports exposed by docker container. Also communications between a DataBase repository and a backend API is done via docker link file configuration and by using JDBC driver and connection credentials configured inside the backend. The other services will communicate with Data Management using an internal security layer Access Control that will let the services to make some processing actions, depending on their role. That Access Control will be validated before starting the process. The Services/User's roles are stored in the Identity Management.

4.4 Information Overview

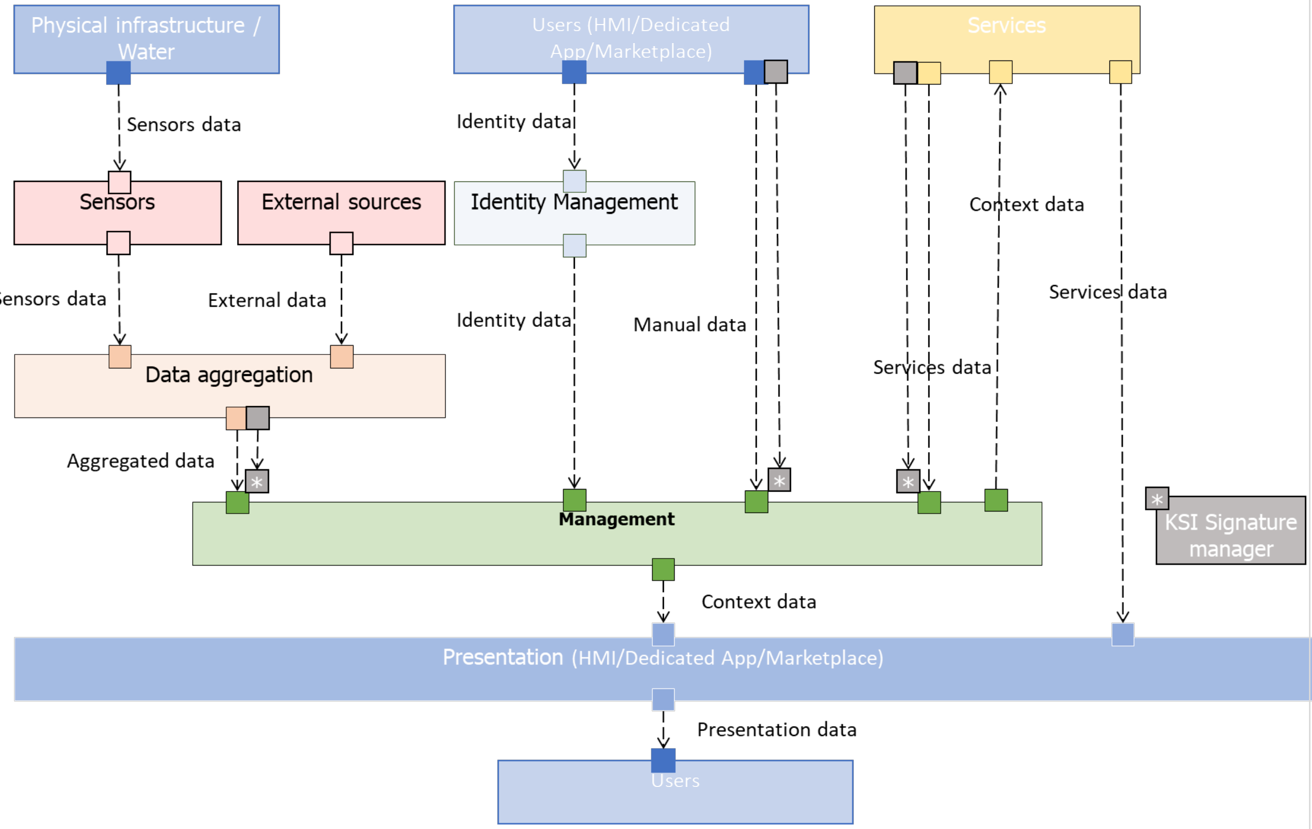

Figure 10. Information Overview.

Figure 10 illustrates, from a high-level point of view, the exchanges of information between NAIADES components/layers. NAIADES main input is data from the physical infrastructures collected mainly via sensors and legacy systems that is sent to NAIADES platform. All the collected data from the field is converted to common data models, FIWARE based data models, and then is shared to the component that will distribute it to the rest of the components, the context manager. Additionally, users will be able to send data to the platform occasionally through the HMI/dedicated APPs, although their user account should be granted writing privileges by the identity manager in order to be able to do it.

The other source of information is generated by NAIADES services. They will collect data from the data manager to analyze and process it so to generate predictions, alarms, suggestions, etc.

The last step for the information is always through the presentation layer where the HMI, dedicated applications or the marketplace frontend will share information in the required visualization formats: tables, graphs, charts, text, etc.

Additionally, security data related to KSI signatures and blockchain for data integrity is shared between all data generators and the KSI Signature manager.

A brief explanation of each data is described below:

- **Sensors data**: Data measured from the physical world through sensors (water quality, pressure, flow), legacy data and infrastructure parameters (equipment parameters).
- **External data**: Data obtained from external sources, such as weather station APIs, other smart cities platforms APIs, etc.
- **Manual data**: Data inserted in the platform by the users (utilities' staff) through the provided user interfaces. It will be mainly static parameters about the infrastructures; or lab measurements when sensors are not available. Continuous annotations data will be avoided.
- **Identity data**: Data related to NAIADES' user access profiles. It includes the information for

registration and login; and the tokens for access control.

- **Aggregated data**: Data from different sources preprocessed to be shared with NAIADES platform. It requires being adapted, for interoperability reasons, to the common data models format to be used in the platform.
- **Context data**: Data shared by the data manager. It includes mainly all the data generated in the platform, formatted according to the common data models, before and after processing.
- **Services data**: Data generated by the services such as predictions and suggestions. Three different services data types can be distinguished:
    - **AI services data**: Data generated by AI services. Generally, it is new data predicted or forecasted from data coming from the aggregators. This data is sent back to the platform to be used by other services or accessed by interface tools.
    - **Dedicated services data**: Data generated by services whose use is exclusive of that service, thus is presented in dedicated interface tools.
    - **DSS data**: Data generated by the Decision Support System. It can be new values data that can be sent to the data manager or specific results to be display through the HMI.
- **Security data**: Data used for data signature purposes. It is shared between the data generators and the KSI manager; component in charge of data integrity through blockchain technology.
- **Presentation data**: Data presented to the user in the form of tables, charts, graphs, text alerts, etc.

4.5 Components/Modules Overview

NAIADES platform was designed as a modular scalable architecture (Figure 11). The modular overview presents the architecture as a series of boxes and sub-boxes that provide specific functionalities on their own and by connecting to other modules of the architecture. All the boxes could be replaced by other modules to improve the platform performance and new modules can easily being added to increase the functionalities, such as adding more services or sensors and aggregators. The new blocks should adapt to the platform formats, communications and security; for that reason, there are some blocks that can/must be integrated by any future component, such as Common Data Models or Data Signature modules, facilitating compatibility with the platform.

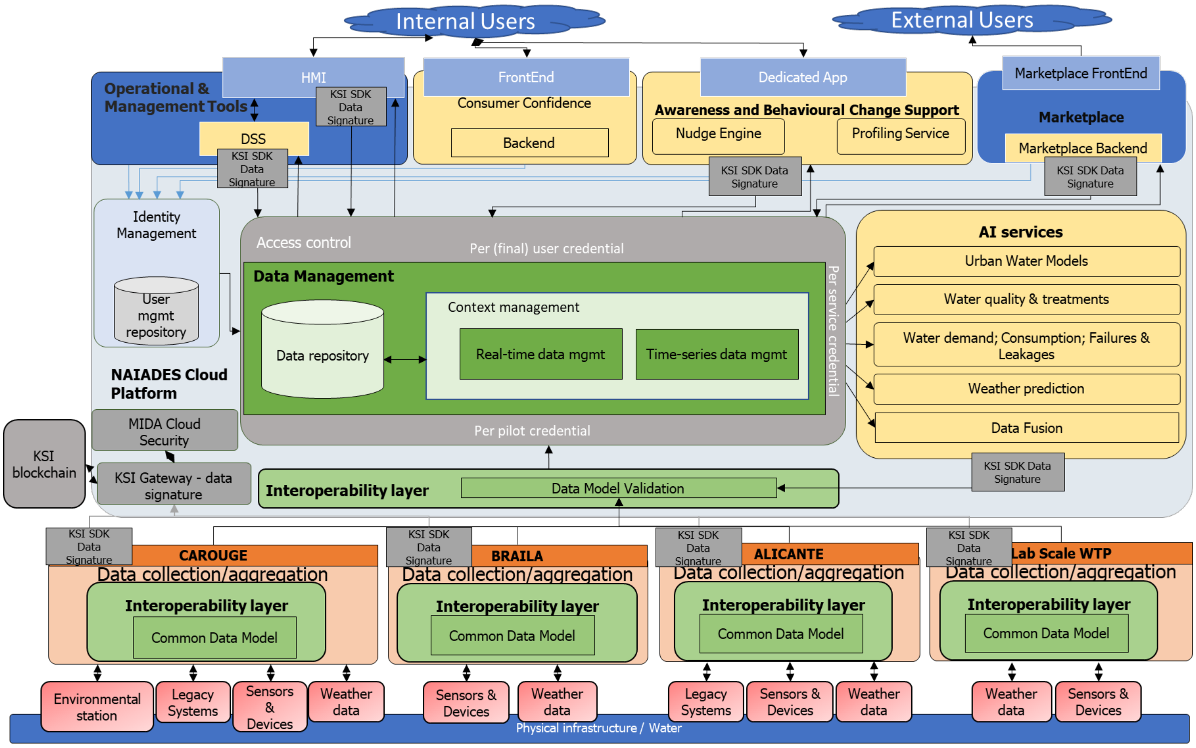

Figure 11. NAIADES architecture modules and layers overview.

4.5.1 Data Collection/Aggregation (DCA)

NAIADES presents four implementations of this module, each of them adapting to the requirements of the pilot where they are integrated: Alicante, Braila and Carouge. The fourth DCA is adapted to the laboratory water treatment plant solution. On the other hand, although their operations can differ, the Common Data Model and the Data Signature modules must be integrated and NAIADES communication protocols must be used to be able to connect to the platform by all Data Collection/Aggregation components.

ALICANTE

Description

Data collection/aggregation is a software component which periodically gets data from Alicante pilot heterogeneous data sources (SCADA system, DDBB, sensor networks...), adapts it to an adequate data format and forwards it to through a dedicated API to NAIADES platform. (will support integrated version of Data signature (0) and Common Data Models (4.5.3)).

Functional Requirements

- Provide a time series of parameters at a given frequency as an available output for the cloud platform (Data model validation module).

Non-Functional Requirements

- Performance: the module must be able to collect all data from at least 1000 parameters every minute with delays <10 seconds.
- Scalability: the module will support data collection from at least 1000 parameters.
- Portability and compatibility. It will run on external premises. Compatible with any Linux distribution.
- Reliability, availability, maintainability: 24/7.
- Security: Data retrieved or gathered at secure FTP server; tokenized and encrypted connections.
- Usability: not critical, end user will not access this module.

Diagram

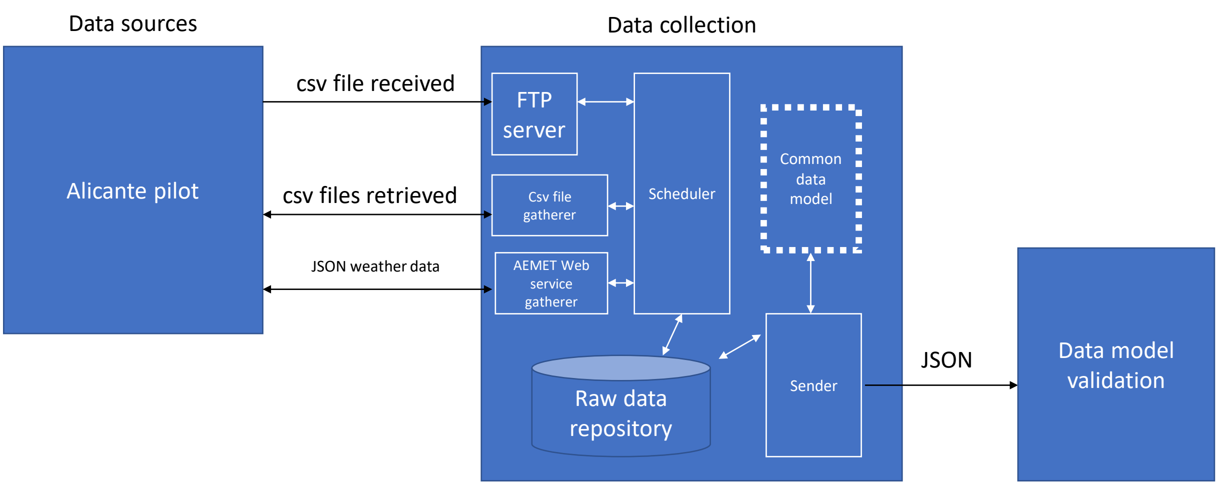

Figure 12. Alicante Data Collection/Aggregation Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

- SFTP.
- SQL.

BRAILA

Description

Data collection/aggregation for Braila is an application module that get sensors data from Braila pilot, periodically, and forwards it to NAIADES platform, to be processed and store.

DCA Braila is a module that will run in a separate environment in NAIADES server. The module is composed by three components:

1- FTP pilot repository – it is an FTP server storage repository in which the Braila Pilot systems will store files at a settled period of time, which contain certain measurements of water. The pilot can store files in the FTP server based on a private connection using access credentials.

2- Backend java application that has the next roles:
    - get pilot measurements by connecting to FTP server, deserialising the files stored in FTP server, get its content, and send the content to NAIADES platform, at a specific given time, e.g.: one-hour frequency;
    - get weather data for Braila pilot from weather API, and send it to NAIADES platform with a frequency of one hour;
    - both water measurements and weather data from Braila have to be modelled by NAIADES common Data Models, thus, the backend calls Data Model tool with raw data to get back the modelled data, and sends the result to NAIADES platform to be validated and processed;
    - the backend is also responsible of pilot authorization, it requests the authorization from Identity Management via OAuth, sending pilot credentials to get back the authorization header. The backend uses the token and send the request header together with the modelled data to NAIADES platform.

3- Common Data Model tool is a module that will convert raw data from the pilot, in modelled data, compatible with Data Validation from NAIADES platform. This module will be integrated locally in the Data collection aggregation (Module 4.5.3).

Functional Requirements

- Provided a time series of a list of parameters at a specific time, the service will provide as a Web Service, just output in a json format, no input parameters are required.

Non-Functional Requirements

- <u>Performance</u>: - medium performance
- <u>Scalability</u>: - medium to low scalability, because the data has a low volume of data, but it is sent every hour. The only volume which will increase continuously will be the FTP repository volume, but after the data will be sent to the cloud NAIADES platform, a procedure will delete old files.
- <u>Portability and compatibility</u>: Data collection aggregation will be a module that will run in a separate server. The backend is portable, because it is dockerized, but the FPT server is not portable, it needs to be installed and configured. Compatibility exists between Data collection aggregation backend and Data Model tool via common java language calling the required method from a given jar.
- <u>Reliability, availability, maintainability</u>: The module will be up and running all time. In case of updates, each component inside (backend, Data Model, FTP server) can be stopped and fixed separately.

- Security: - FTP is private and can be accessed by credentials, Data collection aggregation backend authorize pilot user using NAIADES Identity Management; and sign and sent data with the authorize token.
- Usability: Data collection aggregation has dynamic components: backend, FTP and Data Model tool. Each of them can be managed, and configured independently, it can be quickly updated and reconfigured, if Data Model tool needs another input format backend can easily modify the format needed. Data collection aggregation backend and Data Model tool are written in Java, which is maintained and documented. Basically, Data collection aggregation is not a single standalone application, it contains many modules.

Diagram

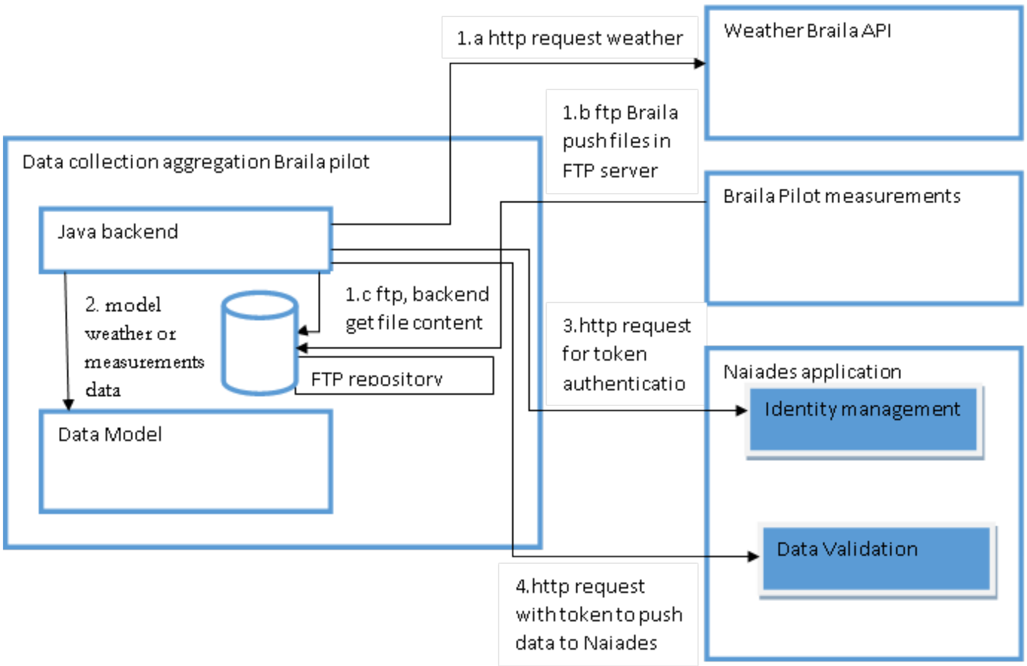

Figure 13. Braila Data Collection/Aggregation Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.
- By WebService using HTTP protocol, with json content, between Data collection/aggregation and Naiades Interoperability layer.
- FTP protocol between Braila pilot and Data collection/aggregation.
- Java.

CAROUGE

Description

The main objective of this component is to collect, aggregate and format sensed data and open data. It also provides data credential to fit into the security mechanism of NAIADES platform. IoT sensors will be connected via LoRA networks and transmitted to the component. It will be a single entry point for Carouge pilot to connect to NAIADES platform.

Functional Requirements

- Collects data from sensors.
- Aggregates information into domain specific structures.

- Provides credential.
- Posts data to IoT Platform.

Non-Functional Requirements

- Interoperability: support more than 50 IoT networks and data communication protocols.
- Availability: automatic data retrieval.
- Security: data retrieved or gathered at secure HTTP server with dedicated tokens.
- Platform compatibility: support FIWARE compatible platform as well as other major IoT platforms.
- Usability: not critical, end user will not access this module.

