PROJECT DELIVERABLE REPORT

Greening the economy in line with the sustainable development goals

D3.6 - NAIADES Data Fusion Middleware - midterm

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	01/06/20	**Duration**	36 months
Project URL	www.naiades-project.eu		
EU Project Officer	Alexandre VACHER		
Project Coordinator	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH		
Deliverable	D3.6 - NAIADES Data Fusion Middleware: midterm		
Work Package	WP3 - Data and Sensors Infrastructure		
Date of Delivery	**Contractual**	M16	**Actual** M16
Nature	R – Report	**Dissemination Level**	PU-PUBLIC
Lead Beneficiary	ADSYS		
Responsible Author	Manuel Fernández, Manuel Ramiro	Email	manuel.fernandez@advanticys.com manuel.ramiro@advanticsys.com
		Phone	
Reviewer(s):	Sergio Montero (IBA), Cédric Crettaz (MI)		
Keywords	Data Fusion, FIWARE, AI modules, Context Data Management, IoT Platform, Data Model Validation		

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
0	26/06/2020	M. Ramiro, M. Fernández	ToC definition
0.1	16/07/2020	M. Fernández, M. Ramiro,	Document update #1
0.2	10/08/2020	M. Fernández, M. Ramiro	Document update #2
0.3	09/09/2020	M. Fernández	Document update #3
0.4	15/09/2020	M. Fernandez, M. Ramiro	Document update #4 Version ready for internal review

0.5	18/09/2020	C. Crettaz	Internal review
0.6	21/09/2020	M. Ramiro	MI, IBA, AIMEN review comments addressed
1.0	28/09/2020	M. Ramiro	Final version
Revised	02/06/2021	M. Ramiro	RP1 review recommendations included
1.1.1	09/06/2021	M. Fernández	Internal feedback
1.1.2	21/06/2021	M. Ramiro	Additions to **Summary & Section 5.2** addressing the RP1 review comments
	25/06/2021 28/06/2021	Juan Fernández C. Crettaz	Internal review AIMEN, UDGA
1.1.3	30/06/2021	M. Ramiro	Comments addressed
1.2	07/07/2021	ADSYS	Revised version - v1.2

Contents

Abbreviations

AI	Artificial Intelligent
API	Application programming interface
CDM	Context data management
DCA	Data collection and aggregation
DFM	Data Fusion Middleware
DMV	Data Model Validation
DSS	Decision Support System
HMI	Human-Machine Interface
IoT	Internet of Things
NGSI	Next Generation Service Interface
REST	REpresentational State Transfer
SDK	Software Development Kit
UI	User Interface

1 Summary

This deliverable summarizes the results from T3.3 as a mid-term report on the Data Fusion module to be developed as part of NAIADES architecture. The document is organized in five (5) sections, section 2 where an introduction to the context and scope of NAIADES architecture is provided, section 3 and section 4, where a detailed description of the technical approach adopted to design and implement the Data Fusion Middleware (DFM) is presented and finally section 5, which relates to the progress achieved so far (up to M16) in the development of DFM v1. The content provided in this document is strongly related to T2.6 and the corresponding deliverable (D2.9) where the overall system design of NAIADES platform, and all its components is reported. Furthermore, two other tasks, T3.1 (Data Harmonization Framework & tools) and T3.4 (NAIADES IoT Framework) and the corresponding deliverables (D3.1 and D3.9) should be taken into consideration for a fully comprehensible understanding of the information presented in D3.3.

The first version of this document (mid-term) is due by the end of M16 (this version will formalize the concept, design approach and initial version of DFM (v1)) as specified in MS3 (Data and sensors infrastructure). This document will be updated so to provide the final report (D3.7) due by M28, which will provide all the information related to the final version of the DFM, including descriptions of software, APIs, and cloud deployment instructions. In order to avoid misunderstandings, it is necessary to clarify that NAIADES DFM is not based on any specific FIWARE component but provide FIWARE compatibility by means of shared Data Models (JSON v2/LD format) and standard APIs within the scope of NAIADES architecture (see D2.9 NAIADES architecture for further details) NAIADES DFM has been conceived as part of NAIADES platform (dedicated data-preprocessing module) but can be leveraged by other FIWARE-compatible platforms and extended to multiple domains (other than water management)

2 Introduction

2.1 Context and scope

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 providers, data consumers and marketplace users are presented here 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., water operators, city council). However, this project is heavily dependent on AI services, therefore large amount of data is required for the development of these services, thus 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, those technical partners will design, deploy physical sensors and develop a platform for the insertion of data from the sensors to the system. As part of the following step, a Data Fusion module will be developed to fuse all the data that originate from different heterogeneous sources (IT legacy systems, sensor networks, external web services, platform modules) and differ in many aspects and create a homogenous database that will be easier to process. Next step will imply to identify or define appropriate FIWARE data models to the data from all sources to be stored in the system's database. Since the system's database will also exchange (provide and receive) data with all the components, the system will provide the needed tools (i.e., APIs, data signature methods), Context Data Management can be considered the core component of the IoT Framework, interconnected to the platform through the communication infrastructure (Cloud platform).

