System architecture blueprint, service requirements and data management plan

Deliverable 3.1

Authors: Eloy Hernandez (EUT), Didac Colominas (EUT), Florian Freutel (ICLEI)

June 30, 2025

Introductory Table

Project Ref. No.	HORIZON-CL6-2023-CLIMATE-01-2; GA No. 101136598
Project Title	Reliability and effectiveness of integrated alternative water resources management for regional climate change adaptation
Duration of the Project	2024-01-01 to 2027-12-31 (48 months)
WP/Task:	WP3 / Task 3.1 & Task 3.2
Document due Date:	2025-06-30
Actual Date of Delivery	2025-06-30
Leader of this Deliverable	EUT
Dissemination Level	PUBLIC
Document Status	Submitted

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101136598. This document reflects only the views of RECREATE consortium, neither the European Commission nor any associated parties are responsible for any use that may be made of the information it contains.

Deliverable Information Sheet

Version	Date	Author	Document history/approvals
1	2025-06-02	Eloy Hernandez (EUT), Didac Colominas (EUT), Florian Freutel (ICLEI)	Draft version created
2	2025-06-23	Lydia Vamvakeridou-Lyroudia (KWR)	Internal review
3	2025-06-23	Sophia Stock (adelphi)	Internal review
4	2025-06-26	Eloy Hernandez (EUT)	Revision completed
5	2025-06-30	Digu Aruchamy (EUT)	Final quality check and Submission

Executive Summary

Deliverable D3.1 is part of WP3 “Interoperable Interfaces and Architecture for Smart Water Management: RECREATE_WT” (M1–M42) and presents the foundational work for the RECREATE_WT tool’s system design. It provides the conceptual and technical architecture that will support the development, integration, and deployment of the tool throughout the project lifecycle and beyond.

The report compiles the work done so far under Task 3.1 “Co-design of the RECREATE_WT visualisation tool” (M15–M42) and Task 3.2 “Design and conceptualisation of a stable, functional, cyber-secure and interoperable architecture” (M1–M12). This includes co-design activities for the visualisation environment, the identification of user needs and priorities, and the specification of architectural requirements related to functionality, cybersecurity, modularity, and interoperability.

Key outputs include the system architecture blueprint, service layer design, preliminary data management strategy, and the initial set of requirements for visualisation components. Additionally, stakeholder feedback gathered through co-design sessions and working groups has helped shape the functionalities and visual elements of the platform, particularly in relation to each case study.

As the architectural and functional foundation of the RECREATE_WT tool, this deliverable is directly connected to deliverable D3.3 “Final Report on Service Configuration” (M42), which will document the full implementation and deployment of the system’s architecture, services, and data processing capabilities.

Table of Contents

List of Abbreviations

API	Application Programming Interface
AWR	Alternative Water Resources
CEN	European Committee for Standardization
CoP	Community of Practice
CS	Case Study
CSV	Comma-Separated Values
DSS	Decision Support System
EU	European Union
IPCC	Intergovernmental Panel on Climate Change
ISO	International Organization for Standardization
JRC	Joint Research Centre (of the European Commission)
KNMI	Koninklijk Nederlands Meteorologisch Instituut
KPI	Key Performance Indicator
NGSI-LD	Next Generation Service Interface – Linked Data
OGC	Open Geospatial Consortium
REST	Representational State Transfer
SAREF4WATR	Smart Appliances REFerence ontology extension for Water domain
SD	System Dynamics
SDM	Smart Data Model
SDMs	Smart Data Models
URI	Uniform Resource Identifier
UWOT	Urban Water Optioneering Tool
WG	Working Group
WP	Work Package
WWTP	Wastewater Treatment Plant

List of Figures

List of Tables

1. Introduction

This deliverable corresponds primarily to Task 3.2, which focuses on the design and conceptualisation of a stable, functional, cyber-secure, and interoperable architecture for the RECREATE_WT platform. Building on the requirements and inputs collected from WP1, WP2, WP4, and WP5, the objective of Task 3.2 is to define the reference architecture that will support the integration of heterogeneous data sources, the execution of simulation models, and the visualisation of the results of the simulation models related to water quality and consumption under different climate and Alternative Water Resources (AWR) scenarios.

In addition to the architecture design, this document also includes initial outcomes of Task 3.1, which covers the identification and analysis of data sources relevant to the platform. This includes the mapping of data, model inputs, contextual information, and stakeholder datasets that are critical for the functioning of the harmonisation and simulation components. These early results provide the necessary foundation to align the architecture with real data availability and expected flows.