Diagram

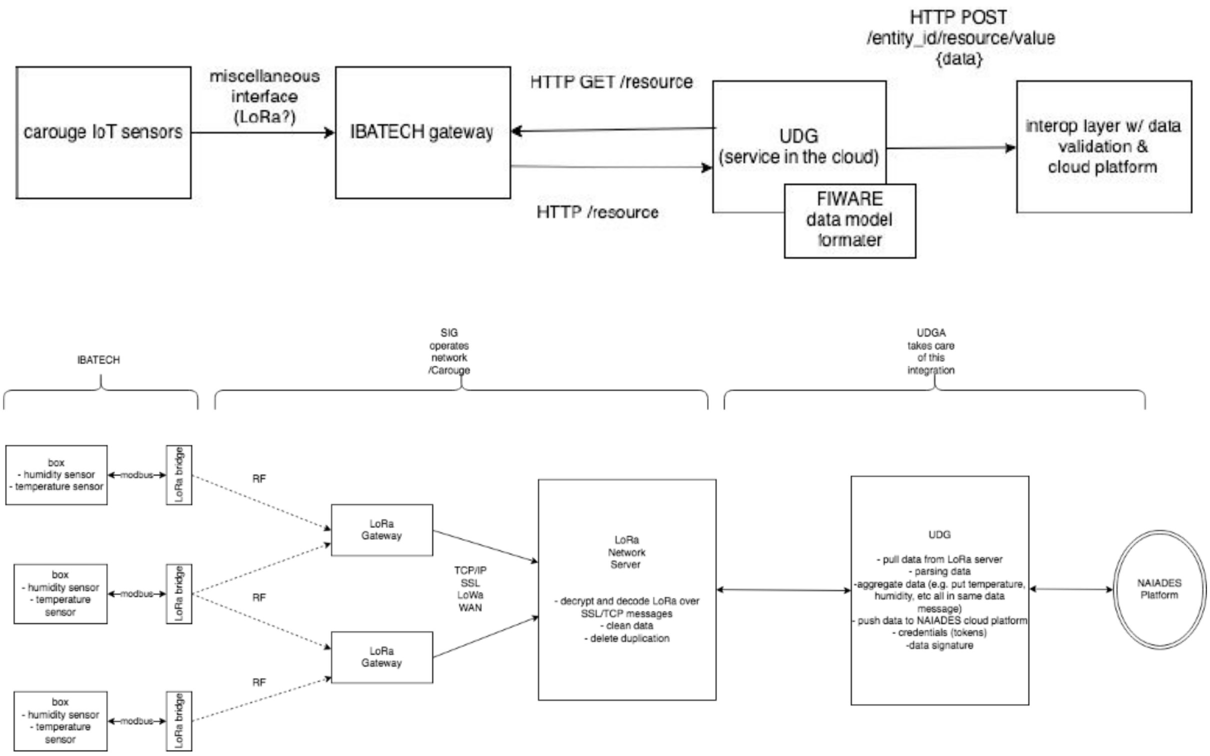

Figure 14. Carouge data collection and aggregation Diagram

Supported APIs

- NGSIv2/NGSI-LD RESTful API
- HTTPS
- MQTT, etc.

4.5.2 Environmental Monitoring

Description

Environmental monitoring is essential for achieving a high level of water services for both residential and commercial consumers, exploiting the efficient use of physical and digital components of water ecosystem. Environmental monitoring station is a key part of a complete system which monitors and examines the consequences of the environmental status in the efficiency, integrity, quality of service and demand in the Water Ecosystem operations. Through localised weather monitoring it helps to increase the water consumption efficiency, plants watering and Evapotranspiration analysis. The environmental station is composed of the following Hardware components:

1. Pyranometer: Solar radiation value.
2. Anemometer: Direction and Speed Wind values.
3. RHT sensor: Humidity, Temperature and Atmospheric pressure.
4. Rain Gauge: Rainfall intensity.
5. Datalogger with Modbus output signal.
6. LoRa WAN bridge.

Functional Requirements

- Environmental monitoring station collects data from weather sensors and connects to LoRa WAN network.

Non-Functional Requirements

- Performance: Monitoring station to work continuously 24/7 integrated 1Wp solar panel and NiMH battery. Includes 3Gmodem and LoRa WAN bridge.
- Scalability: Monitoring station does not provide scalability itself. The scalability is focused towards adding more Environmental Monitoring stations at different location, which could be interesting in the future.
- Portability and compatibility: It has been designed to be easily portable enabling its installation on post/pillar as well as on-walls. It provides LoRa WAN and 3G remote communication compatibility. It also provides Modbus RTU communication locally.
- Reliability, availability, maintainability: Environmental monitoring station provides 3xAA NiMH batteries, which are automatically charged by the 1Wp solar panel to work continuously 24/7. No maintenance is required. If it needs some reparation IBATECH will be responsible for its reparation along the NAIADES project duration.
- Security: It must be installed on places or locations to avoid urban vandalism (e.g. placed at 3m height). Regarding communication security, it uses LoRa WAN frequencies 863-870 kHz with AES crypto-algorithms point to point OF 128 bits. Two modes of operation: CMAC for integrity protection and CTR for message encryption.
- Usability: It is completely autonomous once it is installed and configured.

Diagram

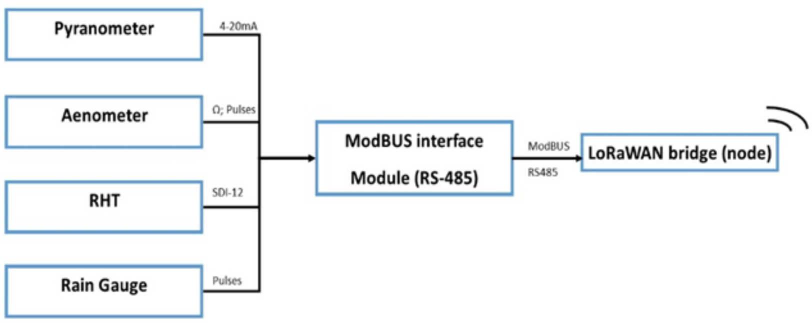

Figure 15. Environmental Monitoring Station diagram.

Supported APIs

- Modbus Interfaces.
- LoRAWAN gateways.

4.5.3 Common Data Models

Description

Existing data sources and data producing components are often in highly heterogenous, non-harmonized data formats. This leads to the need of designing highly expensive data transformation processes for every component. In particular, components from big companies use their own data format only for their own ecosystem, thus, to prevent the lock-in effect and to enable an effortless integration of different components and data sources, an automatic transformation to a common data model is needed and highly beneficial.

This component automatically transforms the input data to the whole system to a standardized and harmonized data format which prevents the need that each other component has to include such a data transformation submodule. This prevents the vendor lock-in and provides the capability for an interoperable IoT solution.

The component will be based on the TALEND ETL software and will be an instantiation of a data transformation process pipeline. It will then be provided as a jar file or as a REST API, so that it can process and transform the data directly by a method call or a simple API call. The common data model will be based on the linked data approach and use JSON-LD as the overarching data type. It will use suitable NGSIv2 or NGSI-LD standard from the EU-FIWARE project(s) as specific models.

Functional Requirements

- Provided a clearly defined existing data input with the corresponding meta data, the component will re-map and transform the data to a suitable FIWARTE NGSI-LD model.

Non-Functional Requirements

- Performance: The transformation will be triggered by other components and return the data in less than 5s, based on the network speed.
- Scalability: Can be implement and deployed at any server or locally, so highly scalable.
- Portability and compatibility: Integrated in NAIADES cloud platform and compatible with all OS which can use JAVA JRE.
- Reliability, availability, maintainability: It is maintained by DISY and each local aggregator during the project 24/7, new input data will be included as it arrives. As the Process pipeline is created by a GUI, the maintainability is easy and can be extended without much effort.
- Security: NAIADES platform cloud securities and NAIADES digital signature.
- Usability: End user will not access this module. The local aggregators can extend the functionality by using TALEND.

Diagram

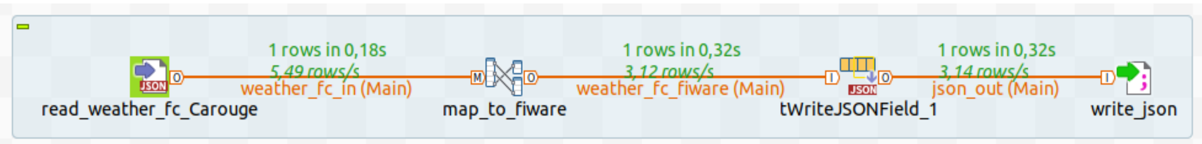

Figure 16. Common Data Models Module Diagram.

Supported APIs

- NGSIv2 and NGSI-LD RESTful API
- Java API

4.5.4 Cloud Platform

Description

The cloud platform is needed to deploy and run applications in the same and public accessible environment. It must be public hosted and accessible for each external NAIADES clients' application, such as Data Collection/Aggregation on pilots and any other external public application that can be integrated with NAIADES services.

NAIADES Cloud Platform is the processing engine of all NAIADES architecture. All data collected in pilots or external applications are processed within the cloud platform.

The client applications that need to connect to the platform, need to login to the platform using username and password provided by the administrators of the cloud platform (who is in charge also of the Identity Management, Module 4.5.8). To access the cloud platform's services or databases, private authentication is required by using an access token based on user, password and a role. The client will sign their requests using the generated token.

The cloud platform is an Amazon cloud hosting service. It is a host physical public machine that will support all internal NAIADES modules; and that will permit access to external applications through http protocols. The cloud will have an elastic IP, this is, a stable IP address, to avoid auto generated IPs at restart time.

The Cloud platform will not provide dedicated services because all NAIADES internal microservices use their own manually installed and configured tools to establish the communication and processing layers between components. All NAIADES Cloud modules will use internal ports that are not exposed publicly outside the platform. A NGINX tool will be used with the elastic IP of an instance that will handle all requests from outside the cloud and redirect them to the internal components based on requested URL names. The Cloud Platform will expose a single external port for NGINX tools to be accessible for all platform clients.

Functional Requirements

- Contain all NAIADES processing layers (data management and services).

Non-Functional Requirements

- <u>Performance</u>: Continuous and moderate concurrent access.
- <u>Scalability</u>: System does not have limitation for concurrent access, services, and tools for improving performance, can be installed using command line or by containerization. Physical machine space can be increased in case of a larger volume of data than expected.
- <u>Portability and compatibility</u>: Compatibility with docker and docker compose containers.
- <u>Reliability, availability, maintainability</u>: By administrators using credentials.
- <u>Security</u>: By using internal security authorization mechanism, by using KSI data signatures, and will be by securing the http protocol with security certificates.
- <u>Usability</u>: It can be accessed by http request.

Diagram

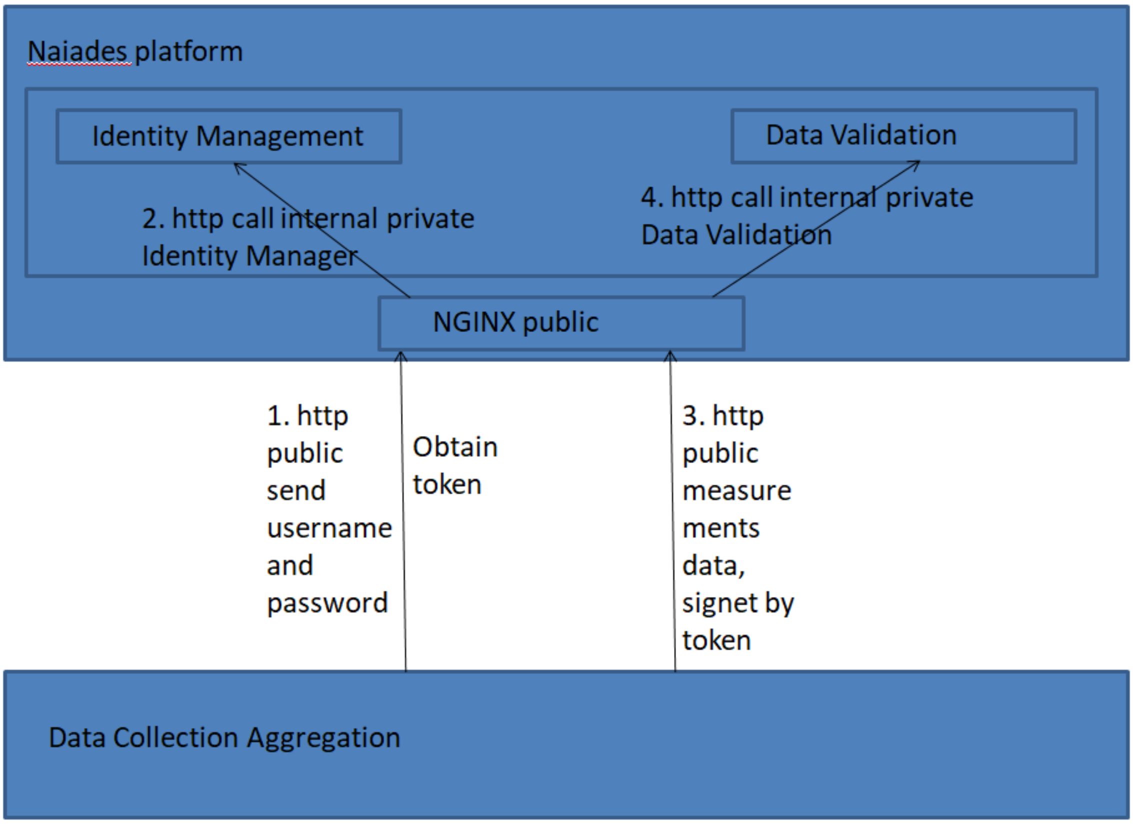

Figure 17. NAIADES Cloud Platform Diagram

Supported APIs

- NGSIv2 and/or NGSI-LD RESTful API

4.5.5 Data Models Validation

Description

The purpose of this model is checking all incoming data regarding the validity of their meta data and data model. As the existing ORION context broker component (4.5.5) does not check this before sending it to other components, this central check is essential, so that every other component can expect the same input data formats and corresponding meta data. As errors in the data writing step can always happen or errors in the transmission, this check is always needed even after the common data model component 4.5.3. These errors otherwise will propagate in the overall system.

As this component is still in early development, many core decisions have still to be taken. This component will provide an API to which every data is sent, there it will check the validity of the data model and either sent it to the context broker or return an error code. At this point it is unclear, if a generic or custom-made error code will be returned to the source of the erroneous data input.

Exact components and programming language are to be decided, to conform with component 4.5.64.5.3, TALEND is regarded as favorite, but other options are in the process of being evaluated, e.g. a specific python program.

Functional Requirements

- Check the validity of the incoming data to the component 4.5.6 regarding its conformity of the data model. Accept and pipe the data to component 4.5.6 if valid, otherwise return an error and stop the data.

Non-Functional Requirements

- Performance: Since this will be a fast look-up, the performance should be high.
- Scalability: Since this will be a fast look-up, the task can be easily parallelized since there are no dependencies.
- Portability and compatibility: TBD.
- Reliability, availability, maintainability: It is maintained by DISY and each local aggregator during the project 24/7 and is based on the chosen data models in component 4.5.3.
- Security: NAIADES platform cloud securities and NAIADES digital signature.
- Usability: End user will not access this module. This component will be used directly by the provider of the overall NAIADES platform server.

Diagram

Internal diagram currently under design. See Figure 11 for module location on NAIADES architecture.

Supported APIs

- NGSIv2 and NGSI-LD RESTful API.
- Java API (TBD).

4.5.6 Context Data Management

Description

Context Data Management (CDM) takes in charge of managing context information at large scale coming from IoT devices and other public and private data sources. This component plays a central role in the NAIADES architecture that makes live data available and accessible to data consumers (e.g., AI modules, HMI, DSS, marketplaces) collected from different data sources.

It is consisted of two major modules: real-time data management and time-series data management.

The real-time data management supports real-time event processing of context events by analysing event streams. It provides functionalities to discover and register the context sources and offers different ways to interact with data sources and external applications by implementing read and write access for context information. The time-series data management provides historical data management. It is optimized for the handling of data organized by time. Time series are finite or infinite sequences of data items, where each item has an associated timestamp.

The communication is through the NGSIv2 and NGSI-LD based API, which are a simple yet powerful Restful API enabling to perform updates, queries or subscribe to changes on context information. The APIs of the CDM define operations and data structures to enable communication and exchanging of information between CDM and the other architectural components of NAIADES platform.

Functional Requirements

- Manages context information.
- Provides APIs for posting and consuming data (current values and historical values).
- Follows NGSI-LD standard.
    - e.g. subscription and notification features.

Non-Functional Requirements

- Performance: Real-time context management supporting real-time user experience.
- Portability and Scalability: Containerized deployment for portability and scalability.
- Interoperability: Any data following NGSI data models can be supported.
- Operability and availability: Automated operation and available 24/7 without human intervention.
- Security: Token based data credential provided by identity management and access control modules as well as NAIADES cloud security module.
- Usability: End user will not access this module.

Diagram

Please refer to Figure 11. NAIADES architecture modules and layers overview.

Supported APIs

- NGSIv2 and/or NGSI-LD RESTful API.

4.5.7 Data Repository

Description

Data Repository is a component that will store the final measurements and historical data. The Data Management module (4.5.6) will use the Data Repository to store processed and modelled data, in order to read it when queried by other components.

Data Repository is made up of a Data Base and a backend application and will run in the NAIADES cloud platform.

- Data Repository data base

  The Data Repository data base is a relational Data Base, PostgreSQL, that contains tables and the relations of it, in order to store modelled and processed data that came from Data Collection Aggregation pilots. In this database, data can be accessed and persisted not directly but using the Data Repository backend application.

- Data Repository backend

  Data Repository backend is a Java application used to let NAIADES platform to make operations on the database. It exposes web services to the Naiades Data Management, using http protocols, so to persist or access data from the database. The Data Repository backend contains a model DTO mapped to the database tables, and private connection configuration to the database.

  The backend also provides logging mechanism and a JDBC driver in order to establish the connection to the database. To ensure better access and persist capabilities, the backend can use typical Hibernate and JPA tool annotations for persist or query for an entity to table mapping, or custom complicated join queries like JDBC, HQL or Query Generators.

  Because of security reasons and user access and process roles, the backend will ask Identity Management (4.5.8) and will check Data Signature (4.5.22) of the user at every service call, as a first step of starting a process. The process of the access data and update/store/delete data will be synchronous.

Currently, it is still pending the exact definition of the tables and the relations between them.

Functional Requirements

- Data repository provide support to store in database processed data from pilots. Context Data management will provide the sets of data that will be stored in the DB and the sets of data to be obtained from it.

Non-Functional Requirements

- Performance: Data Repository Database needs medium performance and medium storage for default binary, with one schema database and few test tables, in around of 550 MB and medium processors. The size of the physical space repository will grow up depending on the number of tables and its columns numbers/types and the volume of data to be contained. The Data Repository Backend needs medium performance, and no physical space for storing. It will handle concurrent access through springboot framework and from database concurrent access layer.
- Scalability: The number of accesses and the amount of data saved are not limited, they depend only on the physical performance of the server machine. Data repository microservice is containerized .
- Portability and compatibility: Both database and backend are dockerized and can be run on any environment.
- Reliability, availability, maintainability: Written using maintained technology and the written code can be customized from any point of view.
- Security: User signature to authorize the requests.
- Usability: It can be used and accessed internally just for the Naiades internal modules.

Diagram

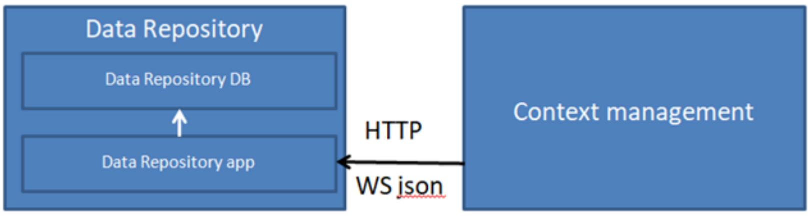

Figure 18. Data Repository Module Diagram.

Supported APIs

- NGSIv2 and/or NGSI-LD RESTful API.

4.5.8 Identity Management

Description

Identity Management is a component of NAIADES platform that stores and handles users, both internal (pilots' users) and external (Marketplace users) NAIADES users. Identity Management is the first layer for getting the authentication of a user (external of internal). Based on the token Data signature, it will check the user role and authorize or not some internally processes.

It is a microservice application that contains a database repository and a backend repository application that handles http web service requests and persists or gets info from the database.

- The Identity Repository DataBase is a MySQL database which will store users and their roles.
- The Identity Management Backend will generate an authentication token, at login, for users, based on row credentials as username and password. This token must be saved locally by each external application that calls NAIADES platform (such as Data Collection Aggregation, Marketplace, Operational And Management Tools, or external application who will integrate NAIADES

platform via Marketplace) at login/authentication time and sign each request to the platform for processing or visualizing NAIADES data.

Functional Requirements

- Identity Management has the role of authenticating all applications integrated with NAIADES Cloud platform, mainly from external NAIADES applications like Data Collection Aggregation, Marketplace, Operational and Management tools, and external public applications via Marketplace.