Third level involves the data consumers, in which the partners "consume" directly data stored in the database without any processing that is connected to a service. In this category are included the partners responsible for the development of the application layer, in turn the responsible to develop the UI of the decision support tool (DSS), and those responsible for other HMI like the consumers' awareness and behavioural tools.

Fourth level is focused on 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, integration of cloud services and the development of the marketplace, the overall security mechanisms that will run throughout the system, the development of urban models, to the user recommendation tool. Regarding the AI services, the system will support water demand predictions, the weather prediction module, the water consumption prediction, failure and leakage prediction and the consumer confidence tool, and also will develop the water quality monitoring, and the dynamic treatment and quality prediction module.

Fifth level regards the consumers of NAIADES services, those which use the outcomes of the services developed (i.e., AI output, raw data collection, data fusion, …) either directly or indirectly. On the other hand, end-users can be considered as those who use the outcomes of these services via the DSS therefore directly from a service, and as such are considered service consumers.

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 services they desire.

3 NAIADES Data Fusion Module (DFM)

3.1 Challenge and Objectives of DFM

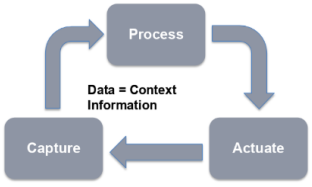
Based on task T3.1 (Data Harmonization Framework and Tools) results, the aim of T3.3 is to develop data fusion capabilities.

For such purpose, we will define a **data fusion module** that provides the capability of *analysing*, *cleaning* and *combining* data from different sources. The results of a specific data fusion operation will be a separate entity in the system, and this will enable them to be used by other components of the application.

One major challenge for the IoT platform is to handle vast amount of data generated from the smart devices that are resource-limited and subject to missing data due to link or node failures. DFM is especially suited to address this concern, as erroneous data can be filtered, aggregated in pre-configured periods and the missing gaps in the time series can be interpolated, if such is the need. The results of these operations are more reliable and consistent data sets, which may be mandatory in order to perform more complex analysis by other components.

The consortium has decided to adapt a FIWARE-based architecture, linked data models and corresponding context services, in line with other H2020 water projects (H2020 SC5-11-2018). Based on this decision, NAIADES IoT framework takes as core component, FIWARE Orion Context Broker for the context management, hence the DFM interaction with the IoT framework will be mainly supported by this component through the correspondent API, in particular making use of the time series methods to query the server for the stored data from the physical entities, as well as the store methods to save the results from the process itself.

3.2 Context of D3.6 within the NAIADES Technical Architecture



FIWARE is an open source initiative defining a universal set of standards for context data management which facilitate the development of Smart Solutions for different domains such as Smart Cities, Smart Industry, Smart Agrifood, and Smart Energy. In any smart solution there is a need to gather and manage context information, processing that information and informing external actors, enabling them to actuate and therefore alter or enrich the current context. The FIWARE Context Broker component is the core component of any "Powered by FIWARE" platform. It enables the system to perform updates and access to the current state of context. The Context Broker in turn is surrounded by a suite of additional platform components, which may be supplying context data (from diverse sources such as a CRM system, social networks, mobile apps or IoT sensors for example), supporting processing, analysis and visualization of data or bringing support to data access control, publication or monetization.

The Data Fusion Module (DFM) tries to bridge the gap that exists between the sensor data that arrives directly from the pilots' Data Collection and Aggregation Modules (DCA) and the actual data that may be needed by the high-level modules of the architecture.

The DFM will periodically retrieve data sets from the CDM's database, combine and clean the data, and the result obtained will be stored in the database to be used by other modules. Each of the data fusion processes will be separate entities of the system, so their data output must be compliant with the data model that the data model validation module (DMV) oversees.