The architecture outlined in this deliverable is designed to enable interoperability and scalability, using open standards such as NGSI-LD (Next Generation Service Interface – Linked Data) and domain ontologies like SAREF4WATR, extended as needed to cover RECREATE specific requirements. The system is built upon FIWARE components for context management, data persistence, and external integration, ensuring that the platform remains flexible and compliant with evolving EU standards (ISO, CEN).

This document details the modular structure, interfaces, data flows, and semantic integration mechanisms that will guide the implementation of the RECREATE_WT tool. It also outlines how the architecture will support both operational monitoring and long-term strategic planning, offering value to a wide range of users, from technical experts to policy makers.

Ultimately, this deliverable sets the foundation for the technical deployment of RECREATE_WT, ensuring that the platform is robust, standards-compliant, and ready to be replicated across different case studies.

1.1 Main concepts and definitions

As a baseline to introduce the guidelines to develop an architecture and specifically a FIWARE architecture solution, it is important to clarify the concepts that are used in any solution in order to have a clear understanding of the components needed and used:

- **Data model (domain data model):** It is the definition of the concepts and their attributes shaping the data that will be managed in a use case. It determines the structure of the data (how data is related) and its type (string characters, number, dates, etc…).

- **Smart data model[1]:** A smart data model includes three elements: The schema, or technical representation of the model defining the technical data types and structure, the specification, a human readable document, and the examples of the payloads for NGSIv2 and NGSI-LD versions. All data models are public and of royalty-free nature. The licensing mode grants three rights to the users: free use, free modification, and free sharing of the modifications. Moreover, it is importance to notice that the water-related data models result from different H2020 projects (FIWARE4WATER, NAIADES, SCOREWATER and AQUA3S) under the initiative of DIGITALWATER2020[2]

- **(Reference) Architecture:** In terms of software engineering, a software architecture refers to the structure of a software system. It means the software components that compose the structure, relations between components and the properties of each one. A reference architecture is an architecture that provides recommended structures and integrations of IT products and services to form a solution. The reference architecture includes accepted industry best practices or recommended technological components.

- **Interoperability:** refers to the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEEXplore, 1991). In general terms, there is a list of interoperability layers defined, which includes[3]: Interoperability governance, integrated public service governance, legal interoperability, organisational interoperability, technical/syntactic interoperability, and semantic interoperability. Regarding the scope of the project, it is important to make special focus on semantic and **syntactic** interoperability definitions (Rocha, Espirito-Santo, & Abrishambaf, 2020):

    - **Technical / syntactic interoperability:** Two or more systems capable of communicating and exchanging data over specified data formats and communication protocols. It is necessary to provide an interface for each schema, such as REST APIs. The message content needs to be serialized to be sent in a format, e.g., XML or JSON. The sender

---

[1] https://www.fiware.org/developers/smart-data-models/

[2] https://www.fiware4water.eu/news/synergy-group-digitalwater2020

[3] https://joinup.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory/3-interoperability-layers

encodes the message, and the receiver decodes the message. In incompatibility occurs when the receiver decoding rules are incompatible with the sender encoding rules.

- **Semantic interoperability:** Is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by end users of both systems. To achieve semantic interoperability, both sides must defer to a common information exchange reference model. The content of the information exchange requests is unambiguously defined.

- **Semantic interoperability through NGSI-LD[4]: NGSI-LD** is an evolution of the **NGSI v2** information model, which has been modified to improve support for linked data (entity relationships), property graphs and semantics (exploiting the capabilities offered by JSON-LD). This work has been conducted under the ETSI ISG CIM[5] initiative and the updated specification has been branded as NGSI-LD[6]. The main constructs of NGSI-LD are ***entity***, ***property*** and ***relationship***. NGSI-LD entities (instances) can be the subject of Properties or Relationships. In terms of the traditional NGSI v2 data model, properties can be seen as the combination of an attribute and its value. Relationships allow establishing associations between instances using linked data.

2. Analytical Platform

Effectively managing and operationalizing processes is a cornerstone of modern data science and engineering workflows. To meet these demands, an adaptable and robust platform has been developed to support the complete lifecycle of data science processes and models. This document provides a comprehensive overview of the platform's technical features, objectives, and capabilities, offering a clear reference for its design, functionality, and implementation.

The platform has been developed as a key component of the RECREATE project, designed to integrate and manage a diverse range of models contributed by various project partners. It serves as the central infrastructure for coordinating and operationalizing collaborative workflows, ensuring seamless integration of efforts across the project's stakeholders.