Non-Functional Requirements

- Performance: TBD.
- Scalability: The number of accesses and the amount of data saved are not limited, they depend only on the physical performance of the server machine. It can be accessed from any internal microservice in case if new services are developed that require authorization and authentication The microservice is containerized.
- Portability and compatibility: Both database and backend are dockerized and can be run on any environment.
- Reliability, availability, maintainability: Written using maintained technology.
- Security: Integrates security dedicated tools for authorization process OAuth2.
- Usability: It can be used and accessed both from inside and outside NAIADES cloud platform.

Diagram

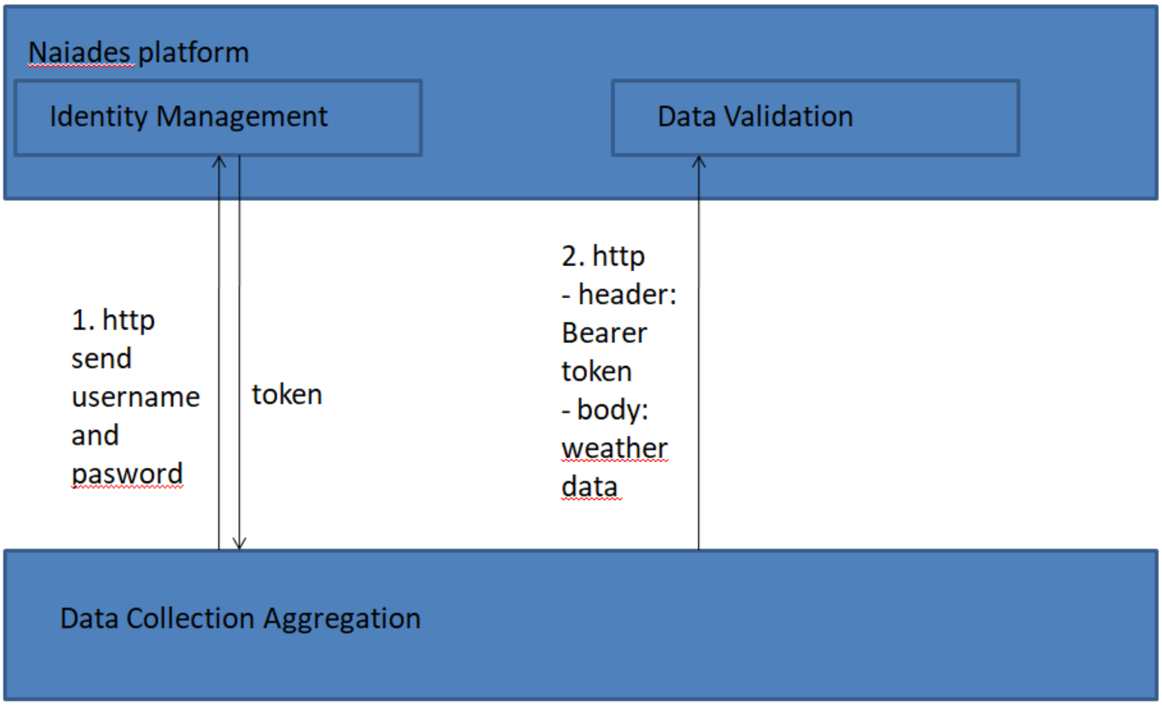

Figure 19. Identity Management Module Diagram.

Supported APIs

- NGSIv2 and/or NGSI-LD RESTful API.

4.5.9 Data Fusion

Description

Data fusion is a software component which periodically gets data from (4.5.6) data repository / (4.5.5) Context management module, combines it and performs statistical aggregation/analysis , exposing the results through a dedicated connector to the (0) Data model validation module.

Functional Requirements

- Provide relevant information extracted from raw data collected (e.g. daily average value extracted from a 1-minute granularity time series) as output and will send it to the data model validation component.

Non-Functional Requirements

- Performance: The module must be able to process data from at least 1000 parameters every minute with delays <10 seconds.
- Scalability: The module will support data fusion from at least 3000 parameters.
- Portability and compatibility. It will run on cloud premises. Compatible with any Linux distribution.
- Reliability, availability, maintainability: 24/7.
- Security: Tokenized and encrypted connections.
- Usability: Not critical, end user will not access this module.

Diagram

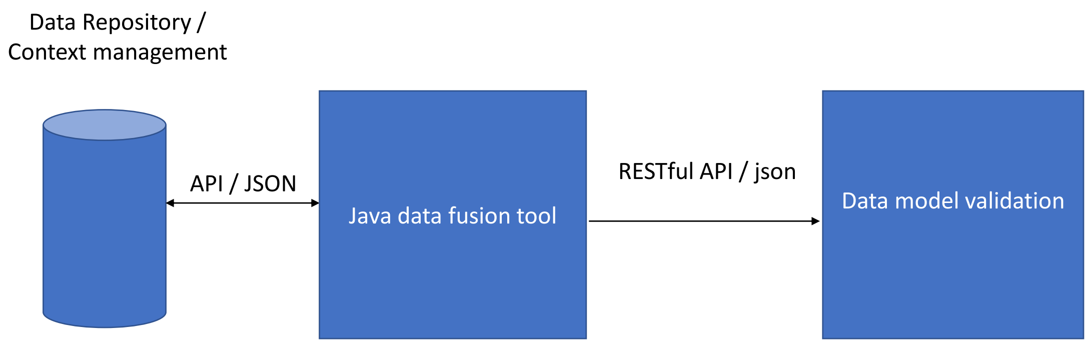

Figure 20. Data Fusion Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.10 Weather Forecast

Description

The Weather Forecast Toolkit is a holistic approach that will provide predictions for a variety of weather-related values such as temperature, precipitation, humidity and wind. For each variable, a different model will be developed using historical values of the given variable. The models will be built on the premises of artificial intelligence; therefore, it will teach itself to understand patterns and correct itself when it makes a bad prediction.

The training and testing of the algorithms will take place in a personal PC in CERTH's premises. The PC is equipped with a NVIDIA Titan 1080Ti graphics card, i7 processor, 32 GB RAM and 16TB of hard disks. It runs Ubuntu 18.04 LTS OS and Python 3.7 is set within Anaconda environment. All models will be built with TensorFlow and related libraries. For deployment, a cloud service, like Amazon Web Services, will be considered in order to run the toolkit entirely in the cloud.

This component will serve two purposes: as a standalone service that will display the predictions of each variable through elaborate visualization, as well as for other artificial intelligence modules that will use these predictions as input for their models. Its interoperability with other services will engage other technical partners into identifying the specifications of the Weather Forecasting Toolkit so that it will provide optimal information for them.

End-users will have at their disposal a highly personalized general-purpose Weather Forecasting Toolkit that will be used by their experts in order to make decisions even for matters that are not directly linked to the water services of NAIADES.

Functional Requirements

- Provide weather forecast to the visualization platform.
- Provide weather forecast as input to other machine learning services.

Non-Functional Requirements

- Performance: Weather forecast AI model should be able to re-train itself every appointed timeframe (minimum 3h or on-demand if it is decided like this). Depending on the amount of historical data, the number of parameters that will be used for the neural network, the forecasting depth and the available hardware, the model should be able to train and produce prediction in a matter of minutes (<10 for the best and <60 for the worst case scenarios).
- Scalability: There will be a variety of weather variables, granularities and time depths. So probably multiple models will be running providing predictions for all different combinations of the aforementioned prediction variables in order to cover all needs for weather forecast.
- Portability and compatibility: The model will be able to run (train and predict) both in local server located in CERTH, but also in cloud infrastructures i.e. AWS, however this has not been decided yet. Also, the weather toolkit will be dockerized, thus it will be able to run in different environments.
- Reliability, availability, maintainability: The weather prediction toolkit will run 24/7 and will be monitored by CERTH in order to deal with any problems that occur, as well as its regular maintenance.
- Security: All required measures of security needed.
- Usability: Not critical, only CERTH will have access to this module.

Diagram

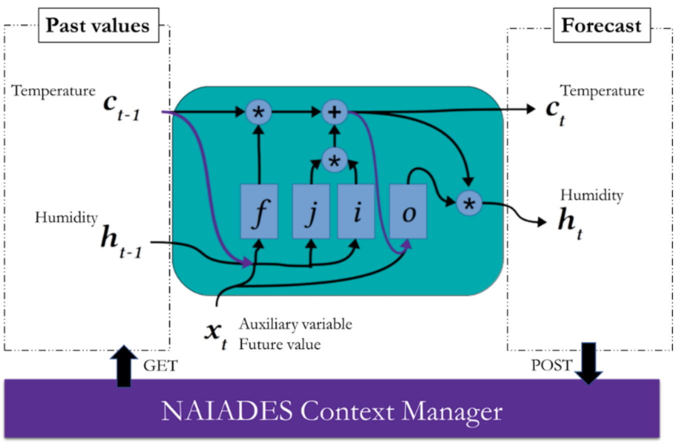

Figure 21. Weather Forecast Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.11 Urban Water Models

Description

The purpose of this component is to produce data of critical events to feed the AI platform. This is because the existing / planned sensors will not be able to detect such events, because of their low probability of occurrence during the project. Our algorithms run existing open source urban modelling systems such as EPANET and SWIMM, in order to produce data for critical scenarios to feed the AI. The urban water modelling systems must be provided by the pilots and must be open source (e.g., EPANET, SWIMM). If other modelling systems are in place, a dedicated modelling system will be built.

Functional Requirements

- To complement the data that is collected by sensors, with data generated by water models using scenario-based critical events.
- In the relevant use cases, to offer optimized operational decision making.

Non-Functional Requirements

- Performance: the obtained pressures and discharges will be subject to the accuracy for which the models were built and calibrated.
- Scalability: the method is standard for any water distribution model built in EPANET.
- Portability and compatibility: the application can run in Windows.
- Reliability, availability, maintainability: The models will run offline and feed a database for the AI, and this will be done one time.
- Security: No specific security issues.
- Usability: The urban water model component will not be accessed by end user, but its outputs will support other modules.

Diagram

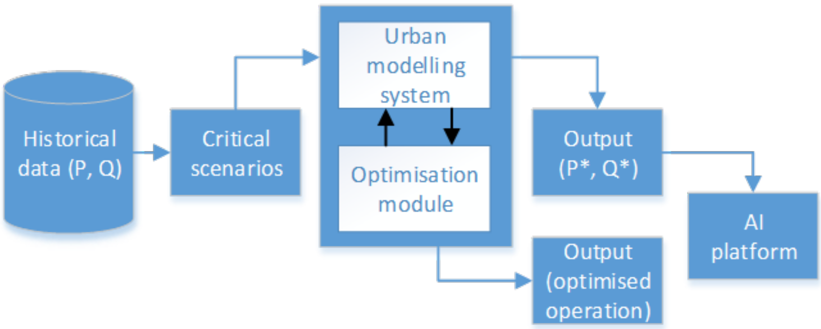

Figure 22. Urban Water Models Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.12 Water Quality Treatment Models

Description

Water from natural courses is fully treated before being supplied to a distribution system from where it will go to consumers in drinking water supplies systems. Water treatment consists in sequential units to eliminate pollutants and pathogens. It usually includes pre-treatment; coagulation, flocculation and sedimentation; filtration and disinfection. Performance models can help in understanding and predicting treatments effectiveness, especially in stream events when abstraction water changes, like during storms or droughts. If these models have good accuracy, they could use for treatment control purposes in order to ensure that quality of the treated water needs and to mitigate risks for the potable water supply.

In NAIADES project, water treatment models will be used to help operators in plants to monitor and control the process. This will be held in a decision support system that will make recommendations for better performance of the plant taking into consideration water quality in real time at abstraction point.

The goal of this component is to be able to detect and react to events related with the quality of water in a dynamical, reactive data driven approach and to increase the performance and safety of the current systems.

Mathematical models are usually used to describe and simulate different processes such as the sedimentation, coagulation-flocculation, filtration, aeration, chemical oxidation, or granular activated carbon adsorption that are used for water treatment. Artificial intelligence, and in particular machine learning tools such as artificial neural networks (ANN) are robust technologies that can handle the complex and dynamic nature of water treatment processes.

This module implements an artificial intelligent support system, using state of the art machine learning techniques, to provide useful information to the experts on the selection and application of appropriated water treatments for the current scenario.

AI models will be trained using historical quality, treatment dosage and weather parameters, so they are able to predict the best treatment dosage values. Data will also be generated on a lab scale WTP at AIMEN where the results will be validated. The historical quality parameters and the expected optimal quality parameters (established by law) are used as input and the historical dosage values as labels to train machine learning algorithms in order to create a model able to collect the relations between water quality and dosage values. Different machine learning algorithms will be used ranging from low to high computational power required, such as Support Vector Machine, Convolutional Neural Networks and Recurrent Neural Networks, which usually is related to low to high precision models. The final model will be selected based on high precision low computation criteria.

The generated models will be included with a Python inference SW that will receive current values of water quality parameters from sensors (or predicted values from component 4.5.13) in order to predict the best current (and future treatment parameters); to be presented as suggestions to the WTP expert staff.

The predictions will work continuously, collecting a series of input values and providing current predictions on an hourly basis. Additionally, it will collect quality forecasts from component 4.5.13 and provide predictions for each of the forecasts provided by that module (twice a day for a week). Only the last predictions (current and forecasted) will be stored in NAIADES repository so they can be retrieved by the users.

Functional Requirements

- Provided a time series of a list of quality parameters from the inlet of the WTP, the service will provide as output a prediction of the best treatment parameters (dosages, times) every hour.
- Provided a time series of a future forecast of the quality parameters (Component 4.5.13) of the inlet of the WTP, the service will provide as output a future prediction of the best treatment parameters (dosages, times).

- The outcomes will be provided in the adequate data models format, so it is approved by the data models validation component.
- The outcomes will be signed using the data signature block.

Non-Functional Requirements

- Performance: The forecasted range will depend on the frequency of the input. If the input frequency is daily, the forecast prediction cannot be shorter than one day. The objective is to provide hourly treatments predictions. The future predictions range will depend on the water quality predictions from Component 4.5.13 and the frequency of the inputs that trained the treatments model. The longest the prediction, the lowest the accuracy.
- Scalability: The models will be based on specific WTP processes (coagulation, filtration and chlorination). New models should be created for other WTP processes.
- Portability and compatibility: Integrated in NAIADES cloud platform and compatible with Windows and Linux distributions.
- Reliability, availability, maintainability: Maintained by AIMEN 24/7 during the project duration. It is required the creation of new models for any new WTP integrated in the platform.
- Security: NAIADES platform cloud securities and NAIADES digital signature.
- Usability: End user will not access this module. Only can be used for user with WTP with same processes.

Diagram

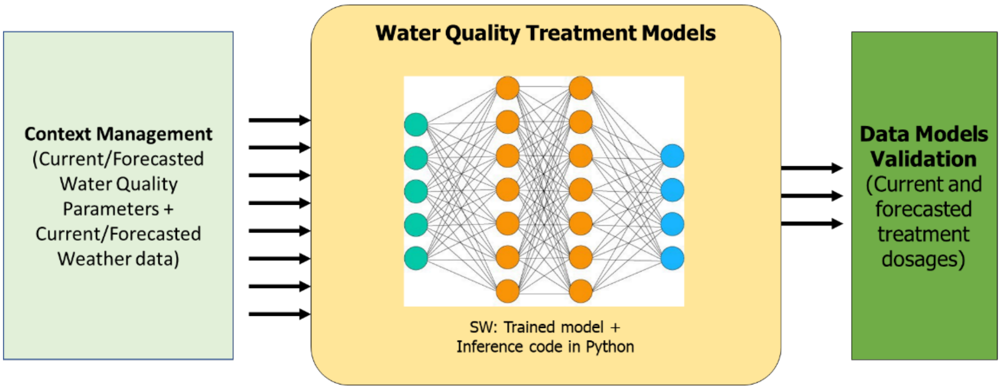

Figure 23. Water Quality Treatment Models Module Diagram

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.13 Water Quality Parameters Forecast

Description

Water sources generally require treatment prior to consumption/usage to ensure that they do not present a health risk to the user. Health risks from poor quality water will be often due to microbiological or chemical contamination.

Water quality in natural courses depends on a wide broad of aspects and it needs to be characterized periodically for water treatment: pathogens, pH, conductivity, suspended solids, organic matter, etc. Most of these parameters are usually determined by laboratory methods, so it is not real time, thus, at times, they are obtained too late to be useful to make a decision.

Water quality prediction could allow a quick response to extreme events that change quality characteristics in a short period of time, such a high organic load (i.e. algae blooms) or suspended solids (i.e. storms).

Water quality modelling involves the prediction of water pollution using mathematical simulation techniques. A typical water quality model consists of a collection of formulations representing physical mechanisms that determine position and momentum of pollutants in a water body. The rapid development of numerical models provides a large number of models to be used to assess water quality, thus machine learning techniques are the most adequate.

The goal of this component is to provide future values of water quality parameters to be able to react to events related with the quality of water in a dynamical, reactive data driven approach and to increase the performance and safety of the current systems.

AI models will be trained using historical quality and weather parameters, so they are able to predict future values of the quality parameters. The generated models will be included with a Python SW that will receive current values of water quality parameters and current and future/prediction values of weather parameters in order to foresee short and mid-term values of water quality.

The historical data will be used as input to train two types of algorithms regression algorithms such as SVR, and Recurrent Neural Networks (RNN) in order to create quality forecasting models; being the latter more robust when big data is available but they imply more computational cost. The best precision low computational model will be selected.

An inference Python code will collect current quality data and current and future weather forecasts from component 4.5.10 (weather forecast) and use them as input to the model. The inference code will supply as output the predicted future values of the relevant water quality parameters. The produced future quality parameters will be useful for DSS module and expert staff to take decisions about treatments, maintenance, water availability, etc.

Functional Requirements

- Provided a time series of a list of quality parameters and the forecasted future weather parameters, the service will provide as output an estimation of the future values of the required quality parameters for the following 7 days (two per day).
- The outcomes will be provided in the adequate data models format, so it is approved by the data models validation component.
- The outcomes will be signed using the data signature block.

Non-Functional Requirements

- Performance: The forecasted range will depend on the frequency of the input. If the input frequency is daily, the forecast prediction cannot be shorter than one day. The longest the prediction, the lowest the accuracy. The precision will depend on the precision of the weather forecasts.
- Scalability: Available to any new user able to provide the required inputs; being possible to create new models for specific new cases. The predictions frequency could be done hourly if required (being the inputs collected at least hourly).
- Portability and compatibility: Integrated in NAIADES cloud platform and compatible with Windows and Linux distributions.
- Reliability, availability, maintainability: Maintained by AIMEN 24/7 during the project duration. It is required the creation of new models for any new use case requiring different quality parameters.
- Security: NAIADES platform cloud securities and NAIADES digital signature.
- Usability: End user will not access this module. Only can be used for user with providing the same quality parameters.

Diagram

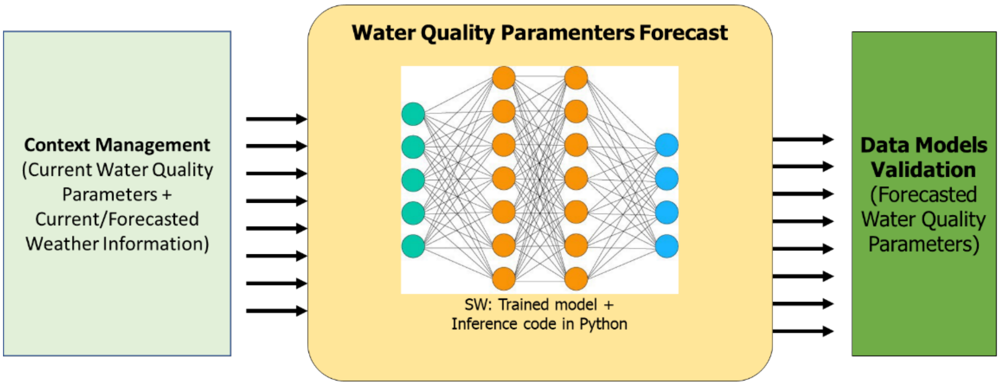

Figure 24. Water Quality Parameters Forecast Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.14 Water Consumption Monitoring

Description

The water consumption monitoring service will be analysing water pressure and flow data in order to classify current user water consumption behaviour.