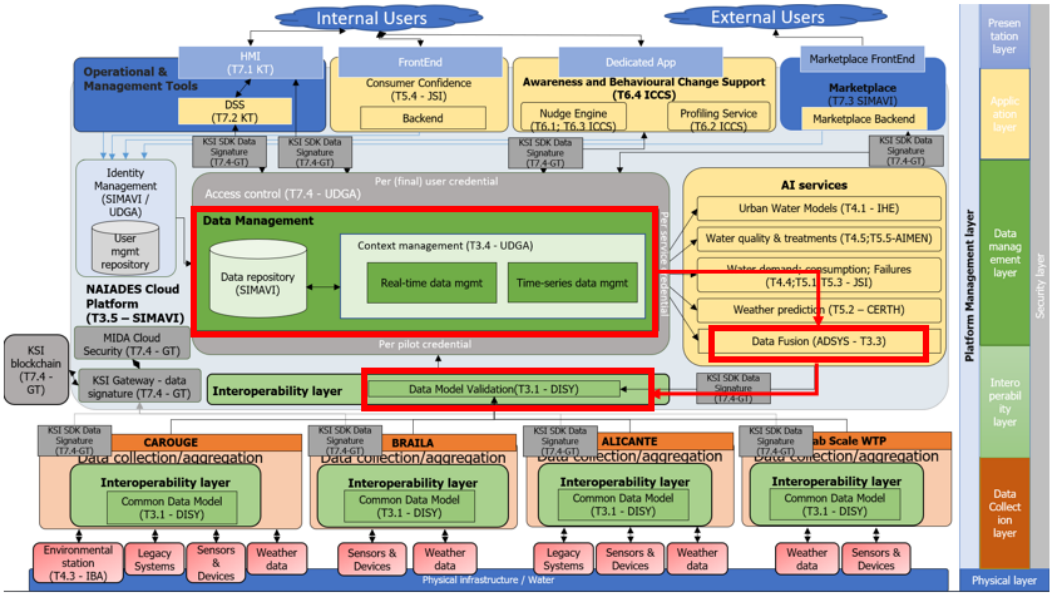


Figure 1. DF module within NAIADES overall architecture

3.3 NAIADES DFM integration approach in Data Management layer

As illustrated in Figure 1, the Data Fusion module has been included under the **AI services** packages this is mainly since from an architectural approach its behaviour is similar to any other AI module.

The DFM "consumes" data from the IoT Framework (through the CDM) and "produces" processed data that are feedback again into the IoT Framework (CDM), in between a verification of the output generated (fused data) is done through the Data model Validation Module to ensure format compatibility. The output generated, which we can call **fused data**, will be stored in the platform's data base via the Context Broker, using the appropriate API methods. Once this output is stored, it can in turn be consumed by any other module in the platform whether is used as an input to feed the AI modules or as an output to be visualized by the platform HMI or any dedicated App.

Figure 2 depicts a detail of NAIADES IoT platform services and communication flow at the level of the Data Management layer.

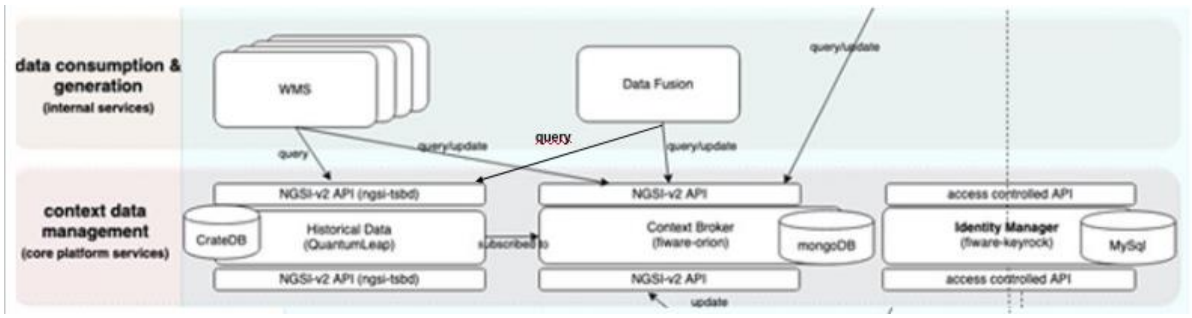


Figure 2 Detail of IoT Framework and communication flow (from D3.9)

3.3.1 Context Data Management module

Context manager component is created leveraging on the open-source FIWARE component called Orion Context Broker. The Orion Context Broker is an implementation of the Publish/Subscribe Context Broker

generic enabler of FIWARE, providing an NGSI interface. Using this interface, clients can do several operations:

- Query context information. The Orion Context Broker stores context information updated from applications, so queries are resolved based on that information. Context information consists on entities (e.g. a water fountain) and their attributes (e.g. water temperature or location of the fountain).
- Update context information, e.g. send updates of temperature
- Get notified when changes on context information take place (e.g. the temperature has changed)
- Register context provider applications, e.g. the IoT sensor provider for the temperature installed in the fountain

Contexts are used to describe observed data (e.g. a temperature coming from IoT sensors), but also data inputted by users through HMI (a threshold for temperature triggering an alarm, e.g. describing that risk is high for bacteria development on a fountain), and also data generated by AI modules which will estimate a certain value (e.g. a recommendation for water irrigation for plants on a certain park). More information on how to use this interface (API) is included in the following sections.

3.3.2 AI Modules

These modules implement 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 decisions for the current scenario. AI models are be trained using historical data.

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

At functional level these modules:

- Provide time series of a list of related parameters, the service will provide as output a prediction every certain time (service dependent).
- Provide time series of a future forecast of the service-related parameters, the service will provide as output a future prediction (alerts, dosages, times, etc.).
- 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.

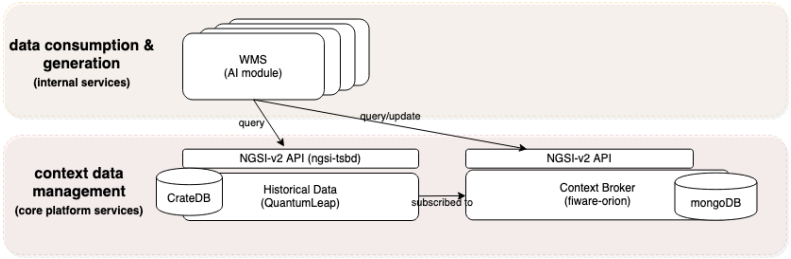


Figure 3 Context data management for Historical data (detail from D3.9)

Context manager is used to store and manage entities describing current values, meaning no historical data is kept. Keeping historical data is a requirement requested by several component of the NAIADES components, e.g. AI modules require high amounts of data to generate their models, just as HMI needs to visualize trends for certain IoT sensed data. For this purpose, it is introduced the usage of the Historical

data component. The component, similarly to the context broker exposes an API which only enables consumption (reading) of data sets.