One of the platform's key capabilities is its ability to manage data originating from the sources identified and characterized in Work Package 1 (WP1), as well as analytical outputs and model results produced in Work Package 5 (WP5), including simulation outcomes, key performance indicators, and CS data. Once configured, the analytical platform autonomously executes all workflows associated with the analytical processes, covering the entire lifecycle from data ingestion and processing to the generation and delivery of actionable insights. This comprehensive, end-to-end functionality is

[4] https://documenter.getpostman.com/view/513743/SVYjSMgh
[5] https://docbox.etsi.org/isg/cim/open/CIM(20)000090_Introduction_NGSI-LD.pdf
[6] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf

fundamental to achieving the project's goals of developing scalable, efficient, and collaborative analytical solutions.

The platform offers flexibility in data acquisition to meet diverse user needs. Data can be inserted manually or automatically configured, allowing the platform to acquire data directly from predefined sources without additional user intervention. Additionally, the execution of models can be automated for continuous workflows or initiated manually by the user, providing adaptability for both real-time and on-demand analytical tasks. This versatility ensures that the platform can accommodate a wide range of scenarios and requirements, making it a critical enabler for streamlined, efficient project operations.

3. General Objective

The platform is designed to provide a cohesive environment that supports the entire process chain, from deployment and execution to result analysis. Serving as the central infrastructure for the project, the platform ensures that all analytical activities are seamlessly integrated, enabling partners to deploy, execute, and manage their models within a unified framework. This approach not only simplifies the operational complexity but also fosters collaboration by providing a consistent interface and shared environment for diverse stakeholders.

A key focus of the platform is its emphasis on reliability, scalability, and technical adaptability. These characteristics are critical to ensuring the integrity of processes and maintaining robust performance under varying workloads and requirements. By leveraging these strengths, the platform can accommodate the diverse models and analytical needs contributed by different project partners, whether they involve real-time data streams, large-scale batch processing, or complex simulations.

The platform's scalability is particularly important in a project of this scope, where partners may deploy models that vary significantly in terms of data volume, computational demands, and operational goals. Its ability to dynamically allocate resources ensures that all deployed models perform optimally without impacting other processes. Furthermore, its adaptability allows the integration of various analytical approaches and technologies, enabling each partner to implement and utilize their unique models without significant reengineering.

In addition to operational reliability, the platform prioritizes usability by providing tools and interfaces that streamline deployment, execution, and result analysis. Partners can deploy their models with minimal effort, define configurations to suit their specific needs, and access analytical results in formats tailored to their workflows. This ease of use ensures that the platform is not only a powerful infrastructure element but also an accessible and practical tool for all stakeholders within the project consortium.

4. Features and Capabilities

The platform delivers comprehensive process management capabilities through its foundational process manager API (Application Programming Interface), which supports the efficient deployment, monitoring, and control of computational workflows. To ensure data integrity and traceability, the platform incorporates robust resource management and persistent logging mechanisms, allowing for reliable auditing and detailed resource tracking. These features provide a solid foundation for managing complex workflows while maintaining transparency and consistency throughout the project lifecycle.

Specifically tailored tools for data science workflows are a cornerstone of the platform, enabling model training, data preprocessing, and execution with precision and efficiency. These tools are complemented by an advanced version control system that ensures the reproducibility of workflows and transparent documentation of model iterations. This functionality is vital for maintaining consistency and accountability, particularly in collaborative environments where multiple stakeholders contribute to analytical processes.

To enhance usability and insight generation, the platform integrates visualization tools that enable users to analyse and interpret process outcomes effectively. These tools empower stakeholders to validate results, uncover patterns, and derive actionable insights, bridging the gap between raw data and decision-making.

The platform’s adaptable architecture is built to handle varying workloads and project complexities, ensuring scalable performance across diverse use cases. Whether managing real-time analytics, handling large datasets, or supporting computationally intensive simulations, the platform is designed to meet the demands of the project’s diverse objectives.

Furthermore, the architecture is being designed with long-term sustainability in mind, aiming to ensure that the platform remains operational and accessible beyond the official end of the project, for at least five years, depending on maintenance and hosting arrangements agreed upon by the consortium.

By combining flexibility, reliability, and advanced features, the platform stands as a comprehensive solution for managing and operationalizing complex analytical workflows within the RECREATE project.

5. Technical Justification (user needs and requirements)

The design of the RECREATE_WT platform is based in a user-centred and data-driven approach that balances technical robustness, semantic interoperability, and functional relevance. From the beginning of the project, an iterative process led by WP2, WP3 and WP5 collected and analysed user needs and expectations across the four case studies (CS) (Costa Brava, Syros, Kalundborg, and North Holland). This analysis is documented in detail in the annexed report ANNEX 1: User Stories and Use Cases, which forms the foundation of the platform's functional architecture.

This annex includes personas, general and case-specific user stories, and detailed use cases describing how different stakeholder profiles—such as water managers, policy makers, infrastructure operators, and researchers—interact with the platform to achieve their goals. These scenarios were essential in identifying shared functionalities and case-specific requirements that would ensure both adaptability and replicability of the platform across regions.