The purpose of the service is to detect states, which leads, with a high probability, to a critical water consumption state. In such a way, water utilities can be notified in advance about potential critical event within the water supply system – so they are able to react in time to prevent such critical events.

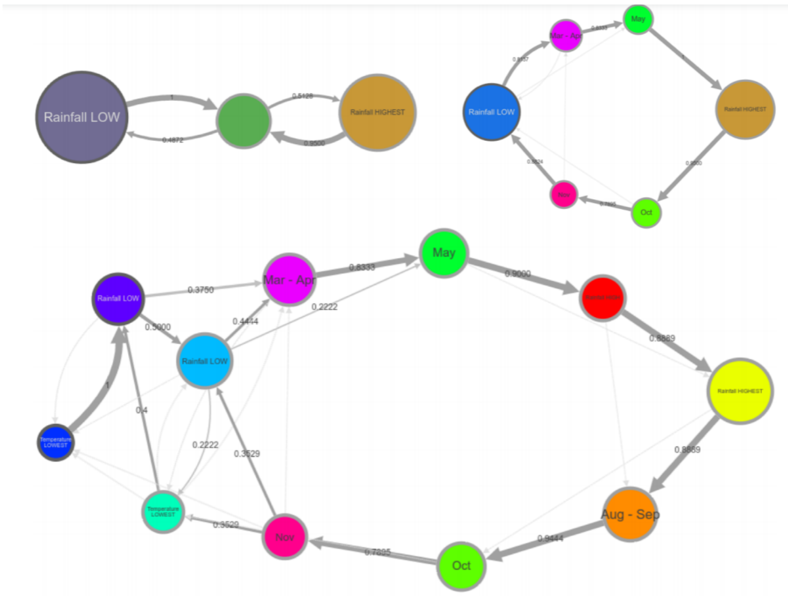

Figure 25: An example of a multi-level overview of the system (weather data analysis).

The system will be able to provide another representation of a complex system, which can be sensed with a set of timeseries. A person has difficulty comprehending multiple time-series simultaneously, therefore our system will transform these timeseries into a state graph and transitions within this state graph. For each state in the graph the user would be able to get a machine description based on the most prominent features in the state, however – it would be useful if a person would be able to provide their own annotations as well.

Based on transition matrix a Markov chain could be employed in order to make predictions. Also, other machine learning techniques could be used for this, which could exploit additional context information.

The majority of the computation is done when building the state models, inference steps are quite cheap.

The model will be built on the premises of artificial intelligence; therefore, it will teach itself to understand patterns and correct itself when it makes a bad prediction. It will operate as an artificial intelligence module for other modules that will use these predictions as input for their modules. It will be communicating with other NAIADES components through API: receiving needed input as well providing output in order for other services to be run.

The training and testing of the algorithms will take place on a local computer in JSI's premises. The computer is equipped with a NVIDIA Titan 1080Ti graphics card, i7 processor, 32 GB RAM and 16TB of hard disks. It runs Windows 10 OS and Python 3.7 is set within Anaconda environment. The inference will be integrated in NAIADES platform.

Functional Requirements

- Provide state graph of a system with manually annotated critical states.
- Provide inference model for state transition, including transition into critical states.
- Provide alerts related to possible critical events within the pipeline system.

Non-Functional Requirements

- Performance: The AI module should be able to make predictions within 5 minutes or less time frame.
- Scalability: The module will be able to perform on multiple state graphs. Live system will ensure parallelized deployment.
- Portability and compatibility: both, local and production environment will be able to run the AI module in terms of building the model as well as run it.
- Reliability, availability, maintainability: The AI module will be running 24/7 under JSI supervision for potential anomalies elimination.
- Security: There are no specific security issues related to AI module.
- Usability: The AI module will be communicating through predefined APIs and as such easy to use for (JSI / Naiades) IT engineers.

Diagram

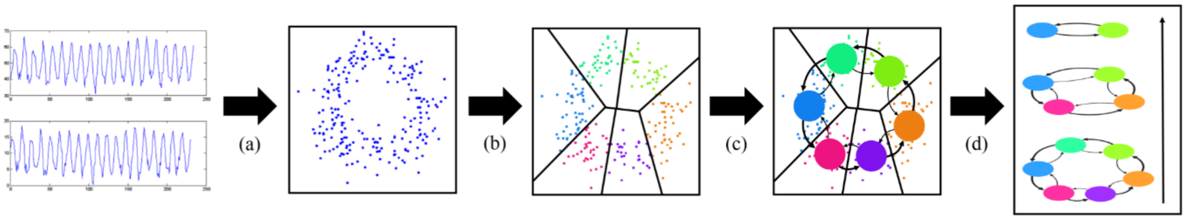

Figure 26: Overview of the systems pipeline. System is able to ingest multiple time series, (a) convert these series into multi-dimensional points, (b) cluster the multi-dimensional space into typical system states, (c) provide transition matrix for the state graph and finally (d) provide multi-level view of the system.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.15 Water Demand Forecast

Description

The water demand forecast will be providing predictions about water needs. It will encompass two subcases:

- Subcase1 will be predicting water consumption for a given urban area under the condition of granted access to its historical water consumption data. The purpose of the service is to enable water utilities to reduce costs in water delivery to their users. This subcase will be also divided in two:
    - Short-term predictions of water consumption up to 7 day in advance.
    - Long-term predictions of water consumption up to 1 year in advance.
- Subcase2 will be short-time predicting the times for plants to be watered. The purpose of the service will be enabling DSS support for watering public gardens with the goal of reducing water consumption as well as time employees are spending on watering public gardens.

The purpose of the service is to enable water utilities to reduce costs in water delivery to their customers.

The system for short-time predictions is based on a classical Big Data lambda architecture. Lambda architecture assumes a single-entry point for data. Data is here divided into two pillars (batch and streaming). Batch pillar is responsible for offline data analytics, potential batch model preparation. Streaming pillar brings the machine learning models to life. It is able to create enriched feature vectors from raw data streams, it is able to fuse these data streams into comprehensive feature vectors and those can then be used in different applications. The components will provide online (incremental learning) components for anomaly detection and prediction and their batch counterparts.

Online models are completely independent. They only require initial configuration (e.g. definition of the feature vectors) while learning and inference processes are then automatic. According to our initial experiments incremental models are:

- PRO:
    - Faster
    - More robust
    - Require no re-training
- CONTRA:
    - Less accurate

Offline models are traditional machine learning models, that can be based on standard statistical learning techniques such as linear regression, decision trees, random forest, gradient boosting, or any version of neural networks configurations, including deep learning. These models are slower, require bigger ecosystem to support their production deployment, they require constant re-evaluation and re-training, however, it is very important to know that these models are usually much more accurate.

The most important feature of modelling is however not the choice of the model, but generation of feature vectors. In these components we will develop innovative incremental data fusion process, that only relies on the streaming data.

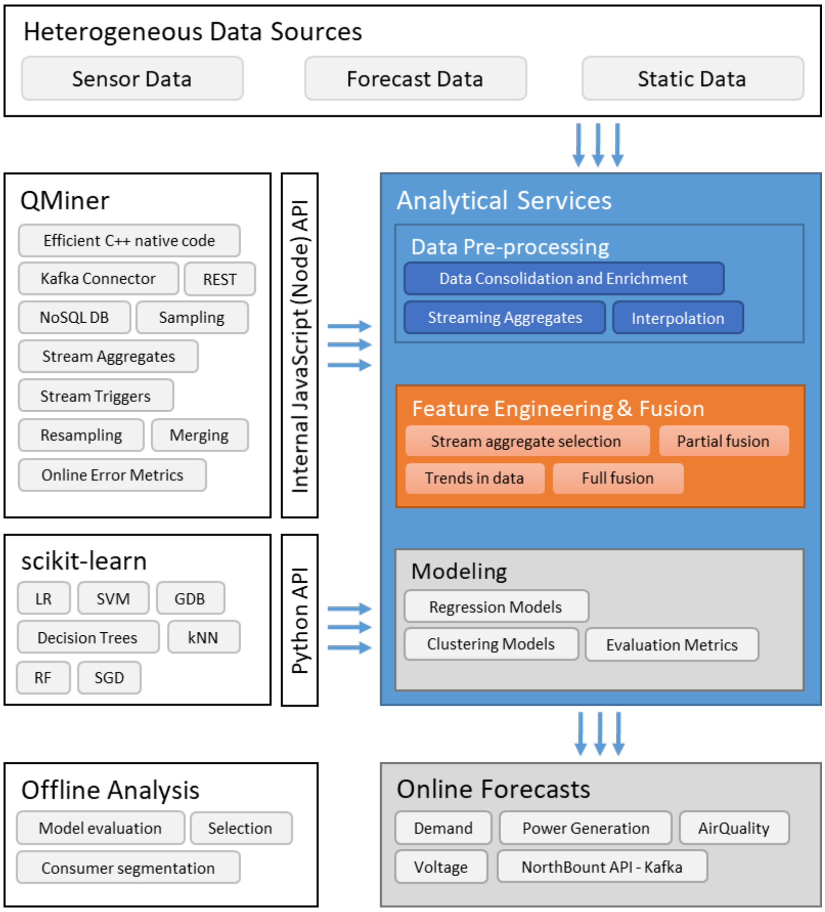

Figure 27: Internals of data fusion and modelling components.

The data fusion and incremental modelling part of the framework will be developed with NodeJS implementing advanced streaming analytics functionalities of QMiner in-house open-source library. Batch models will be provided within the framework based on Python. Python is the most popular platform for machine learning and therefore it is logically to exploit the huge ecosystem of solutions. Due to nature of the system (many independent sensors), the system can be easily parallelized and therefore there is no need for the usage of dedicated big-data frameworks.

The model will be built on the premises of artificial intelligence; therefore, it will teach itself to understand patterns and correct itself when it makes a bad prediction. It will operate as an artificial intelligence module for other modules that will use these predictions as input for their models. It will be communicating with other NAIADES components through API: receiving needed input as well providing output in order for other services to be run.

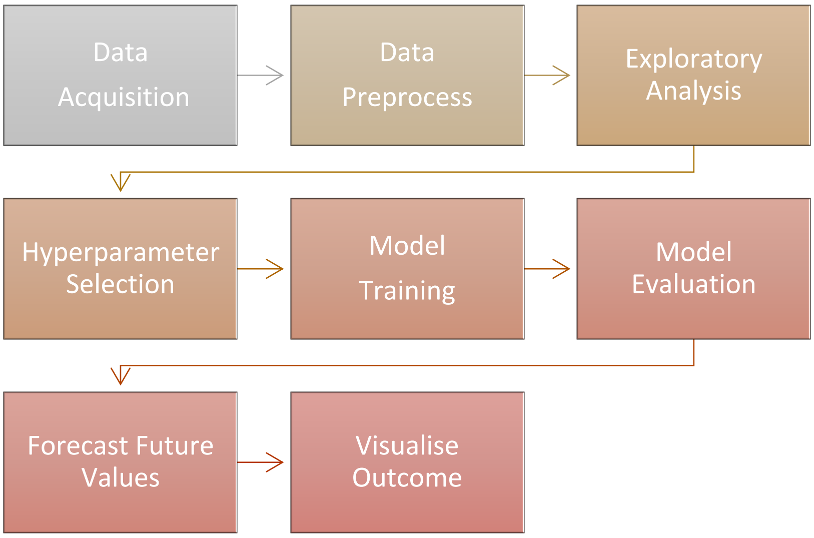

Figure 28. Long-Term Prediction Approach

In the **long-term** water demand prediction, a batch processing approach is used to analyse and train the model that was built to forecast the future water demand in monthly and biweekly intervals. Time series analysis is put to use in order to utilize the data that was retrieved. After the initial preprocess of the historical data an exploratory data analysis takes place to define the necessary alterations on the time series before the fitting of the model but also define the time series properties and characteristics.

An important part of the prediction process that is performed before the training of the model is hyperparameter tuning. The method helps the model make more accurate predictions and avoid the case of overfitting. The training face is achieved with the help of ARIMA and SARIMA forecasting models that suit best the structure of the data that is processed. For the evaluation of the model, prediction errors are utilized to validate the accuracy of the fitted model and its predictions.

After the training and the evaluation part, the forecasted values are produced for the next few years. The last step is to visualize monthly and biweekly prediction graphs which contain a comparison of the historical values with the forecasted outcome.

The training and testing of the algorithms for short-term predictions will take place on a local computer in JSI's premises. The computer is equipped with a NVIDIA Titan 1080Ti graphics card, i7 processor, 32 GB RAM and 16TB of hard disks. It runs Windows 10 OS and Python 3.7 is set within Anaconda environment. The training and testing for long-term predictions will take place on KT's premises. The inference will be integrated in NAIADES platform.

Functional Requirements

- Provide water demand prediction for 24 hours in advance for a given urban specified area (Subcase1).
- Provide predictions for when the plants are to be watered (Subcase2).

Non-Functional Requirements

- Performance: The AI module should be able to predict water consumption within a time frame of 60 seconds or less.
- Scalability: The AI module will be highly scalable and be able to provide predictions for at least a couple of hundreds of use-cases.
- Portability and compatibility: both, local and production environment will be able to run the AI module in terms of building the model as well as run it.

- Reliability, availability, maintainability: The AI module will be running 24/7 under JSI supervision for potential anomalies elimination.
- Security: There are no specific security issues related to AI module.
- Usability: The AI module will be communicating through predefined APIs and as such easy to use for IT engineers.

Diagram

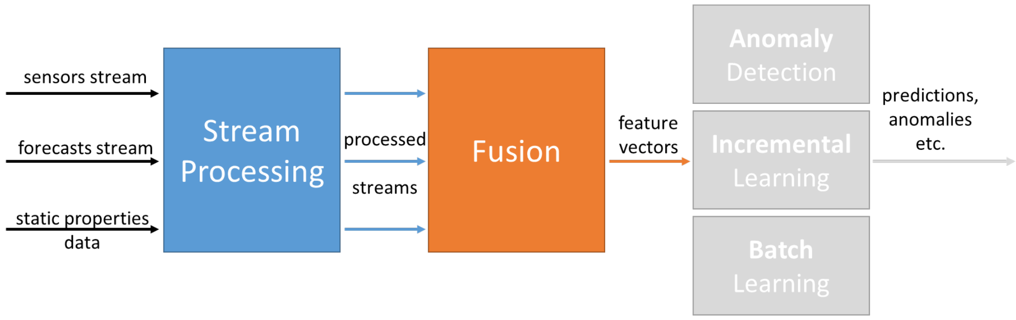

Figure 29: Conceptual architecture for batch learning, incremental learning, and anomaly detection. This figure above depicts conceptual architecture of data-processing pipeline, which is valid for different scenarios (monitoring, forecasting and anomaly detection). Data ingestion is done via API connectors as described below, however, internally the components us their own APIs for communication.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.16 Failure and Leakage Prediction

Description

The "Failure & leakage prediction" will be looking for outliers within the data provided.

Currently it will be detecting outliers for two subcases:

- Subcase1 will be monitoring water pressure and flows in order to detect consumption anomalies (e.g. pipe leakage) and will provide alerts when such anomaly will be identified.
- Subcase2 will be monitoring the presence of saline level within the water to detect anomalies in water quality (e.g. saline intrusion) and will provide alerts when such anomaly will be identified.

The module purpose is to reduce time and costs for water utilities to diminish unwanted water losses within the pipeline system.

The model will be built on the premises of artificial intelligence; therefore, it will teach itself to understand patterns and correct itself when it makes a bad prediction. It will operate as an artificial intelligence module for other modules that will use these predictions as input for their modules. It will be communicating with other NAIADES components through API: receiving needed input as well providing output in order for other services to be run.

The training and testing of the algorithms will take place on a local computer in JSI's premises. The computer is equipped with a NVIDIA Titan 1080Ti graphics card, i7 processor, 32 GB RAM and 16TB of hard disks. It runs Windows 10 OS and Python 3.7 is set within Anaconda environment. The inference will be integrated in NAIADES platform.

Functional Requirements

- The module will be providing alerts about anomalous measure points within the pipeline system on a daily level.

Non-Functional Requirements

- Performance: The AI module should be able to make predictions within 5 minutes or less time frame.
- Scalability: The AI module will be highly scalable and able to provide predictions for at least a couple of hundreds of use-cases.
- Portability and compatibility: both, local and production environment will be able to run the AI module in terms of building the model as well as run it.
- Reliability, availability, maintainability: The AI module will be running 24/7 under JSI supervision for potential anomalies elimination.
- Security: There are no specific security issues related to AI module.
- Usability: The AI module will be communicating through predefined APIs and as such easy to use for (JSI / Naiades) IT engineers.

Diagram

The same infrastructure will be used as described in previous subsection (4.5.15).

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.17 Consumer confidence

Description

Currently this module is under reconstruction. The initial idea was discarded due to the lack of useful data, so it has being repurposed and it is awaiting confirmation (please consult the forthcoming deliverable D5.7 for further detail). The new model will consist of a Global Water Observatory that aims to provide the user with insightful information from heterogeneous data sources (including news, social media, indicators, biomedical research) that provide several perspectives on the priority topics of NAIADES.

This module is being built as an external dashboard where the user can explore a variety of ingested datasets and their highlights, where text mining and data mining state-of-the-art methods take an important role. The input is taken mostly from external sources over several APIs and the data will be ingested to JSI's servers and all computations done on-premises.

As in 4.5.17, the infrastructure required is on JSI's side including a computer equipped with a NVIDIA Titan 1080Ti graphics card, i7 processor, 32 GB RAM and 16TB of hard disks. It runs Windows 10 OS and Python 3.7 is set within Anaconda environment. Also here, the inference will be integrated in NAIADES platform and some of the data can be sourced over APIs.

Functional Requirements

- The module will be providing exploratory dashboards allowing the user to investigate water-related problems in the context of news, indicators, scientific research, etc.
- Some of the exploratory data modules will provide API access for further 3rd party integration
- As complementary to what is available in the Global Water Observatory, the JSI will also make available focused dashboards that are comprehensive and extend capabilities of research (see D5.7 for more detail)

Non-Functional Requirements

- Performance: This module will be performing independently as an external asset of the NAIADES platform
- Scalability: The used technology is known to be highly scalable

- Portability and compatibility: both, local and production environment will be able to run the module in terms of building the model as well as run it.
- Reliability, availability, maintainability: The module will be running 24/7 under JSI supervision for potential anomalies elimination.
- Security: There are no specific security issues related to this module.
- Usability: The module will be communicating through predefined APIs and as such easy to use for (JSI / Naiades) IT engineers.

Diagram

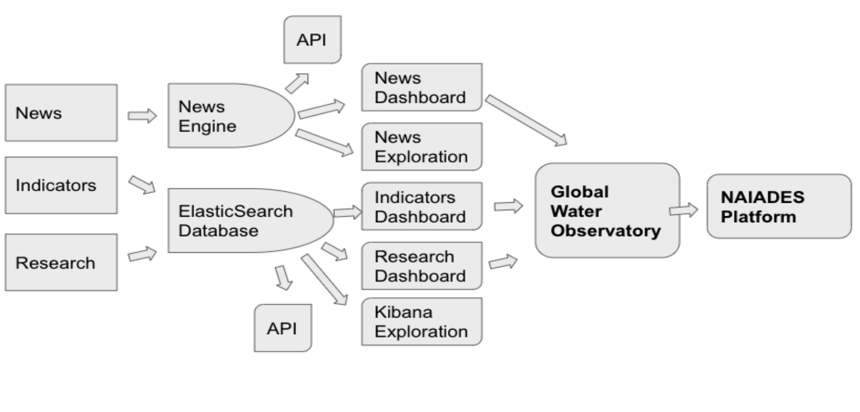

Figure 30. Consumer confidence - Global Water Observatory diagram

Supported APIs

- N/A

4.5.18 Awareness and behavioural Change support

Description

The water consumption awareness and behavioural change support hub creates water consumption awareness by coupling data science, HCI and psychology research into advanced data-driven water consumption analytics exposed through web interfaces. It also supports water consumers to change their behavior with respect to water use therefore boosting sustainability.