3.3.3 Data Model Validation module

The purpose of this model is the check of all incoming data regarding the validity of their meta data and data model. As the existing ORION context broker component 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 Validation component. 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 check the validity of the data model received 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 the Common Data models to be adopted (FIWARE existing data models) or to be developed (FIWARE-compliant), TALEND is regarded as favorite, but other options are in the process of being evaluated, e.g. a specific Python program.

3.3.4 Data Repository

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

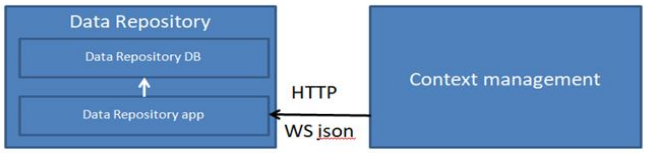


Figure 4 Data Repository & Context Management

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

- Data Repository database: The Data Repository database is a relational database, 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 on the behalf of the other NAIADES components. It exposes web services to the NAIADES Data Management, using HTTP protocol, so to persist or access data from the database. The Data Repository backend contains a model DTO (Data Object Transfer) 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 and will check Data Signature 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.

3.4 NAIADES DFM component specification (v1)

3.4.1 Overall Description

As already described, the Data Fusion Module is a software component in charge of periodically combining, aggregating and cleaning data stored from the different entities, being the generated output a new separate entity.

As such, inside the DFM different processes run at independent periods to perform such actions. These are called **Data Fusion Processes**.

For each Data Fusion Process defined, it will be necessary to configure the following:

- **Execution period:** the period that will indicate the start of the data fusion process. Essentially it will be a cron style string which will enable the precise execution of the process.
- **Input signals:** the entities whose historic values will be retrieved on each execution.
- **Default start date:** default date on which to start retrieving data for the fusion process. If the last execution date is more recent, then that will be used.
- **Pre-process:** any cleaning or aggregation action on the raw readings is done here.
- **Process:** the actual data fusion process performed with the readings recovered.
- **Post-process:** Last step actions on the data processed, if necessary.

The Data Fusion Process is a **pipeline of successive stages**, where different actions are performed. After each firing, which is marked by the configured **execution period**, data from the required entities (the input signals) will be retrieved from the default start date or the last insert date, whichever is more recent. A **pre-process** will be done on the raw signal data, such as any aggregation of the data on set periods of the day, or data cleaning of undesired readings. Once this is done, the resulting readings are passed to the **process** itself, where combination of different signals is done at signal reading level, that means, on readings that share the same date. The process is the most complex stage, and it will resemble a flow diagram on which filtering on the reading will be performed if needed, both at signal reading value and date levels, isolated mathematical operations, etc.

The actual stored process configuration will be in the form of a JSON object, but to clarify the following flow diagram is an example of what will be basically what will be done in this stage:

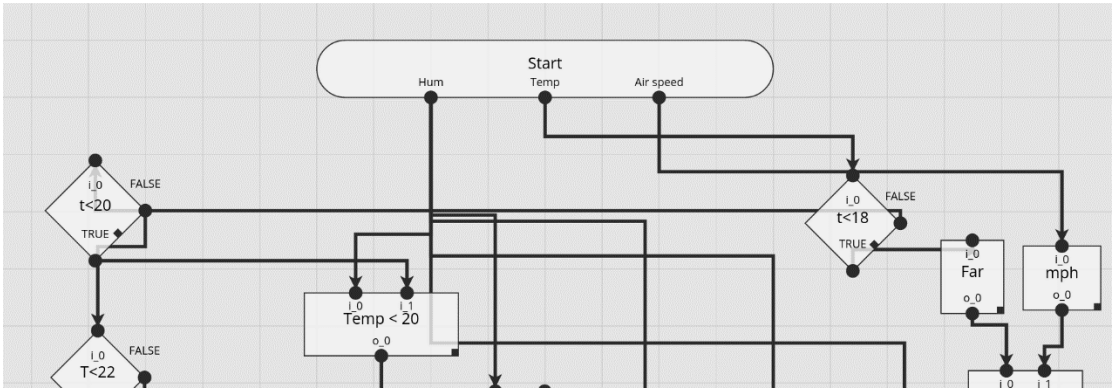


Figure 5. Example fusion process

Finally, a **post process** is done on the output of the fusion process itself, on which additional actions are performed on the data set. Examples of these actions are keeping only the average values, the maximum or minimum, interpolate missing data, etc.

3.4.2 Inputs and Outputs

For DFM, it is paramount to handle data sets to perform the adequate fusion operation, as in many cases it is not a real time combination of signals what is required but rather an output based on an historic set.

Because of that, **inputs** of the DFM are going to be the data set of the different entities that are stored in the database. Therefore, the data is gathered from the Context Data Management (CDM) layer in the form of API requests, on which a specific range of signal readings is retrieved between two given dates, depending on the actual configuration of each individual data fusion process. This data will mainly belong to the **DCAs from the pilot sites,** but it could be the case that the data used belong to another **high-level AI module**, or even another data fusion process. If it belongs to a valid entity of the system, it is feasible to be used as input.