The Water consumption awareness and behavioural change support hub consists of three main modules: (i) A web-based application for the efficient watering of flower boxes, which enhances city employees' awareness on the water needs of plants and makes use of the NAIADES AI services to provide optimal scheduling of watering activities, leading to reduced water consumption and watering-related costs; (ii) A web-based dashboard for water management companies and public officials that makes use of the NAIADES data aggregation middleware and optimally water consumption data to interested stakeholders, in order to monitor and understand how water is consumed in a specific area or consumption point, while considering contextual parameters such as the time of consumption, weather conditions etc. Decisions on water consumption mitigation measures can rely on such information and such measures can be monitored after their implementation; (iii) A behavioural change support web-based application for inducing sustainable water use behaviours among water consumers with a focus on school students, a group of consumers that can provide a channel for generating great impact as on the one hand they will evolve to the responsible citizens of tomorrow, while on the other hand they can transfer the knowledge, attitudes and behaviours they shape to their families, leading to a cascading effect.

Functional Requirements

- Provide visual representations of behavioural change and awareness strategies (e.g. comparisons, self-monitoring, suggestions, etc.) and communicate them to the users through web interfaces.

Non-Functional Requirements

- Performance: The web-based applications of the water consumption awareness and behavioural change support hub should provide 5 seconds or less response time in a Chrome desktop browser, including the rendering of text and images.
- Scalability: The solution shall be able to support an annual growth of 10% of new users.
- Portability and compatibility: the web-based applications of the water consumption awareness and behavioural change support hub should be compatible with all major browsers
- Reliability, availability, maintainability: The web-based applications of the water consumption awareness and behavioural change support hub should be available 99 percent of the time.
- Security: Some of the data presented through the water consumption awareness and behavioural change support hub may be confidential. NAIADES has to include the necessary mechanisms for authentication and access control, as well as mechanisms to guarantee data protection at the data communication layer. The water consumption awareness and behavioural change support hub will adhere to such mechanisms and specifications.
- Usability: The web-based applications of the water consumption awareness and behavioural change support hub should be easy to learn and navigate; buttons, headings, and help/error messages should be simple to understand, while the applications should appear easy to use, rather than intimidating, demanding and frustrating.

Diagram

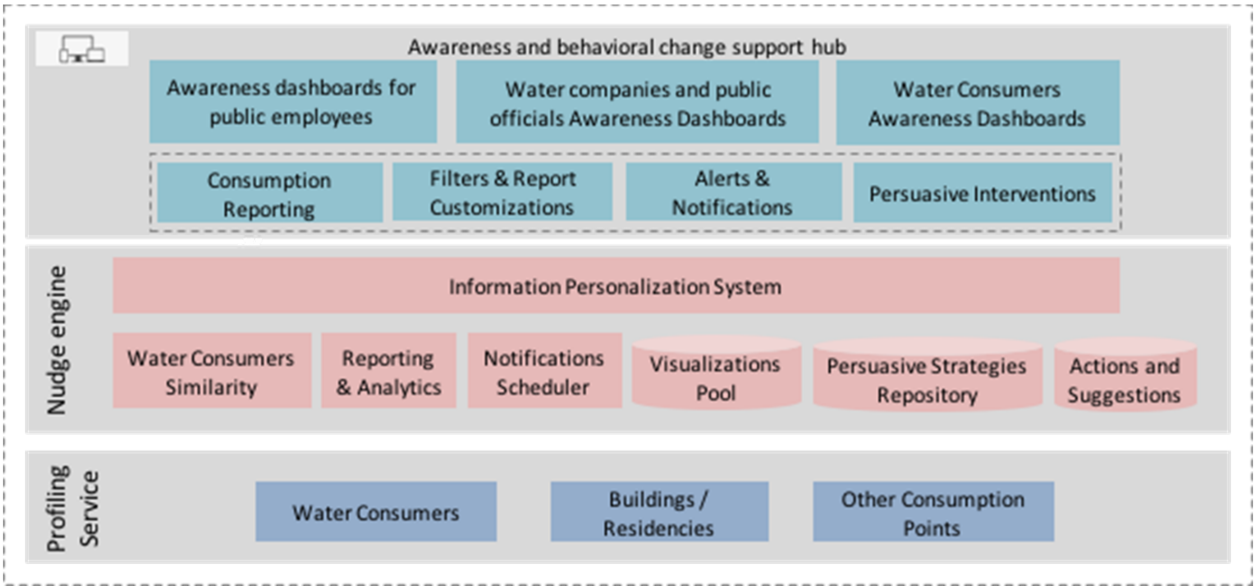

Figure 31. Awareness and behavioural Change Support Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.19 Decision Support System

Description

The Decision Support System (DSS) component resides on the user's side. It shall consist of advanced information processing mechanisms, fully utilising raw data and measurements from NAIADES components dealing with data collection (e.g., Weather prediction, Water demand, Consumer confidence). DSS shall integrate lower level decisions and alerts that lead to high-level decisions and plan suggestions that are sent to NAIADES front-end (HMI) via a REST API.

The DSS will combine information received from IoT infrastructure and the AI toolkits and based on the semantic models (WP4 & WP5) produced as well as from all stakeholders involved in the complete value chain. The data will be potentially received via HTTP using web service based on industry-standard web-services protocols (SOAP/REST) and formats (XML and JSON) provided from the partners involved in the ongoing processes and they will be processed in real time. They will be coupled with the associated requests to certain parts of the value chain, SOP (standard operating procedures) and response strategies. To conclude we can say that we are going to process near Real-time & time series data regarding Water quality, demand, consumption etc.

Functional Requirements

- The DSS system shall provide feedback to the involved internal or external actors or suppliers, in terms of actionable knowledge and recommendations for next steps

Non-Functional Requirements

- Performance: The DSS shall process the incoming data and provide its output in near real-time
- Scalability: The DSS shall be able to support larger volumes of data and increase its capabilities in a cost-effective manner over time.
- Portability and compatibility: DSS's API shall provide outputs compatible with all major browsers and devices.
- Reliability, availability, maintainability: The DSS shall be operational and available most of the time (~99%).
- Security: The DSS shall be compliant with the NAIADES mechanisms and specifications to guarantee data protection and authentication and access control.
- Usability: The DSS shall be user-friendly, easy to navigate and use, and should provide outputs to the user in an easy to understand and friendly way.

Diagram

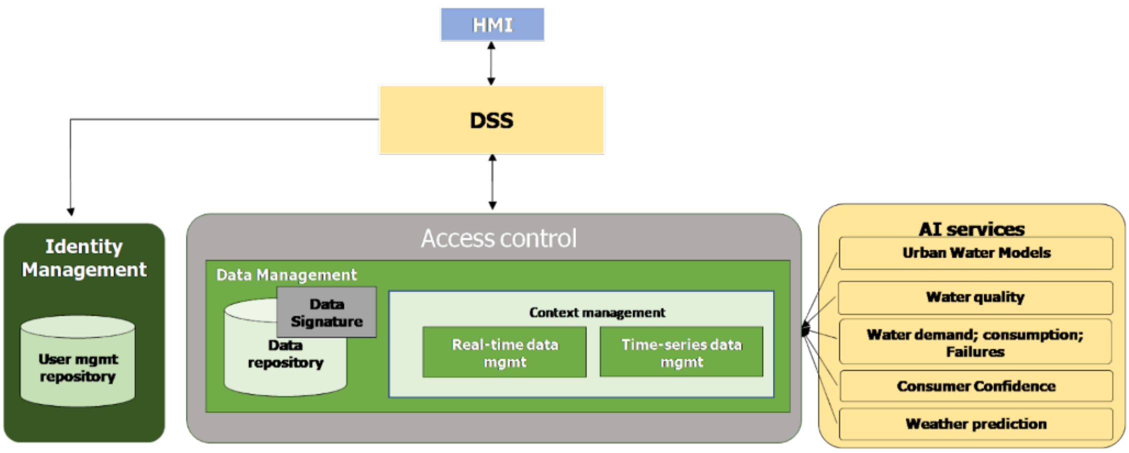

Figure 32. Decision Support System Diagram

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.20 Human Machine Interface

Description

HMI shall be the administrative aspect of the frontend part of the NAIADES platform. It shall display the relevant to the user outcomes of the rest of the toolkits. HMI is being developed to be a Reactjs + Redux application in a dockerised format.

Functional Requirements

- User Login/ Logout
- Provide information about the status of the NAIADES platform
- Display the outputs of several components in an intuitive way and friendly way
- Enable user interaction with the NAIADES platform

Non-Functional Requirements

- Performance: The HMI shall visualise the user-requested output in near real-time and in a friendly, comprehensible way.
- Scalability: The HMI aspires to increase user acceptance and number of users over time, utilising its user-friendly design.
- Portability and compatibility: The HMI shall be compatible with all major browsers and responsive to devices like smartphones and tablets.
- Reliability, availability, maintainability: The HMI shall be operational and available most of the time (~99%).
- Security: The HMI shall be compliant with the NAIADES mechanisms and specifications to guarantee data protection and authentication and access control.
- Usability: The HMI shall be user-friendly, easy to navigate and use, assuming characteristics like i) pleasing screen design; ii) symmetrical layouts; iii) appropriate arrangements of options/menus; and iv) informative plots and graphs.

Diagram

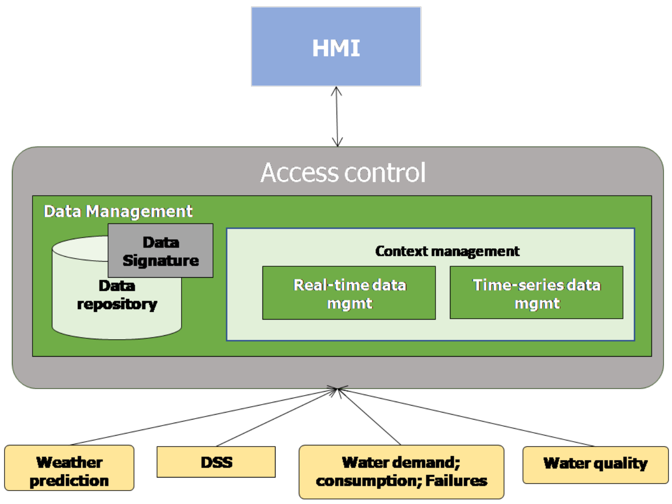

Figure 33. Human Machine Interface Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.

4.5.21 Marketplace

Description

This is a public application, accessible from the public users on the internet, created in order to market solutions/services of NAIADES application. External applications can be integrated with Naiades services through Marketplace application API.

Marketplace application lets users external to NAIADES to preview NAIADES' main services presenting a short description of them. Marketplace offers internationalization in three languages. Marketplace users must be logged in for viewing specific and important information for integration with NAIADES platform, such as services' API documentation. In order to do that, Marketplace frontend contains a register form and a login form, for creating an account and login. Register, login and logout will call directly NAIADES Identity Management for user authorization.

NAIADES Marketplace is divided into two parts:

1. Public NAIADES services

   It allows public users to see the NAIADES services exposed and a short description of them, in many languages. Marketplace does not require a user authentication to show this information.

2. NAIADES services API

   It allows users to see external NAIADES API documentation and to test and access it (when possible). It is required user authentication, that consists in a register and login procedures.

   Once user is authenticated in Marketplace frontend platform, he can see all API documentation as a url of the web service, type of the request, accept type, payload request, and access token to sign the request. Also, it will be possible to test the API directly in the frontend.

In order to use NAIADES services, from an external application, the user must create an account in NAIADES Marketplace frontend. After that, the user can call /loginPublic, using the credentials created before, to obtain the token. After loginPublic, it can call all services (using the token authentication) that Marketplace backend expose which are documented in Marketplace frontend.

The main functionality regarding showing API and test or call it, has the next flow:

1. User register or login in frontend form.

   Marketplace will send the user and password to the Identity Management, and get back an authentication token which is stored in frontend browser cookie.

2. User call shows API

   Marketplace frontend will call Marketplace backend with authentication token, backend will call Identity Management to verify the token received from frontend, and if it is ok, it will respond back to the frontend with needed API documentation.

If the user calls API from external application, the external application calls first /loginPublic, based in the credentials created in NAIADES Marketplace frontend, and get a token. After that call needed API of the Marketplace backend using that token, backend receives the request ask Identity Manager for validity of the token, and finally, send the request to the NAIADES platform Data Management.

Functional Requirements

- NAIADES Marketplace exposes NAIADES services (description and APIs) to public users.
- Marketplace also offer support to external applications for integration and know-how.

Non-Functional Requirements

- Performance: Marketplace frontend and Marketplace backend does not have hard processing functions; they just send data to the NAIADES Data Management for hard processing. Both frontend and backend can be installed and run in minimum to medium performance machines, and size of target files are around of 150 - 200 MB.
- Scalability: Marketplace user access will depend on internal Identity Management (for authentication and authorization.) performance. The application is containerized and will use internal NAIADES Data Management layer for storing data and process some data. Marketplace will not have dedicated user and data storage layers.
- Portability and compatibility: Yes, both frontend and backend are dockerized and can be run on any environment.
- Reliability, availability, maintainability: Yes, because it is written using maintained technology and the written code can be customized from any point of view.
- Security: Yes, because we used authentication token to authorize the requests and can be secure using https protocol.
- Usability: It can be used both public and by authentication.

Diagram

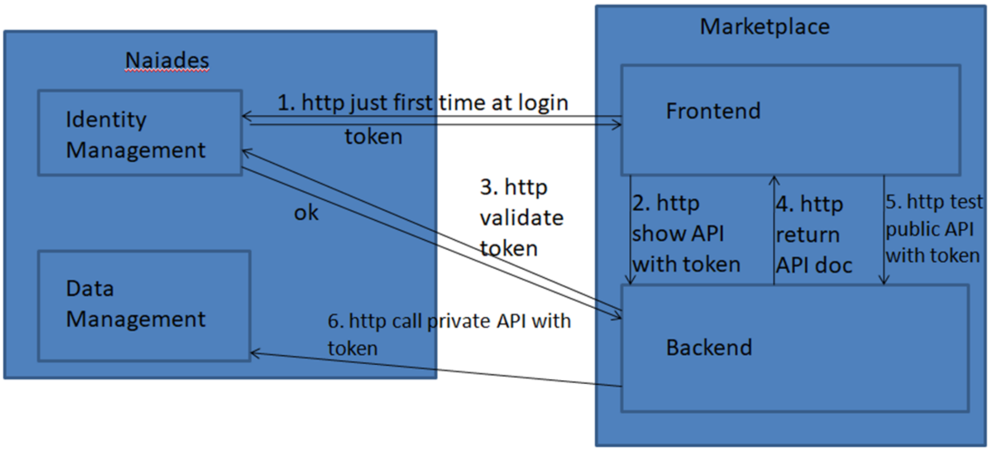

Figure 34. Marketplace Module Diagram.

Supported APIs

- NGSIv2/NGSI-LD RESTful API.
- Java.

4.5.22 Data Signature & KSI Gateway server

Description

Signing next to the data input source gives added value as it is possible to analyse data movement in the system. The platform operator will benefit from lower management costs and transparency that allows prevention of misuse. Users will be clear that data is not manipulated after the signing and that the service providers are not responsible for the events after the signing.

KSI signing software is next to the data generators, such as the collector/aggregator and AI services, allowing data signing close to source to provide data integrity as early as possible. KSI signatures are independently verifiable proofs of integrity, signing time and signing entity.

Functional Requirements

- Gateway Server (KSI Gateway block)- a software component at the customer premises providing access to the Aggregation or Extender Service. A dedicated Gateway or cluster of Gateways is usually installed for an organization to provide the service to the users of that organization.
- Application Integration (Digital signature block) - Guardtime provides fully featured SDK-s for C, Java, GO and .NET to facilitate KSI service integration to customer applications.
- The way KSI Gateway is deployed in terms of network zones depends on the network architecture of the user organization and how the applications are going to use it for signing. A classic example is that an organization has a KSI Gateway deployed in their network and it provides access to KSI services for the users of that organization.
- The server is not accessible by third parties without access control mechanisms.
- Gateway should be behind a firewall or have local iptables/firewall configured to allow traffic only from authorized IP addresses.
- Gateway should be accessible to applications that sign data and extend signatures. Corporate deployments usually expect access to the Gateway to be through secure network.
- Gateway should have access to upstream Aggregators and Extenders in KSI network. This communication is not required to be over secure networks.

Non-Functional Requirements

- Reliability, availability, maintainability: 24/7.
- Performance: 1 000 000/1 sec.
- Scalability: depends on conditions (up to 100 000 000 users).
- Usability: not critical, end user will not access this module.
- Portability and compatibility: KSI SDK is provided for C, Java, GO and .NET to facilitate KSI service integration to applications.

Diagram

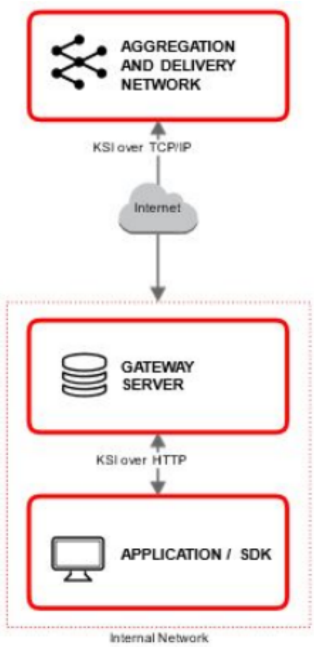

Figure 35. Data signature Diagram.

Supported APIs

- KSI SDK (Java, C, GO, .NET) to facilitate KSI service integration to customer application.

4.5.23 MIDA Cloud Security

Description

MIDA is an important component for system owners and operators, who need to be ready for the cloud environment failures. MIDA allows to collect, sign, and monitor cloud configurations and machine statuses. State configuration statuses are captured, normalized, and secured. These states are baseline for any security alerts, policy validation and dashboard use. Key features of MIDA:

- Provide real-time and provable awareness to detect changes across the infrastructure.
- Decrease time to detection of misconfigurations, unauthorized access to resources and deployment of assets.
- Provide cryptographically immutable inputs to event correlation and detailed analysis.
- Decrease cost of governance and audit via monitoring objects

MIDA consist of different modules, each with its own specific function:

- MIDA Agents and Services create immutable and portable state snapshots. MIDA Agents capture state within applications, servers, and other infrastructure assets. MIDA Services capture the events directly from the cloud infrastructures through API and integrated polling.
- MIDA State Management Services consist of internal applications that process, persist, protect, and visualize data gathered by MIDA Agents and Services.
- KSI Blockchain provides data authenticity, time, and participant identity. These are achieved through the creation of the KSI Signature that seals the snapshots created by MIDA Agents and Services.

Functional Requirements

- Provable and real-time continuous state and integrity monitoring to detect misconfigurations or malicious changes across the infrastructure.
- Cryptographically immutable and portable evidence of actual individual machine or cloud infrastructure states.
- Streamlined and accurate state capture.
- Enhanced durable data for analytics.

Non-Functional Requirements

- <u>Reliability, availability, maintainability</u>: 24/7.
- <u>Performance</u>: 1 000 000/1 sec.
- <u>Scalability</u>: depends on conditions (up to 100 000 000 users).
- <u>Usability</u>: not critical, end user will not access this module.
- <u>Portability and compatibility</u>: KSI SDK is provided for C, Java, GO and .NET to facilitate KSI service integration to applications.
- <u>Security</u>: server is not accessible by third parties without access control mechanisms.

Diagram

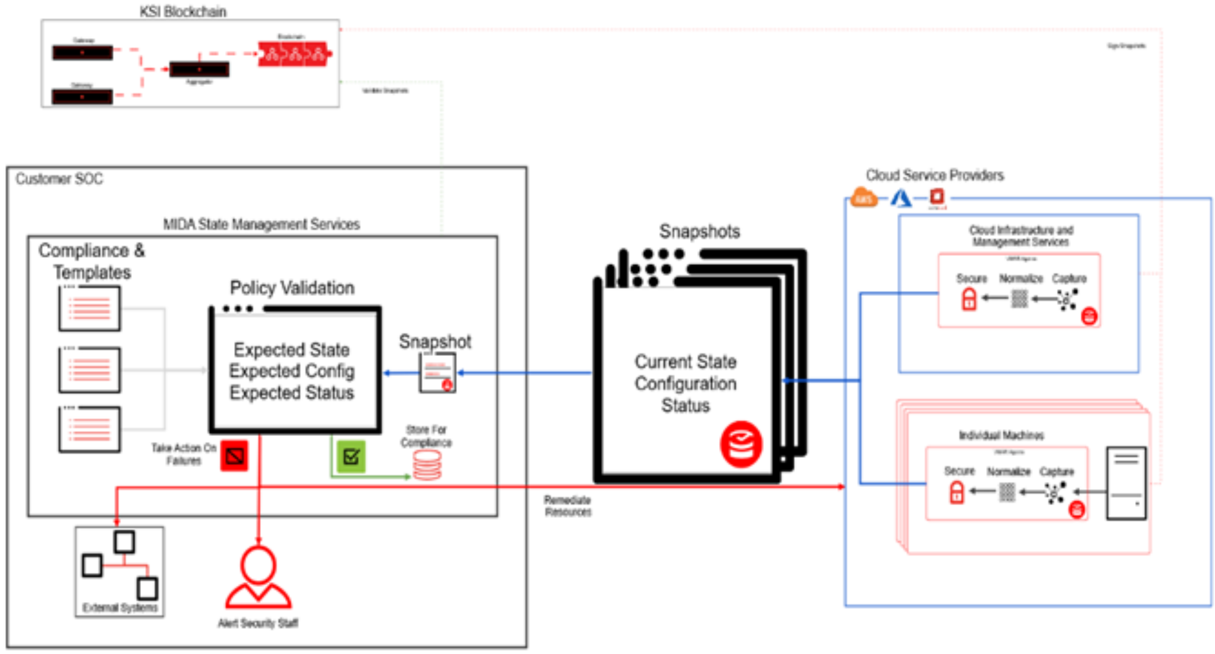

Figure 36. MIDA Cloud Security Diagram.

Supported APIs

- JAVA.

4.6 Deployment Overview

This Section provides insights on the physical implementation and deployment of NAIADES system by describing the SW and HW requirements of each component.

4.6.1 Data Collection/Aggregation

ALICANTE

Software Requirements

- Operating systems: n/a.
- Programming language: Java application.
- Software frameworks & libraries: n/a.
- Server: n/a.
- Database: MongoDB temporary Raw data repository.

Hardware Requirements

- CPU: n/a.
- RAM: 8 GB.
- HDD: TBD. Physical storage needs will depend on amount of parameters and poll period.

BRAILA

Software Requirements

- Operating systems: Microsoft Windows (for develop and test), Centos 7(for production).
- Programming language: Java 8.

- Software frameworks & libraries: spring-boot version: 2.1.6. RELEASE; apache-maven version: 3.6.0; CsvReader, Rest API, and custom java libs.
- Server: apache-tomcat 8 and docker (for develop and test) and Docker and Docker compose (for production); ProFTPD as FTP server for file repository.
- Database: no DataBase.

Hardware Requirements

- CPU: medium CPU like Intel I7.
- GPU: no needed.
- RAM: at least 8 GB RAM.
- HDD: for backend, DataModel and FTP server binaries files needs 300MB and for FTP storage needs 20GB.

CAROUGE

Software Requirements

- Operating systems: microservice, runs on docker.
- Programming language: Python3.
- Software frameworks & libraries: misc.
- Server: 5.53.108.182.
- Database:
  - [connectivity NOK] Dumps data on filesystem if no connectivity.
  - [connectivity OK] Pushes data to context manager and historical data base.
    - Component should authenticate towards IoT Platform.
    - Component should aggregate data using domain specific data models.

Hardware Requirements

- CPU: minimal.
- RAM: at least 4GB.
- HDD: at least 504.6.1 GB.

4.6.2 Environmental Monitoring

Software Requirements

- Operating systems: n/a.
- Programming language: n/a.
- Software frameworks & libraries: n/a.
- Server: n/a.
- Database: n/a.

Hardware Requirements

- It needs a LoRa Gateway in the existing LoRa WAN network.

4.6.3 Common Data Models

Software Requirements

- Operating systems: Any.
- Programming language: Java 8.
- Software frameworks & libraries: Based on the input data and of the TALEND distribution.
- Server: Integrated in the NAIADES platform server.

- Database: NAIADES repository.

Hardware Requirements

- See https://help.talend.com/ for up to date requirements for the ETL tool to create new components.

4.6.4 Cloud Platform

Software Requirements

- Operating systems: Ubuntu 18.04.5 LTS.
- Programming language: None (all modules to run in docker, or install to be discuss in case of Kubernetes).
- Software frameworks & libraries: Yum, NGINX, telnet, docker, docker-compose, others (depends on other modules needs).
- Server: Amazon Web Services Cloud Hosting Server. (DNS, IP: TBD).
- Database: n/a.

Hardware Requirements

- CPU: 4 X CPU about 2.3 GHz.
- GPU: n/a.
- RAM: 16 GB RAM.
- HDD: 250 GB Storage.

4.6.5 Data Models Validation

Software Requirements

- Operating systems: Any.
- Programming language: TBD.
- Software frameworks & libraries: TBD.
- Server: Integrated in the NAIADES platform server.

Hardware Requirements

- CPU: TBD.
- RAM: TBD.
- HDD: TBD.

4.6.6 Context Data Management

Software Requirements

- Operating systems: Linux.
- Programming language: C++.
- Software frameworks & libraries: Docker, Shell, Linux.
- Server: Integrated in the NAIADES platform server.
- Database: MongoDB.

Hardware Requirements

- CPU: 4xCPU.
- RAM: at least 8GB.
- HDD: TBD. Depends on volume of transit data.

4.6.7 Data Repository

Software Requirements

- Operating systems: All operating systems are allowed since it use docker or event Java based server like tomcat that has support for Windows, Linux, etc.
- Programming language: Java for backend and PostgreSQL for database.
- Software frameworks & libraries:
  - DataBase: PostgreSQL 11 relational database, access using jdbc driver from backend application.
  - Backend: Java 8(intent to improve to java 11 or 14), SpringBoot 2, Apache Maven 3.6.0, Hibernate, JPA, JDBC, custom libraries.
- Server:

  Data Repository: database with backend can be run using their own docker and docker compose image. Also they can be run separately:
    - The DataBase can be installed using installer or binary kit on different OS as Windows, Linux, etc.
    - The Backend can also be run in a java server like Tomcat, Jboss, WebLogic, etc.
- Database: n/a.

Hardware Requirements

- CPU: Medium performance (eg: i5 or i7).
- GPU: n/a.
- RAM: Medium performance (eg. 8gb-16gb).
- HDD: HDD or SSD.
    - Databse_around of 550 MB default, depending on tables size.
    - Backend_around of 70MB default, depending on mapping java entity–table complexity (not bigger than 100 MB).

4.6.8 Identity Management

Software Requirements

- Operating systems: All operating systems are allowed since it uses docker and docker compose scripts.
- Programming language: TBD.
- Software frameworks & libraries: TBD
- Server: NAIADES cloud.
- Database: MySQL internal docker image.

Hardware Requirements

- CPU: Medium performance (eg: i5 or i7).
- GPU: n/a.
- RAM: Medium performance (eg. 8gb-16gb).
- HDD: TBD.

4.6.9 Data Fusion

Software Requirements

- Operating systems: n/a.
- Programming language: Java application.
- Software frameworks & libraries: n/a.
- Server: Integrated in the cloud.
- Database: NAIADES repository

Hardware Requirements

- CPU: n/a.
- RAM: 4GB.
- HDD: TBD - Physical storage needs will depend on number of parameters and poll period.

4.6.10 Weather Forecast

Software Requirements

- Operating systems: Linux OS, Ubuntu 16.04 LTS.
- Programming language: Python 3.7.
- Software frameworks & libraries: Jupyter Notebook, Tensorflow, Keras, Pandas, Numpy.
- Server: Either locally in CERTH or in cloud platform such as AWS.
- Database: TBD.

Hardware Requirements

- CPU: Intel 7 (at least 6^{th} gen).
- GPU: NVIDIA GTX 1080Ti.
- RAM: 32GB.
- HDD: 256GB.

4.6.11 Urban Water Models

Software Requirements

- Operating systems: Windows.
- Programming language: Python.
- Software frameworks & libraries: EPANET (existing water distribution modelling software)
- Server: will be offline.
- Database: data files in CSV or Excel format can be produced to be imported to AI database.

Hardware Requirements

- CPU: n/a.
- GPU: n/a.
- RAM: n/a.
- HDD: n/a.

4.6.12 Water Quality Treatment Models

Software Requirements

- Operating systems: Windows and Linux.
- Programming language: Python 3.7.
- Software frameworks & libraries: Anaconda, Python libraries (scikit-learn, pandas, statsmodels, tensorflow, cudnn), CUDA libraries.
- Server: Integrated in the NAIADES platform server.
- Database: NAIADES repository.

Hardware Requirements

- CPU: i7.
- GPU: Nvidia GTX 1080Ti or superior.
- RAM: 16GB.
- HDD: Order of Kbytes – few Mbytes.

4.6.13 Water Quality Parameters Forecast

Software Requirements

- Operating systems: Windows and Linux.
- Programming language: Python 3.7.
- Software frameworks & libraries: Anaconda, Python libraries (scikit-learn, pandas, statsmodels, tensorflow, cudnn), CUDA libraries.
- Server: Integrated in the NAIADES platform server.
- Database: NAIADES repository.

Hardware Requirements

- CPU: i7.
- GPU: NVIDIA GTX 1080Ti or superior.
- RAM: 16GB.
- HDD: Order of Kbytes – few Mbytes.

4.6.14 Water Consumption Monitoring

Software Requirements

- Operating systems: Linux OS / Windows / OSX.
- Programming language: > Python 3.6+.
- Software frameworks & libraries: Anaconda, Jupiter notebooks, NodeJS 8+.
- Server: Integrated in the cloud.
- Database: NAIADES repository.

Hardware Requirements

- CPU: i7.
- GPU: NVIDIA GTX 1080Ti.
- RAM: 32GB.
- HDD: 100GB – 1T.

4.6.15 Water Demand Forecast

Software Requirements

- Operating systems: Linux OS / Windows / OSX.
- Programming language: > Python 3.6+.
- Software frameworks & libraries: Anaconda, Jupiter notebooks, NodeJS 8+.
- Server: Integrated in the cloud.
- Database: NAIADES repository.

Hardware Requirements

- CPU: i7.
- GPU: NVIDIA GTX 1080Ti.
- RAM: 32GB.
- HDD: 100GB – 1T.

4.6.16 Failure and Leakage Prediction

Software Requirements

- Operating systems: Linux OS / Windows / OSX.
- Programming language: > Python 3.6+
- Software frameworks & libraries: Anaconda, Jupiter notebooks, NodeJS 8+.
- Server: Integrated in the cloud.
- Database: NAIADES repository.

Hardware Requirements

- CPU: i7.
- GPU: NVIDIA GTX 1080Ti.
- RAM: 32GB.
- HDD: 100GB – 1T.

4.6.17 Consumer confidence

Software Requirements

- Operating systems: Linux OS / Windows / OSX.
- Programming language: > Python 3.6+.
- Software frameworks & libraries: Anaconda, Jupiter notebooks, NodeJS 8+.
- Server: Own server at JSI premises.
- Database: Own repository at JSI premises.

Hardware Requirements

- CPU: i7.
- GPU: NVIDIA GTX 1080Ti.
- RAM: 32GB.
- HDD: 100GB – 1T.

4.6.18 Awareness and behavioural Change support

Software Requirements

- Operating systems: Windows, Linux.
- Programming language: GO, NodeJs, Python.
- Software frameworks & libraries: Django, Grafana.
- Server: All apps will provide in self-contained docker images.
- Database: Not relevant.

Hardware Requirements

- CPU: Intel or AMD processor with 64-bit support; 2.8 GHz or faster processor; 8 cores.
- GPU: Nvidia GeForce GTX 1050 or equivalent.
- RAM: 16 GB.
- HDD: 1TB.

4.6.19 Decision Support System

Software Requirements

- Operating systems: Linux.
- Programming language: Python.
- Software frameworks & libraries: Django REST.
- Server: self-contained dockerised image.
- Database: NAIADES repository.

Hardware Requirements

- CPU: Intel or AMD processor with 64-bit support.
- RAM: ~256 MB.
- HDD: <500 GB.

4.6.20 Human Machine Interface

Software Requirements

- Operating systems: Linux.
- Programming language: Javascript HTML, CSS.
- Software frameworks & libraries: ReactJS, Redux, ChartJs.
- Server: Self-contained dockerised image.
- Database: NAIADES repository.

Hardware Requirements

- CPU: Intel or AMD processor with 64-bit support.
- RAM: ~256 MB.
- HDD: <500 GB.

4.6.21 Marketplace

Software Requirements

- Operating systems: All operating systems are allowed it use docker or event java based server like tomcat that has support for Windows, Linux, etc.
- Programming language: Java and TypeScript.
- Software frameworks & libraries:
  - Frontend: Angular 8, NodeJs >11 (preferred 13), Material designer, Angular Materials, i18n internationalization, HTTPClient for http request.

- Backend: Java 8(intent to improve to java 11 or 14), SpringBoot 2, Apache Maven 3.6.0, custom libraries.

- Server:
  - Frontend can be run in the same server as backend run, or in an separate server. Frontend application can be run in Node js server or in any java based server.
  - Both frontend and backend can be run inside docker/docker-compose container. Both frontend and backend can be run in a java server like Tomcat, Jboss, WebLogic,, etc
- Database: - no Data Base

Hardware Requirements

- CPU: Medium performance (eg: i5 or i7).
- GPU: - No need.
- RAM: Medium performance (eg. 8gb-16gb).
- HDD: HDD or SSD (the applications takes around 150 MB).

4.6.22 Data Signature & KSI Gateway Server

Software Requirements

- Operating systems: Windows, Linux.
- Programming language: C, java, GO, .NET
- Software frameworks & libraries: https://guardtime.com/library/tsp.
- Server: virtual or physical.
- Database: SQL relational database.

Hardware Requirements

- CPU: 1 GHz
- GPU: n/a.
- RAM: 2GB.
- HDD: 2GB.

4.6.23 MIDA Cloud Security

Software Requirements

- Operating systems: Red Hat Enterprise Linux (RHEL) 7.1; Linux distribution CentOS 7.1.
- Programming language: java VM 8.
- Server: virtual or physical.
- Database: SQL relational database (PostgreSQL 11 preferred).

Hardware Requirements

- CPU: 1 GHz.
- RAM: 2GB.
- HDD: 2GB.

4.7 NAIADES Security Mechanisms

4.7.1 Authentication and Access control

NAIADES authentication and access control mechanisms provide a specific set of security features such as identity management, access control, authentication, and authorization. It is based on the OAuth2 standard protocol that will delegate and generate access token and protect NAIADES from unauthorized access. The identity and authentication management component will issue different roles and permissions to each user and application, where the access tokens can be used to grant or deny access to the APIs exposed by the platform components based on provided roles and permissions.

As an example, we have the Carouge city pilot (Figure 37), which generates data and needs to be push update requests to the NAIADES platform, where the context is stored. The security interaction required for authorizing and authenticating such request.

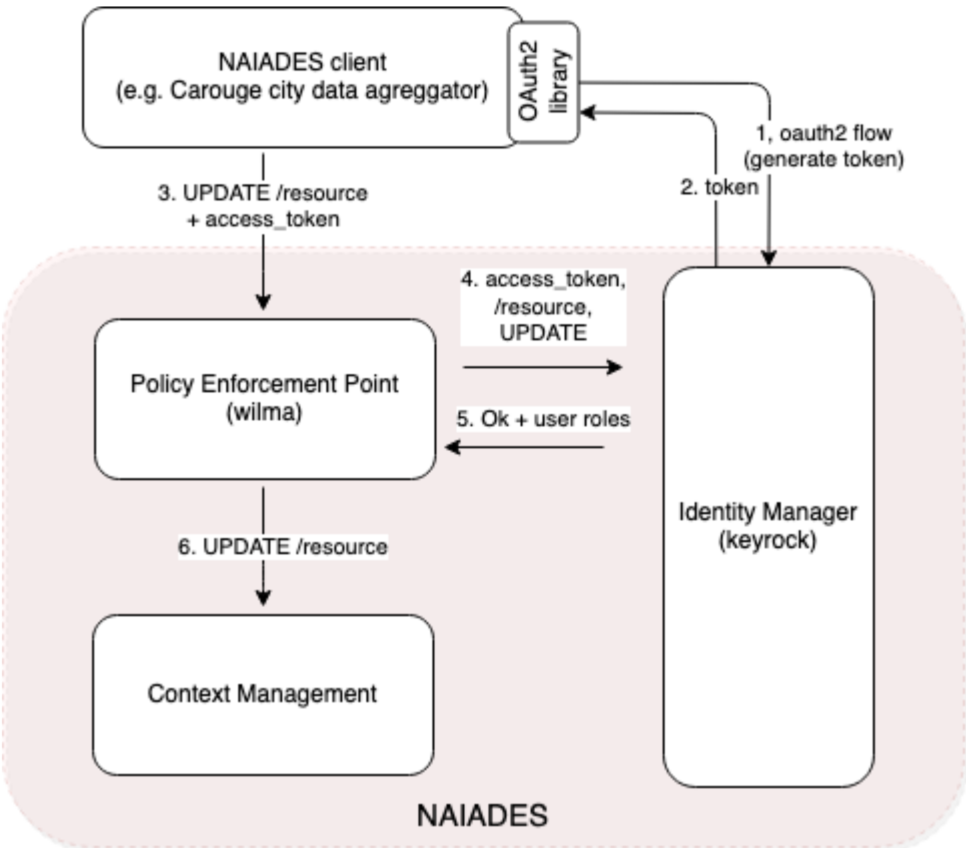

Figure 37. Carouge authentication and access control example.

The Identity and Authentication Management is the first step for accessing data, services, and applications. It provides secure and private identification and authentication for users, trust management, and Identity Federation towards applications. It provides management of user life-cycle functions by providing account creation and management, and enforcement of policies and procedures for user registration, identification, and authentication. It supports the enforcement of policies and procedures for user registration, secure and private authentication, and user profile management. In addition, it allows to link an application with the user account, in order to enable that application to authenticate on the behalf of users, by interacting with its Oauth2 APIs. This module interacts with the Authorization and Accounting component to exchange information on access request to assets.

The Authorization Management provides authorization and accounting capabilities which are critical aspects to support NAIADES services and applications. It enforces a set of conditions defining whether users have granted on the access to a specific resource. For the Authorization, it provides a Policy Enforcement Point (PEP) which intercepts resource access requests, makes access control decision requests, and enforces access control decisions. It also provides a Policy Decision Point (PDP) that evaluates access request by checking authorization policies for rendering an access control decision. Moreover, it provides a Policy Retrieval Point (PRP) that connects to the Policy Management component and a Policy Information Point (PIP) to obtain applicable authorization policies according to an access control decision request and attributes that are needed for evaluating authorization policies, for example the IP address of the requester, creation time of the resource, current time or location information of the requester. This information is combined in order to get a finial access control decision. This module interacts with the Identity and Authentication.

4.7.2 Data signature

NAIADES platform is using Guardtime KSI blockchain data signatures to protect the origin of the data.

Traditional solution for data authentication which relies on centralized trust authorities (Public Key Infrastructures – PKI) suffer from problems of scalability and resilience (single point of failure). Furthermore, data integrity relies traditionally on the 'hardened box' concept where perimeter security keeps 'bad' actors out and 'good' actors in. Data transferability is facilitated by checksums and key-based digital signatures, which rely on trusted functions like key management, certification infrastructure and providing root of trust. Guardtime's KSI blockchain technology enables massive scale data authentication without reliance on centralized trust authorities. KSI can be used to ensure data integrity, traceability, provenance, and auditability along all the data lifecycle (processing, formatting, logs etc.). The data changing history and event integrity can be retrieved from the blockchain security solution and the data validity, time of change and signing entity is ensured in a way that third party validation independent from system can be used.