On the other hand, the **output** of the DFM, or fused data, will be a reading with value and date information, that will have to belong to a separate valid entity of the platform. As such, it will be inserted in the database by making use of the same API of the CDM. Obviously, as any valid message exchange within the platform, the output will have to adopt the existing data model to be sure that the message is not rejected by the Data Model Validation module.

3.4.3 Main functionalities

The main functionalities achievable by the DFM are basically provided by the stage that takes place at each moment:

- **Functionalities of Pre-Process:**
  The pre-process is the first stage of the data fusion. On this stage, from the gathered readings of the selected signals it is possible to individually aggregate them on certain pre-fixed periods of time. This allows to turn cumulative signals into incremental, as it is the case of energy meters where the data is always growing, but also to get the average on that aggregation period, or the maximum, or the last value. This ensures that the data set of each of the signals involved is ready for the next stage.
- **Functionalities of Process:**
  The process is the fusion algorithm itself, on which the readings from the previous stage from all the signals involved are used together. On this stage, the operations are performed on signal reading level, so to be able to combine two signals together the date of their respective readings must be equal. In this stage, it is possible to do mathematical operations using one or more signals together. It is also possible to filter readings depending not only on their signal value but also according to their date or time.
- **Functionalities of Post-process:**
  The post-process is the third and last stage of the data fusion, and it involves performing actions on the signal readings that have "survived" up to that point. It is often used to obtain from the result data set a single value, the average on a certain period, the maximum, or the minimum.

3.4.4 Software requirements

Regarding software requirements, the DFM will be a Java program, and as such it will require to have a JVM installed. The configurations and status of the module's processes will be stored in a SQL database, so an installed MySQL or MariaDB server is adequate.

The programs can be launched as a system service in a Linux environment, or *dockerized* in a container if that is the desired approach.

3.4.5 Hardware requirements

Any Linux environment is more than capable of running the DFM. For starters, we assume 4GB of available system RAM, but obviously should the amount of concurrent data fusion processes grow in number then it will have to be increased accordingly. In the case of processing power, it is also hard to estimate, so to begin with, we assume a dual core configuration is available. This will ensure that the parallel processes run without excessive delays.

4 Activities Performed during the period

4.1 DFM v1 prototype design

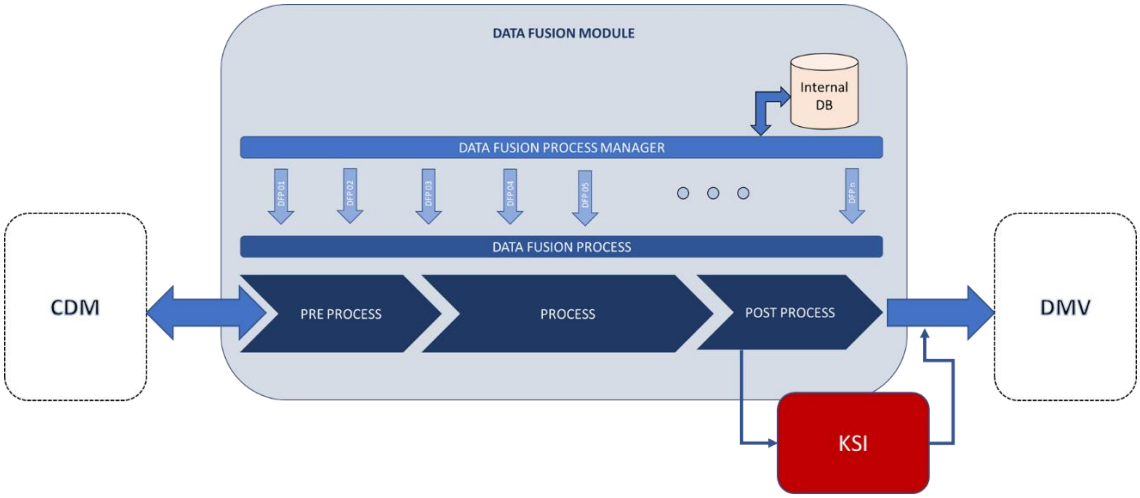


Figure 6. DFM Subcomponents

The DFM has the following subcomponents:

- **Internal database:** Storage component in charge of keeping the active data fusion process configurations.
- **Data fusion process manager:** In charge of retrieving the active configurations and launching the concurrent data fusion processes at the specified time periods.
- **Data fusion process:** Each of the individual data fusion configuration will generate an independent process thanks to the process manager. These threads will oversee the fusion actions themselves, from retrieving the data from the CDM API to the execution of the consecutive stages to the insertion of data conforming to the DMV, including the signing of the output messages thanks to the KSI module.
- **Data fusion stages:** Already explained in detail in 3.4, the data fusion stages involve all the actions from the pre-process of data to the post-process of the output of the fusion stage.

4.2 Integration in NAIADES Data Management layer

The Data Management layer is directly linked to FIWARE data/context management. 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 exploit 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.

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 NGSI-v2 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.