NAIADES will include an innovative mature blockchain technology into its system which is provided by Guardtime. KSI is an industrial scale full stack blockchain infrastructure [11], the deployment of which offers a myriad of new security solutions [12] and service revenue opportunities for water utilities. It will enable the water service providers in the NAIADES platform to guarantee the state of data exchange network without relying on trusted administrators or the procedures that define the security of the network.

4.7.3 Cloud Security (MIDA)

For the protection of operating environment, MIDA product concept is used. As you can see from architecture diagrams, NAIADES relies largely on cloud environment. It is possible because we do not process delicate personal data and water utility companies do not compete directly. Still, there must be trust between data owners and NAIADES operator. Trust is also needed between AI development team, administrator, and utility. There is transparency layer that allows it.

MIDA contains 3 functional components. These are provision, manage, operate. Provision allows to enrol, provision and revoke cloud devices (VMs, router configurations). Management contains state monitoring of devices and changes/updates. Operate allows to create cryptographically verifiable data.

5 NAIADES Use Cases Architecture

Architecture description per pilot use case:

5.1 Alicante

Alicante is located in the southeast coast of Spain. It has semi-arid Mediterranean climate with no local water sources and extremely irregular rain events. The seasonal population variation is from 300k to 500k inhabitants, which produces seasonal variation of water demand. Water is transported from water bodies about 170km away from Alicante. In the last 20 years a reduction of 20% in the supplied volume has been observed, even though the number of inhabitants has increased in the same period by about 50,000. However, further reduction of water demand is required, and for this active awareness campaigns are needed. In terms of wastewater management, Alicante has two main treatment plants, with a combined capacity of 135,000 m3/day, and a current treated volume daily mean of about 69,000 m3. In both wastewater treatment plants reuse happens, one with 41.2% and the other with 35%. However, saline water intrusion in the drainage system, of the order of about 12% of the volume that comes to the wastewater treatment plants, prevents for increasing this rate further. The estimated total economic cost of saline intrusion is estimated to be around 1M Eur per year.

Therefore, in conjunction with the Alicante stakeholders, three main Use Cases in Alicante (UCA) have been identified in this pilot, namely UCA1) water demand forecast; UCA2) detection of saline intrusion to sewerage, and UCA3) Water consumption and awareness campaign.

- UCA1 considers a short term (1-day to 1 week), for operation purposes that aim to the optimization of water production and energy costs by fulfilling the water demand. To this end, several data sources are available, among others historic records, weather forecasts and eventual social networks data; UCA1 also can consider a medium-long term (more than three months) forecast, for planning purposes, with impacts in decisions related to water resources management, water purchase and finance.
- UCA2, detection of saline intrusion into the sewer system, considers the objective to detect and monitor saline intrusion by means of flow and conductivity sensors as well as smart data analysis. To this end, a small area with the following characteristics is to be defined: high density of sewer mains, pipes with high age of construction.
- UCA3, municipal water consumption and awareness campaign, includes two objectives. An analytical tool for municipal consumption and an awareness campaign based on real consumption data for schools. The main idea is to evaluate the consumption evolution per type (e.g., schools, sport facilities, gardens, fountains, etc.); also, to make comparisons among these types, seeking to provide “smart” informative messages.

Based on the three use cases defined above, the NAIADES components required for this pilot include mainly all NAIADES modules, excepting some AI services such as Water Quality forecasting (see Figure 38). Some of the services will only applied to specific use cases whereas others are common to all the use cases, such as the Data Fusion or the Urban Water Models services, in charge of pre-processing or generating new useful data for the other services. Furthermore, Consumer Confidence was added in this pilot but it is currently under modifications so it will change in the future version of the architecture.

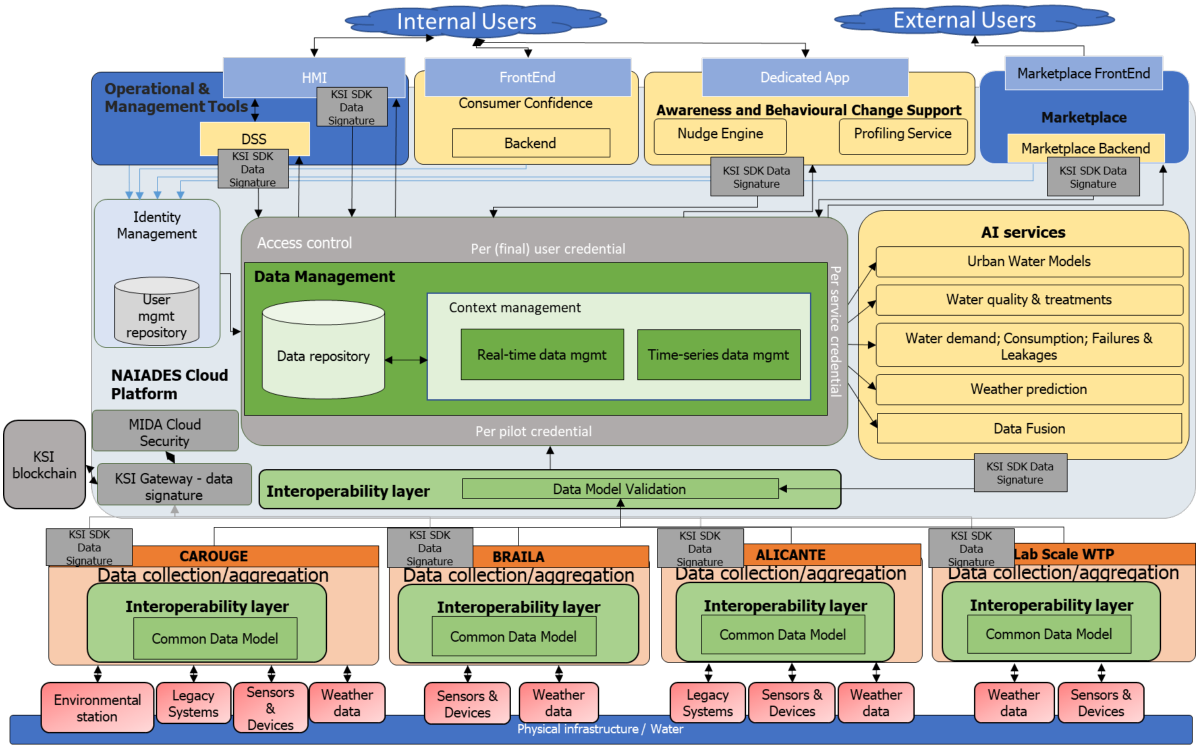

Figure 38. NAIADES platform deployment for Alicante pilot.

5.1.1 UCA1 – Water Demand Forecast

UCA1 in Alicante make use of existent measurements of water consumption, flow and pressures; and weather data from nearest stations (obtained from AEMET OpenData service [13]); thus no extra sensors are required. At aggregation level, the data is to be adapted to be sent to NAIADES platform, this is, acquired from current Alicante sources, converted to the specific data models (still under definition), signed and tokenized.

The UCA1 data that arrives to NAIADES platform is checked to corroborate the data models are free of errors and forwarded to the data manager. At this point, the tokens are reviewed to authenticate the source and provide access to the services linked with this pilot. The main services linked to this UC are Weather Prediction, Water Demand AI services and the DSS. Those services will be in charge of forecasting water consumption. Once the data is generated by the services, with the correct data models format, service signature and token, it is sent back to NAIADES Data Manager, to be distributed and/or stored in the repository. Before arriving to the data manager, the data models are previously checked, and they data is authenticated.

Finally, the HMI, when the internal Alicante user is connected, will request the relevant data to the data manager to be displayed via web in the form of graphs, tables, etc.

5.1.2 UCA2 – Detection of Saline Intrusion to Sewerage

UCA2 data will follow the same path and changes that UCA1 data. The main differences are:

- Data collection/aggregation: UCA2 will also make use of existent measurements but it will require the addition of extra sensors for conductivity and water level (the level value would be transformed to flow by AMAEM) in certain locations not monitored yet. Thus, it is required also to add the required technology to aggregate data from the new sensors.

- Services: The main services will be the Weather Prediction and Failures & leakages AI service and the DSS. They will be in charge of generating the data that shows abnormal levels of salinity that can be related to saline intrusions in specific locations and generating the required alarms.

5.1.3 UCA3 – Water consumption and awareness campaign

UCA3 main changes are related with the services and interface. The only service to be used for this use case is the Awareness and Behavioural Change Support Tool. It will collect data from the data manager but will not store any outcome as the other services. It will use their dedicated application to show the results to the users. Users will be identified through the connection of the dedicated app with NAIADES identity manager.

5.2 Braila

Brăila, Romania is the capital of Brăila County a city in Muntenia, eastern Romania and a port on the Danube River. The region has a moderate continental climate with great variation between summer and winter are very significant and winters frequently bring strong snowstorms. The precipitation is around 500 mm/yr. Water consumption has fallen over the last 20 years due to and the departure of industrial consumers and population loss due. This is because of migration to other countries after entering the European Union, bringing population from 234,110 in 1992 to 180,302 according to the last 2011 census. Historically Braila has relied on water from the Danube River and, to a lesser extent, on some available groundwater wells. The main source of water is the Danube River from which about 1,670 m3/h are extracted.

In the specific city of Brăila there are no major problems in what concerns the production of water. However, the city has more problems in what concerns the water distribution networks, mainly water losses. Although the city has a water loss strategy with the purpose of decreasing losses, this problem is still ongoing, and leakage detection is a priority. At the moment, average water losses amount to about 750 l/h/km, and the aim is to reduce it down to 50 l/h/k. In 2018 they had 41percent of losses/non-revenue water. They have a street network with low pressure and the network for the apartment blocks is served by station that increases the level of pressure.

Currently, to monitor warnings/events on faults (leakages, bursts) as well as unusual water consumption, CUP Braila does thorough network inspections and relies on the damages reported by dispatchers. They have historical values of consumption per sector and they monitor if this value goes above a threshold calculated on the basis of historical data, to detect a possible leakage. Most of the network is sectorised and macro-metered; however, this is not the case for all the network. In the context of the NAIADES project, they would like to monitor flow as well as pressure in the sector of Radu Negro. They prioritized this area because it is residential and has many building blocks, schools, and government institutions. At the moment, CUP Braila has managed the operation of the network entirely based on a SCADA system that displays the measurements in several points of the network. Until a couple of months ago, however, they were doing these operations without a water distribution network model. They have now one that has been developed by a University in Bucharest. The same applies for the drainage network. In addition, there is the need to know the water quality in the network, as this is unknown and many areas. However, no historical data of the treatment process is available.

In order to solve some of their problems, CUP Braila has defined two Use Cases (UCB), namely: UCB1) water consumption forecast; UCB2) Leak detection.

- UCB1, consumption aims to make water demand forecast. To this end, historical data for the district Brailita has been obtained, for a period between 2018 and 2020. Sensor installation for the district of Radu Negru is still in progress, and therefore historical data from Brailita will be used to continue with the work regardless the sensor installation for Radu Negru.

- UCB2, leakages, aims to detect and provide solutions to reduce leaks in the network. To this end, hydraulic models in combination with in-situ measurements and AI will be used.

Based on the previously defined use cases, the NAIADES components required for this pilot include mainly all NAIADES modules, excepting some AI services such as Water Quality forecasting and the Awareness and Behavioural Change Support Tool (see Figure 39Figure 38). Some of the services will only applied to specific use cases whereas others are common to all the use cases, such as the Data Fusion or the Urban Water Models services, in charge of pre-processing or generating new useful data for the other services.

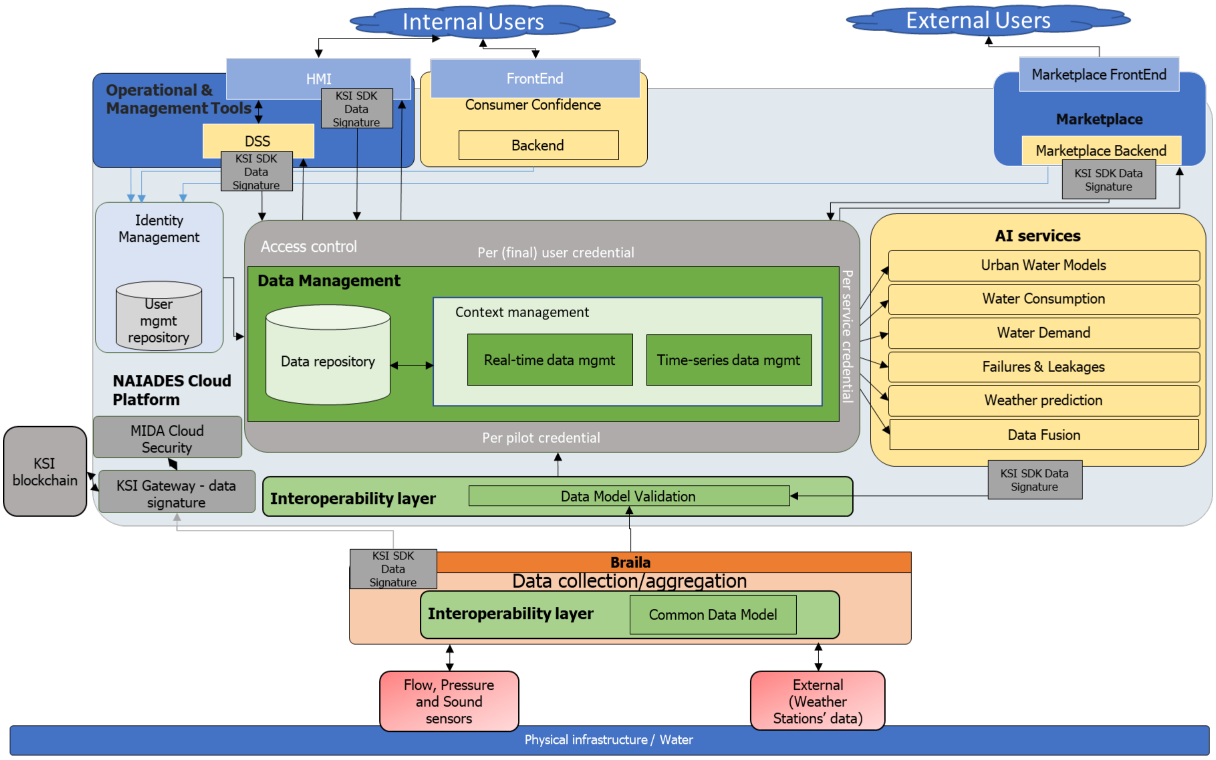

Figure 39. NAIADES platform deployment for Braila pilot.

5.2.1 UCB1 – Water Consumption Forecast

UCB1 in Braila is aimed to analyze consumption data and predict future demand in an area not digitalized yet, only the weather data can be obtained from the nearest stations (obtained from AEMET OpenData service [14]). The sensors and all the aggregation technologies will be installed by the end of the summer 2020 to collect flow, pressure and consumption levels at 4 different locations. At aggregation level, the data is to be adapted to be sent to NAIADES platform, this is, once sensors are installed and data is collected, it will be converted to the specific data models (still under definition), signed and tokenized (following the security modules and specifications).

At NAIADES platform level, the data is firstly checked to validate the data models format. Afterwards, the tokens are reviewed to authenticate the source and provide access to the services assigned to Braila user. The main services linked to this UC are Weather Prediction, Water Consumption and Water Demand AI services and the DSS. Those services will be in charge of forecasting water consumption and to extract any useful information from the water consumption data. The new data generated by the services is sent to the data manager to be shared or/and stored in NAIADES repository, again previous data model validation and data authentication.

Finally, the HMI, by demand of the identified Braila user (NAIADES internal user), will request the necessary data to the data manager to be displayed via web in the form of graphs, tables, etc.

5.2.2 UCB2 – Leakages detection

The difference with previous use case is the NAIADES services in use and the required sensors. In this case, the Weather Prediction and the Failures and Leakage detection AI services and the DSS; will be the services in use. They will analyze the pressure, consumption, noise and weather data in order to detect anomalies, check that those anomalies are not linked to normal extra consumption (temperature changes…) and create alerts that indicate possibility of leakages in one of the four measurement locations. Therefore, four pressure and noise sensors will be installed in four locations of interest.

5.3 Carouge

Carouge is a city that is part of the Geneva canton, which in itself is a financial center and hosts companies, financial institutions and industries. Carouge has a population of approximately 22,500, which has grown significantly in the last decades, from 15,036 in the1990s to 22,458 in 2018. The unemployment rate is around 4.9% and as of 2008, 37.7% of the population are resident foreign nationals. The climate of the city is moderately continental, with cold winters and warm summers. Precipitation amounts to about 1,000 mm per year, distributed over the seasons. Carouge receives its water supply (and also its public gardens' irrigation water) from the public multi-utility Services Industriels de Genève SIG. They, in turn, draw the water from Lake Geneva (90%) and deep wells (10%).

As a response to a strong citizen demand to have more nature landscapes and biodiversity in the city, and to support the development of ecological gardening practices accessible to all, the municipality, in partnership with the group of environmental activists, created 180 garden boxes at different points in the city. The city invites residents to plant and use the gardens. As a result, everyone can grow and harvest freely and without registration the plants and vegetables of their choice, in an ecological way. All expenses for the boxes are taken from taxes.

The City of Carouge provides the water in trucks with cisterns. Workers from the city go to each of the 180 vegetables/flowers, boxes, and water them manually. The water comes from the multi-utility company SIG. The current problem is that the boxes must be watered 1 or 2 times per week, and even more frequently if the weather is warm. This activity is not only time consuming for the city workers, but also water inefficient. On the one hand, the rapid grow of the city makes pressure on budget to hire more people, so the load of work for city employees is bigger and bigger, and on the other hand more groundwater is used during drought periods". However, the municipality is proud of the garden boxes program and wants to continue expanding it since it is highly regarded by the population. For this the city needs technological solutions that can help same money and time.

Another water-related problem that the municipality is facing is that it has two water fountains people get contact to, in summer, and the municipality is responsible for the water quality in them, as they can become a public health issue. For this reason, pH, bacteria, and chlorine are monitored. In particular, Chlorine must be frequently monitored as the State became very strict on this topic. One of the fountains had to close in the past due to the presence of bacteria. It is stressed that the most popular one, the Fontaine des Tours, has a capacity of 150,000 l and it consumes 2 million l per year, which brings questions about its sustainability.

Based on the above description, two Use Cases in Carouge (UCC) are to be addressed in the NAIADES project, namely UCC1 – Watering, and UCC2 – Fountains, by means of technology. Sensors will be placed;

smartphone applications will be developed, and weather information will be used to improve decision making.

- The objective of UCC1 is to reduce the amount of water and the time of the employees in charge of the garden.
- The objective of UCC2 is to improve the information and availability of the water quality of the fountains, thus to assure water quality on fountains and plan beforehand any fountain maintenance related with water quality so to maintain the fountains operative for the users.

Based on the previously defined use cases, the NAIADES components required for this pilot include mainly all NAIADES modules, excepting some AI services such as Water Treatments or Urban Water Models (see Figure 40Figure 38). Some of the services will only be applied to specific use cases whereas others are common to all the use cases, such as the Data Fusion, in charge of pre-processing or generating new useful data for the other services.

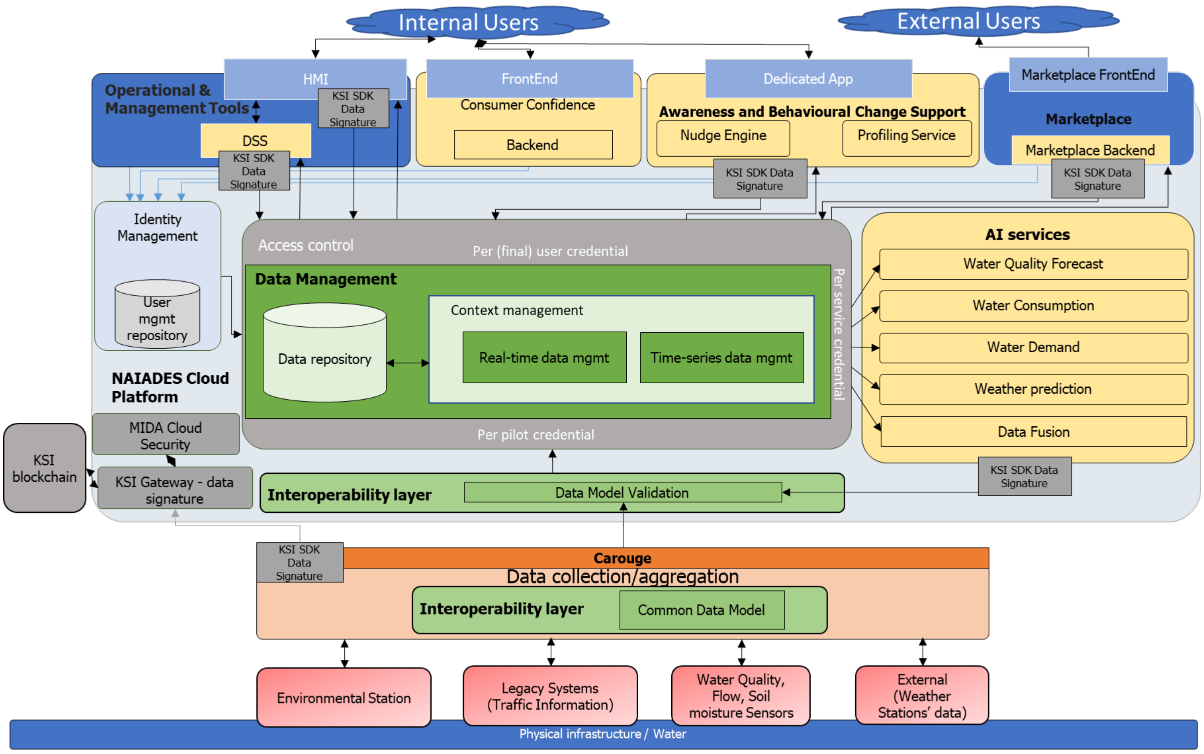

Figure 40. NAIADES platform deployment for Carouge pilot.

5.3.1 UCC1 – Watering

UCC1 in Carouge is aimed to analyze water requirements from plant boxes and provide suggestions about the optimal amount of water required. The digitalization for this use case will be done by the end of summer 2020 in 8 different locations (plant boxes), only the weather data will be start being collected from the nearest stations (obtained from AEMET OpenData service [15]). The digitalization will include measurements of soil moisture, water flow sensor (solution still under definition) and a dedicated environmental station. At aggregation level, the data is to be adapted to be sent to NAIADES platform, this is, once sensors are installed and data is collected, it will be converted to the specific data models (still under definition), signed and tokenized (following the security modules and specifications).At NAIADES platform level, the data is firstly checked to validate the data models format. Afterwards, the tokens are reviewed to authenticate the source and provide access to the services assigned to Carouge user.

The main services linked to this UC are Weather Prediction, Water Consumption and Water Demand AI services and the Awareness and Behavioural Change Support Tool. Nevertheless, the functionality of the services for consumption and demand will be slightly different from the other pilots since the water demand objective is different. The Awareness and Behavioural Change Support Tool will be in charge of analysing the results from the demand and consumption services in order to generate useful information for the city staff (watering schedules, amount and route). The new data generated by the AI services is sent to the data manager to be shared or/and stored in NAIADES repository, again previous data model validation and data authentication. The data generated by the Awareness and Behavioural Change Support Tool will not be sent to the data manager, it will be directly presented in its dedicated application in the form of graphs, tables, alerts, etc. On the other hand, it will allow users to update information in the platform related to the plant boxes.

5.3.2 UCC2 – Fountains

Fountains use case in Carouge is focused on the forecast of water quality to assure water quality and prevent and/or plan future maintenance activities. The main differences with previous use case are in the sensors and the services and presentation layer. Water quality sensors such as chlorine and pH sensors will be installed in the drainage of one fountain in Carouge. This use case will make use of the Water Quality Prediction AI service and the DSS. The AI service will forecast future values of the parameters captured by the sensors and the DSS will use quality thresholds on the current and future values to trigger alarms through the HMI.

5.4 Lab scale Water Treatment Plant (lsWTP)

This pilot appeared as a necessity to demonstrate Water Quality Treatments services. Due to the difficulties to access real treatments data from WTP and the impossibility to modify the treatments to validate the services; a lab scale WTP was decided to be created based on Braila WTP processes, so to generate data to create Treatments models and to demonstrate services results.

The main objective of this pilot is to demonstrate the AI service is able to suggest the optimal treatments parameters in order to obtain the best quality treated water for any incoming water quality.

Based on the above objective, one Use Case will be demonstrated in the lab scale WTP (UClsWTP).

- UClsWTP. This use case will provide suggestions about the best treatments to be applied for any water quality in the inlet of the WTP and, supported by the DSS, it will trigger alarms when notable changes between the current and the suggested treatments are detected. Furthermore, the same procedure will be applied by using forecasted values of quality parameters (provided by the Water Quality Forecasting AI service), so to provide the treatments and alerts beforehand so the WTP is able to plan the required actions in advance.

Based on the previously defined use cases, the NAIADES components required for this pilot include mainly all NAIADES modules, excepting some AI services such as Water Consumption and Water Demand (see Figure 41Figure 38).

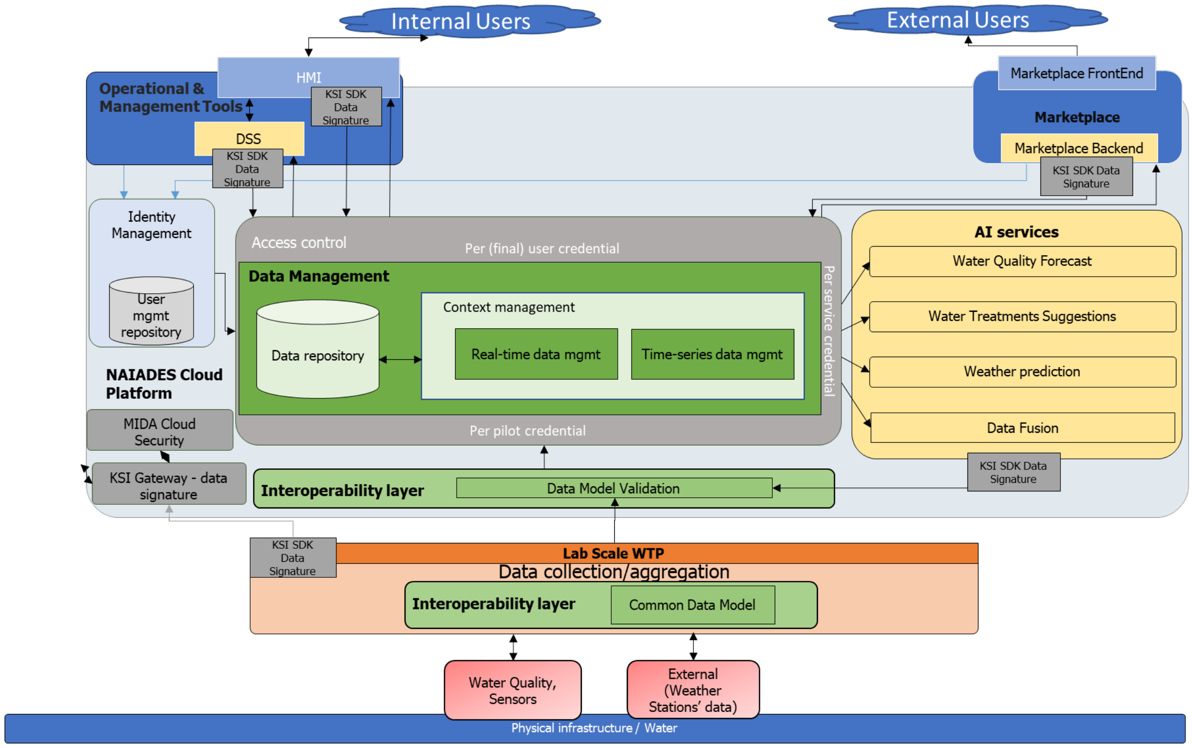

Figure 41. NAIADES platform deployment for lab scale Water Treatment Plant pilot.

5.4.1 UClsWTP1 – Water Treatments prediction and forecasting

UClsWTP will require the creation of a lab scale water treatment plant, whose processes will be based on an existent WTP from Braila. The input of the lab scale WTP will be sensorized so to measure quality parameters such as pH and Chloride. Additionally, the current treatments values on the WTP will be shared and the weather information will be collected since it will affect the water quality input. This use case will use the AEMET OpenData service [13] to obtain the weather data. At aggregation level, the data is to be adapted to be sent to NAIADES platform, this is, once sensors are installed and data is collected, it will be converted to the specific data models (still under definition), signed and tokenized (following the security modules and specifications). At NAIADES platform level, the data is firstly checked to validate the data models' format. Afterwards, the tokens are reviewed to authenticate the source and provide access to the services assigned to lab scale pilot.

The main services linked to this UC are Weather Prediction, Water Quality Forecasting and Water Treatments modelling AI services and the DSS. The data sent by the aggregator, current water quality parameters is to be used by the AI services (treatments and forecasted water quality). These services will generate new data that is sent to the data manager to be shared or/and stored in NAIADES repository, again previous data model validation and data authentication. The data generated by the Water Quality Forecasting AI service is shared again so the Water Treatment Suggestions analyse it to generate forecasts of future treatments. Both outputs from the Treatments module, current and future treatment parameter values, are to be shared by the data manager so the DSS and the HMI can access them. The DSS will generate alerts when the difference between the current values of the treatments and the suggestions are very different; and the HMI will display all the water parameters monitored and forecasted; and suggested treatments. The main idea for this use case is that any internal user to have access to the results, with special interest on Braila pilot users.

6 NAIADES architecture heterogeneity and benefits for end users

The NAIADES architecture was designed taking into account the requirements of water utilities/water managers (D2.5), in particular the NAIADES end users' necessities (D2.7) and also the capabilities of the NAIADES technological partners and the state of the art reference architectures: RAMI, IIRA, FIWARE, presented in Section 2. The objective of NAIADES is to provide a platform for water utilities/managers which encompasses different water management services and takes into account the combination of all previous requirements to then allow to make better, informed decisions with this broad support. Therefore, the architecture main efforts were focused on the concepts that are the basis of most state-of-the-art IoT platforms:

- Interoperability to facilitate information exchange, adoption of standards and data sharing across services. NAIADES adopted compatibility with FIWARE as a result of the interest shown by the EC to foster this framework for the water sector. It was not originally NAIADES objective to work on the improvement of FIWARE capabilities, but to use it as a tool. FIWARE compatibility allowed us to join the DW2020 group where we collaborate with other European water related projects in different task forces such as the FIWARE data models, where specific data models for the water sector are being designed, including water distribution network infrastructure and water quality and GIS data models (https://github.com/smart-data-models/SmartWater ; https://github.com/smart-data-models/dataModel.SatelliteImagery). Although these data models are not required by the current solutions, the NAIADES platform is flexible enough to include them once they are completed. The NAIADES modules in charge of the interoperability will be updated to accept the new data models into the platform - Data Models Validation (4.5.5, and D3.1), to translate end users data - Common Data Models (4.5.3, and D3.1) and to create the entities required for managing the new data models - Context Data Manager (4.5.6, and D3.9). The update of these three modules will be enough to allow the adaptation of the platform to new standards. More details about how NAIADES data models handle different types of data and the tools to adapt any data to NAIADES platform can be found in D3.1.
- Modularity (and scalability) to facilitate the inclusion, update and removal of services. The NAIADES platform allows the easily integration of any new service into the platform. The interoperability related modules allow any service to easily work with any type of end users' data; thus, it is possible to integrate any service that processes that information. This easy integration and interconnection between services allows the creation of a pipeline "on the fly" and also adding or removing any services to the platform to fulfil the evolving needs of the water utilities. This and the integrated manner in how we leverage all this, is the key benefit and our approach in how we apply the modularization. Each module in the architecture is focused on a specific task/service whose results can be used as outcome or as input of other services. These modules can be updated or replaced by novel services without affecting the platform. The NAIADES cloud infrastructure and the use of container technologies – in our case Docker – makes it very easy for the platform manager to scale the system if the need arises by just deploying additional container on the cloud and thereby scaling it up. By the end of the project, all services will be presented in the NAIADES marketplace (D7.5). It will present to external users (including service providers) the current NAIADES services and combined solutions so they can propose their solution as services for the platform.
- Security to provide a secure platform that permits the trustable sharing of information among sensors and other type of data management services. NAIADES includes data signature (based on Blockchain technology) to ensure data integrity (4.5.22); and authentication (4.5.8) to enable

the different services managing data to trust data origin and its contents; and to grant access only to those allowed. More information is provided in D7.6.

The NAIADES services included in the platform and architecture were established at proposal level (water demand and quality forecasts, failure and leakage detection, etc.). They were adjusted to the NAIADES end users' requirements since they are the real use cases where the services will be validated. The NAIADES services fulfil end users' necessities by smartening the water sector (see D2.5), this is, by implementing state of the art AI solutions which are not currently widely applied on the water utilities. The NAIADES use cases definition during this first period have mainly focused on guaranteeing the use cases integration and their application in the NAIADES end users.

The services' functionalities and validation are based on NAIADES end users' needs; thus, they process the data as expected by the end users. This does not mean that other SCADA, GIS and Asset Management solutions cannot be supported by the NAIADES platform. The current version of the architecture, FIWARE-compatible, will allow to operate any data provided by the end users (see D3.1). Currently, regarding spatial information, all the FIWARE data models used in the NAIADES project contain a mandatory attribute "location" to identify the current position of an entity. At the same time, all the received data in the Orion context broker have attributes of the type DateTime. So, it is possible to determine the location, the date and the time of each generated data. For example, to solve a watering management problem where a truck is watering various flowerbed installed in Carouge, the data about the current location of the truck and the static position of the flowerbeds permit to determine when and which flowerbed is watered by the truck. A query based on the FIWARE localization properties is made to the Orion context broker without any other data processing. So, it is easy to get the required information to link the truck and a given flowerbed in terms of spatial-temporal aspects. The user interfaces are also using the location info provided with the data models to use it with 'google-maps-react' to show the locations that are triggering alerts in a map; and other spatial data usage can be found in the specific services (e.g., D6.2 – sections 3.3, 4.4.5 and 4.4.6). When it comes to asset management, it is possible by creating FIWARE entities per asset; FIWARE provides enough data models to deal with most assets and the missing water sector related ones are in progressive definition and development by DW2020 (https://github.com/smart-data-models/SmartWater). The specific asset management solutions will require the integration of dedicated services for this purpose. Currently, none was required by NAIADES' end users, but NAIADES architecture modularity and interoperability will allow their easy integration. The NAIADES use cases will continue their evolution during the second part of the project. The application of NGSI-ld was planned for this period and although semantic functionalities have already been included in the watering use case at Carouge, semantic knowledge benefits will also be evaluated in the other use cases and DSS functionalities. Furthermore, other suggestion from the PO, reviewers and end user requirements will also be considered.

The NAIADES architecture design solves both common and key requirements from most end users, related to data management, IoT infrastructure and security. Further requirements related to topics such as smart solutions, etc. are not directly covered by the architecture. These are addressed in the different services of the NAIADES solution. The architecture eases their integration in a common platform where the user can access all of them and thereby allows them to leverage the maximal benefit in their interplay together in one platform. Furthermore, the architecture is flexible enough to adapt to new requirements by upgrading the required modules.

On the whole, the NAIADES architecture will help the water utilities to easily access services to speed up decision-making due to efficient data analysis and processing (Process excellence). The access to the NAIADES platform will be easy and its centralized approach will allow it to ensure that all components are always updated and provide the opportunities of the latest technologies (Brand and Innovation). It will be a European level platform to be reached by water utilities from different countries and it will thereby elevate utility brand and engagement in the water industry by providing this central point to cooperate and exchange knowledge. Finally, by allowing the ease of integration of further (AI) services, it provides a leverage for the utilisation of many services for all participating water utility. More information about the link between NAIADES solutions and the needs of the water urban sector can be found in D2.5.

7 Conclusions

The deliverable D2.9 gathers all the specifications about NAIADES platform architecture still under definition in T2.6. This first version of the architecture has taken into consideration the first outcomes from T2.4 and T2.5 being possible to establish the grounds of the architecture, nevertheless, it will continue evolving and adapting to future issues detected during the validation of the first prototype. This document will be updated and the final version will be delivered as D2.10.

Two main strategies were followed for the design of the architecture: a top-down approach and FIWARE compliance. A top-down approach was used, starting from a general black-box architecture adequate to solve end users' problems with service providers technologies, following by a detailed definition and specification of every component. Furthermore, it was settled as key point the architecture to be modular and FIWARE compliant to assure adaptability and facilitate collaborations and interconnections to other H2020 European initiatives.

References

[1] J. K. C. &. S. C. Frysak, «Benefits and pitfalls applying RAMI4. 0,» *IEEE Industrial Cyber-Physical Systems (ICPS),* pp. 32-37, 2018.

[2] S. W. M. B. D. J. J. R. D. P. C. A. .. &. K. A. Lin, «Industrial internet reference architecture.,» *Industrial Internet Consortium (IIC), Tech. Rep.,* 2015.

[3] F. Foundation, «FIWARE architecture,» [En línea]. Available: http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Architecture. [Último acceso: 13 05 2020].

[4] SIG, «SIG smart city,» [En línea]. Available: https://ww2.sig-ge.ch/a-propos-de-sig/nos-engagements/smart-city. [Último acceso: 17 06 2020].

[5] FIWARE, «FIWARE Environment Data Model,» [En línea]. Available: https://fiware-datamodels.readthedocs.io/en/latest/Environment/doc/introduction/index.html . [Último acceso: 26 11 2020].

[6] FIWARE, «Fiware Weather Observed Data Model,» [En línea]. Available: https://fiware-datamodels.readthedocs.io/en/latest/Weather/WeatherObserved/doc/spec/index.html . [Último acceso: 26 11 2020].

[7] FIWARE, «FIWARE Parks and Gardens Data Models,» [En línea]. Available: https://fiware-datamodels.readthedocs.io/en/latest/ParksAndGardens/doc/introduction/index.html .

[8] G. B. S. S. S. &. A. S. M. Bora, «OSI reference model: An overview,» *International Journal of Computer Trends and Technology (IJCTT),* vol. 7, nº 4, pp. 214-218, 2014.

[9] M. &. J. P. Kircher, Pattern-oriented software architecture, patterns for resource management, John Wiley & Sons, 2013.

[10] E. B. F. M. V. M. &. M. M. Fernandez, «The secure three-tier architecture pattern,» *International Conference on Complex, Intelligent and Software Intensive Systems,* pp. 555-560, 2008.

[11] A. K. R. L. H. R. N. D. G. Ahto Buldas, «Keyless Signatures' Infrastructure: How to Build Global Distributed Hash-Trees,» de . *Secure IT Systems - 18th Nordic Conference, NordSec 2013, Proceedings. LNCS 8208, Springer*, 2013.

[12] A. T. R. L. R. G. K. B. S. F.-H. Ahto Buldas, «Efficient Record-Level Keyless Signatures for Audit Logs,» de *Secure IT Systems - 19th Nordic Conference, NordSec 2014, Proceedings. LNCS 8788, Springer*, 2014.

[13] AEMET, «AEMET OpenData,» [En línea]. Available: https://opendata.aemet.es/centrodedescargas/inicio. [Último acceso: 27 05 2020].

[14] A. N. d. meteorologie, «MeteoRomania,» [En línea]. Available: http://www.meteoromania.ro/. [Último acceso: 27 05 2020].

[15] Prevision-meteo, «Previsionmeteo,» [En línea]. Available: https://www.prevision-meteo.ch/. [Último acceso: 27 05 2020].