Context data providers and consumers communicate with CDM module with Context Management APIs. DFM is both a data consumer and a data provider. DFM will use the set of interfaces provided by CDM. The APIs provided define operations and data structures to enable communication, and exchange of information between CDM and NAIADES architectural components. The logical model of these interfaces is based to NGSI meta model, which is widely used in FIWARE based real world data description. Context Management APIs are intended to manage the entire lifecycle of context information including updates, queries, registrations, and subscriptions. According to the NGSI specification, it consists of the following set of APIs:

- Manage Context API: it is for the synchronous creation, modification and deletion of context entities.
- Query Context API: it provides methods for retrieval and discovery with different search filters of entities present in the Context Management module (Context Data Broker).
- Subscription API: it provides methods to manage the subscriptions to asynchronous notifications about entity updates. When some update occurs, the client will get an asynchronous notification.

5 Conclusions and Future Work

5.1 Conclusions and Results

This deliverable D3.6 "Data Fusion Middleware – Mid-term" represents the initial approach to DF design and development, it contains information about the specifications, architecture, reference implementation and API of the DF within the NAIADES technical framework. This design refers to the specifications discussed and agreed according to the architecture and functionalities defined in T2.6 (D2.9). This first version of the architecture has taken into consideration the first outcomes from T2.6, T3.1, T3.4 and T3.5 being possible to establish the grounds of the prototype, nevertheless, it will continue evolving and adapting to future issues detected during the validation of this first protype during the integration phase with the other components of the platform.

DF module current version (v1) supports the fusion process of data sets composed of more than 1 signal, the aggregation of signal readings and the simple post-process of the results, in the way of calculating average, maximum and minimum values. Further steps include the automated retrieval of data sets via API, the use of the data fusion process configurations stored in the internal database, and the insertion of data via API conforming to the DMV.

5.2 Future Work

Future work will be focused in completing the definition of the functionalities needed by the HMI and AI developers in terms of fused data (considering the data available in the context management layer), this work is strongly related to the identification/definition of the FIWARE shared data models, ongoing work being undertaken in T3.1, complete the implementation and test of the API provided by the Context Data Management module, move from the development platform (SDK-1) to deploy the DF module in the Production platform (SDK-2/cloud platform), implement KSI data signature service to ensure security specifications at platform level.

The development plan will be aligned to the availability of the necessary components and tools and will deliver the final version of the DFM (v2). The final version (v2) of (NAIADES) DFM will provide a way to extend data pre-processing (analysing, cleansing and combining data - *DoA*) by means of combination of simple and complex operations. It provides functionalities to filter time-series, combine data and apply mathematical and logical operations to selected inputs (data). These functionalities can be applied to data stored in the (NAIADES) historical DB. DFM (v2) provides a customized set of functionalities based on NAIADES end-user (water utility managers) needs, derived from the use cases defined with the pilot owners (Carouge, Braila, Alicante) to help them improve specific water management situations affecting the day-to-day operations but can easily be extended with additional fused data supporting new functionalities. DFM v2 will support (NAIADES) HMI and DSS operations, taking care of data processing and enabling direct data consumption from the historical DB.

(NAIADES) DFM is not a NAIADES AI service but a data filtering/pre-processing module responsible for processing raw data and making them available for consumption by a set of (NAIADES) services

Some examples of the data pre-processing conditions to support DSS operation, based on Alicante scenario, are shown in the table below:

		Consumption point / DSS action	
	Data preprocesing condition	School	Green area
Short term analysis	Daily consumption increase >50% compared with the last week average	- Analyze night flow (**) and look for internal leaks	- Look for internal leaks - Analyze hourly consumption pattern and compare with irrigation schedule
	Night flow > 10% of the total daily flow (use last 3 days average of night flow to discard other events)	- Is there garden irrigation at nights? - Look for internal leaks	- No specific action
	Hourly consumption >0 for all hours during 24 hours (or more)	Look for internal leaks	Look for internal leaks
Medium term (1 week) analysis	Weekly Consumption increase >15% compared with the same week of the last year	- Analyze context (e.g. weather conditions, changes in use, holidays) - Monitor consumption evolution (*) - Analyze night flow (**) to look for internal leaks	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*)
	Wekly Consumption increase >20% compared with the last week	- Analyze context (e.g. weather conditions, changes in use, holidays) - Monitor consumption evolution (*) - Analyze night flow (**) to look for leaks	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*)
Long term (1 month) analysis	Monthly Consumption increase >30% compared with the last month	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*) - Consider water awareness campaing	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*)
	Monthly Consumption increase >20% compared with the same month of the last year	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*) - Consider water awareness campaing	- Analyze context (e.g. weather conditions, changes in use) - Monitor consumption evolution (*)

In a similar way to the DSS, specific data-processing needs are being identified for the HMI and the AI services with their developers. This process applies not only to Alicante but to all three pilots, Carouge and Braila, where end-users (water utility managers) and developers are involved in a close collaboration to identify pre-processing conditions, applicable to the data available in order to enrich information generated by the platform to help water utility staff with the decision-making and regular water management operations.

Fully detailed list of fused data (inputs, conditions and operation) related (NAIADES) platform service (i.e., DSS, HMI, AI) and pilot scenario (Alicante, Carouge, Braila) will be provided in the final version of this deliverable (v2) together with a comprehensive assessment of the interoperability potential and added value of this tool to water or cross-domain platforms. DFM system comprises a set of reactive rules in charge of detecting certain situations of interest by means of the correlation, aggregation and pattern matching over a set of data streams. This component provides a high-level and uniform access layer to the platform. Consequently, other modules of the platform access it by means of RESTful APIs. The produced output of DFM data is in JSON-V2/LD format. JSON LD is a lightweight linked data format. It is based on the successful JSON format and provides a way to help JSON data to interoperate at web-scale. JSON-V2/LD is an ideal data format for programming environments, REST Web services, and unstructured databases (i.e., MongoDB). The produced JSON data is made available in the NAIADES database and can be used by the analysis layer tools or used for interlinking purposes with available public or private data.

6 Annex

This annex includes the list of parameters/data types (inputs and outputs) initially identified by the AI module developers during the design phase of the modules.

The AI modules covered here are the following:

- Urban Water Models
- Water treatment Models
- Water consumption
- Failures and leakages prediction
- Weather forecasting
- Water demand prediction
- Consumer confidence analytics
- Water Quality forecast

Task	Partners	Inputs		Outputs		Description
		Name	Type	Name	Type	
T4.1 – Urban Water Models	**IHE**, JSI, EUT, CUP	**Existing urban water models**		Final scenarios of critical events	Report	
				After modelling of critical events,		
		Water distribution network model	Mathematical model	critical values of:		
		Urban drainage network model	Mathematical model	Flow in rivers / canals / pipes (m3/s)		
		River network model	Mathematical model	Water levels in rivers / canals (m)		
		Historical time series of		Flood extension	map	
		Flow in rivers and urban canals / pipes (m3/s)	Numerical	Flood peak at selected sites	numerical	
		Water level in rivers and urban canals (masl)	Numerical	Pressure in WDN	numerical	
		Precipitation (mm/h)	Numerical	Pressure in WDN	map	
		Water quantity incidents, causes	Report			
		Water quality incidents, causes	Report			
		Scenarios of critical events, with stakehol	Report			
T4.2 – Water Treatment Models	**AIMEN**, EUT	**WTP:**		Water Quality Index	Numerical	Based on quality data from water treatment plants to be monitored, a lab scale model (water lab) will be created. On this lab scale model, different treatments on different quality data will be simulated in order to determine, in combination with the models provided by WTPs, the best treatments on different scenarios. This service will provide a water treatment model that will be used on T4.5. Additionally, the lab scale model will be used to validate T4.5 dynamical decision tool.
		Flow (m3/s)	Numerical	Total Suspended Solids – TSS (mg/L	Numerical	
		Coagulant dose (Al2(SO4)3, FeCl2, FeSO4,...) (m	Numerical	TOC (mg/L)	Numerical	
		O3 dose (mg/L)	Numerical	Total Phosphorus – TP (mg/L)	Numerical	
		Chloride dose (mg/L)	Numerical	Total Nitrogen – TN (mg/L)	Numerical	
		Water treatment models	Model	Ammonia nitrogen (mg/L)	Numerical	
		Water sensors:		Pathogens (E. coli, Salmonella)	Numerical	
		Turbidity	Numerical	Flow (m3/s)	Numerical	
		Electrical Conductivity – EC (µS/L)	Numerical	Coagulant dose (Al2(SO4)3, FeCl2, F	Numerical	
		Dissolved oxygen – DO (mg/L)	Numerical	O3 dose (mg/L)	Numerical	
		pH	Numerical	Chloride dose (mg/L)	Numerical	
		Temperature (°C)	Numerical	Water treatment model	Model	
		TOC (mg/L)	Numerical			
		Water lab:				
		COD (mg/L)	Numerical			
		BOD (mg/L)	Numerical			
		Ammonia nitrogen (mg/L)	Numerical			
		Total Nitrogen – TN (mg/L)	Numerical			
		Total Phosphorus – TP (mg/L)	Numerical			
		Total Suspended Solids – TSS (mg/L)	Numerical			
		Ammonia nitrogen (mg/L)	Numerical			
		Pathogens (E. coli, Salmonella)	Numerical			
T4.4 – AI empowered critical water consumption monitoring	**JSI**	Water flow (m^3/min)	Numerical	Water consumption monitoring tool	GUI	Data analysis results will be provided into a web application, where a user will be able to monitor water consumption. Analysis is expected to be built in Python, web application tools are yet to be determined. Possiblities are a PHP / Python API & React or PHP / Python dynamically generating html / JS.
		Date time user water demand (m^3)	Numerical			
		Weather data	Numerical / Categorical			
		Water distribution scheme	Graph with additional attributes for nodes			
T4.5 – AI water quality monitoring & dynamical water treatment	**mb**	T4.2 ones	Numerical	T4.2 ones	Numerical	A dynamical decision tool will be developed in order to suggest the best treatment on water treatment plants. This service will require historical quality data from WTP before treatment and the treatment applied; and also historical weather data if it is available. This data will be analysed to detect trends and changes on quality data and will provide treatment suggestions automatically. T4.2 outcomes will be used to develop the decision tool and to validate the results. The algorithms will be developed using Python. Python files will be generated (the algorithms and the prediction models) to be integrated in the WMS. It will be desirable to have GPU processing with CUDA installed. It is required to have all python libraries including tensorflow-gpu, keras, etc.
		Weather station:				
		Rain (mm/m2h)	Numerical			
		Weather Temperature (°C)	Numerical			
		Solar radiation (W/m2)	Numerical			
		Wind velocity (km/h)	Numerical			
		Atmospheric pressure (mBar)	Numerical			

Task	Partners	Inputs		Outputs		Description
		Name	Type	Name	Type	
T5.1 - Failure & leakage prediction	JSI	Water flow (m^3/min)	Numerical			Data are expected to be analysed with Python. What exact libraries and frameworks wil be used are yet to be decied, however, it is desirable to have access to most common solutons (e.g. Cuda, tensorflow,..)
		Local pressure distribution (Pa)	Numerical			
		Date time user water demand (m^3)	Numerical			
		Weather data	Numerical / Categorical			
		Water distribution scheme	Graph with additional node attributes			
		Water quality parameters		lerts in case of unexpected behaviou	Numerical	
T5.2 - Weather forecasting toolkit	CERTH	Weather Temperature (°C)	Numerical	Weather Temperature (°C)	Numerical	
		Rain (mm/m²h)	Numerical	Rain (mm/m²h)	Numerical	
		Solar radiation (W/m²)	Numerical	Solar radiation (W/m²)	Numerical	
		Wind velocity (km/h)	Numerical	Wind velocity (km/h)	Numerical	
T5.3 - Water demand prediction toolkit	KT, JSI	Water flow (m^3/min)	Numerical			Data are expected to be analysed with Python. What exact libraries and frameworks wil be used are yet to be decied, however, it is desirable to have access to most common solutons (e.g. Cuda, tensorflow,..) Data are expected to be analysed with Python. What exact libraries and frameworks wil be used are yet to be decied, however, it is desirable to have access to most common solutons (e.g. Cuda, tensorflow,..). KT and JSI will be jointly working on data aggregation problems, which is perceived as a critical part of the task. Once data aggregated, KT and JSI will be conducting coordinated analysis and predictions.
		Local pressure distribution (Pa)	Numerical			
		Date time user water demand (m^3)	Numerical			
		Weather data	Numerical / Categorical			
		Water distribution scheme	Graph with additional node attributes			
		Additional parameters	T4.4, T5.1, T5.2			
		Area housing type	Categorical			
		Geographical info	Categorical			
		Residential and demographics characteristics (e.g. income, education etc.)	Categorical			
		Historical water consumption data	Numerical			
		Time-based events (e.g. major public holidays)	Numerical	Water demand predictions.	Numerical	
T5.4 - Predictive AI analytics for consumer confidence	JSI	Water quality parameters	Numerical / Categorical			By scraping freely available web sources, the sentiment will be determined, which will be a base for assesing customer confidence. The sentiment will be matched to available data (water quality params), which is how we'll establish a correlation between available data nad customer confidence.
		Assesmengts via social media	Numerical	Assesment of consumer confidence.	Numerical	
T5.5 - AI analytics & prediction for the quality of the water	AIMEN	T4.5 ones	Numerical			This service requires historical data about water quality, weather, consumption and social parameters in order to generate machine learning models able to predict short/medium term water quality parameters. Apart from historical data, the data collected during the project will be used to validate and retrain the models to improve their performance. The ML algorithms will be developed using Python. Python files will be generated (the algorithms and the prediction models) to be integrated in the WMS. It will be desirable to have GPU processing with CUDA installed. It is required to have all python libraries including tensorflow-gpu, keras, etc. We need to know how to access and how to send data from/to the WMS platform.
		Weather forecast:				
		Solar radiation (W/m2)	Numerical			
		Wind velocity (km/h)	Numerical			
		Rain (mm/m2h)	Numerical			
		Weather Temperature (°C)	Numerical			
		Atmospheric pressure (mBar)	Numerical			
		Social variables:				
		Population	Numerical			
		Consumption parameters:				
		Water demand forecast	Numerical			
		Water available	Numerical	Water Quaility Index Forecast	Numerical	

Task	Partners	Inputs		Outputs	
		Name	Type	Name	Type
T6.1 - Strategies for Public Awareness and behavioural change	ICCS, AMAEM, CUP	N/A	N/A	N/A	N/A
T6.2 - User Profile	ICCS, AMAEM, CUP	**Water sensors:** public building metered consumption	timeseries	user profile	data model TBD
		User-provided information (through the NAIADES app): user characteristics consumer stated preferences	categorical categorical		
		Infered information: observed end-user interactions with the NAIDES mobile application	categorical		
T6.3 - Personalised nudging engine	ICCS	user profile	data model TBD	behavioural change strategies (e.g. persuasive messages, graphs) that will be communicated through the app	data model TBD
				suggestions for efficient water use that will be communicated through the app	data model TBD
T6.4 - Personalised Water Behavioural Change Application	ICCS	visual representations of behavioural change strategies (e.g. persuasive messages, graphs)	data model TBD	Application	GUI
		suggestions for efficient water use	data model TBD		GUI
T6.5 - non-ICT public awareness and beha	IHE, ICCS, AMAEM, CUP				