The platform addresses key technical challenges typical of data-driven water management tools, including the integration of real-time and historical data, management of model complexity, and usability across diverse user types. Its architecture supports advanced process orchestration, data integrity, and reproducibility, through modular and version-controlled analytical pipelines. These pipelines form the backbone of both streaming (e.g., real-time monitoring) and batch (e.g., simulation-based planning) processes.

Two major tools emerge directly from the user stories:

- **Monitoring Dashboard:** Designed primarily for water managers, this dashboard enables real-time observation of water quality and quantity indicators, integrated with IoT sensors across different locations. It includes configurable alerts for threshold exceedance and supports decision-making for water distribution and emergency response. The platform supports this tool via streaming data connectors, alerting mechanisms, and responsive visualization components.

- **Decision Support System (DSS) for AWR Management:** Tailored for policy makers and planners, the DSS allows the simulation of long-term water management strategies involving Alternative Water Resources (AWR). It integrates projections from climate models, demand scenarios, and reuse options to evaluate policy alternatives. The platform facilitates this functionality through punctual analytical pipelines, which manage complex, resource-intensive simulations while maintaining consistency and traceability of inputs and results.

The platform's emphasis on reusable and configurable components further strengthens its ability to serve diverse needs. Using shared configuration templates, users can define data sources, transformations, and outputs that can be adapted per region or scenario without reengineering the system. This is particularly valuable in a multi-case study context, where environmental, regulatory, and infrastructural conditions vary.

Moreover, the analytical platform acts as the central framework for managing all data-driven processes. It handles the lifecycle of analysis: from ingestion and harmonization, to execution, storage, and visualization. It allows the submission, queuing, and monitoring of processes, ensuring both the responsiveness required by real-time applications and the reliability needed for strategic simulations.

In summary, the functionalities outlined in the annex have directly informed the modular design and architectural priorities of RECREATE_WT. These user-driven requirements continue to guide the implementation and will support ongoing development and validation in WP6 and WP7, ensuring that the platform remains not only technically sound, but also meaningfully aligned with real-world needs.

5.1 Data Sources

The data sources relevant to each CS have been systematically identified and catalogued under D5.1 in WP5. These sources will serve as the foundation of the Data Acquisition Layer. The different data sources for each CS are as follows:

- **CS-1 - Atlantic region – North Holland (The Netherlands)**
  - Drinking water system (network, production locations, buffers, dune system)
  - Primary water ways (surface water)
  - Industrial abstractions and discharges from/to groundwater and surface water and discharges on the sewage system.
  - Agricultural abstractions of surface water and groundwater
  - Sewage water treatment plants (location, capacity)
  - Current and expected drinking water demand (daily patterns, distinguish between households and industry)
  - Meteorological data (precipitation) (KNMI)
  - Discharges and water quality of river Rhine (Rijkswaterstaat)
  - Current and predicted salinity levels at Andijk (PWN)
  - Industrial non-potable water demand
  - Industrial abstractions and discharges from/to groundwater, and surface water and rainwater harvesting, and discharges on the sewage system.
  - Agricultural abstractions of surface water and groundwater
  - Sewage water treatment plants (capacity)

- **CS-2 - Kalundborg Industrial Symbiosis (Denmark)**
  - Flowrates (lake water, groundwater, wastewater, etc.) and chemical characterisation (including metals, trace organic chemicals with the focus on persistent mobile chemicals, PFAS, etc.) of all existing flow streams
  - Surface water volume (lake, reservoirs in CS area)
  - Data for QMRA: pathogens (bacteria, viruses, protozoa)
  - Surface conditions of runoff areas (roofs, streets, etc., soil type)
  - Geological data/Characterisation of subsurface and aquifer (to assess conditions for bank filtration)
  - Groundwater wells
  - Groundwater levels in aquifers
  - Historical time series of temperature, precipitation and evaporation.
  - Distribution networks
  - Energy demand of existing WWTP and DWTP
  - Estimates for full-scale water reclamation plant: quantity and quality of produced water and brine, demand of energy and chemicals of the system

- Estimates for seawater desalination: quantity and quality of produced water and brine, demand of energy and chemicals of the system
- Estimates for rainwater storage and treatment: quantity and quality of produced water (and concentrate/brine?), demand of energy and chemicals of the system
- Water demand (quantity and quality) of new industries
- Flowrates and wastewater types of new industries
- Energy and chemicals demand of future (expanded) WWTPs
- Weather Data
- Estimates for future water availabilities of conventional water resources (groundwater, surface water; e.g. based on regulative restrictions)

- **CS-3 - Mediterranean Self-Sustained Area - Syros South Aegean (Greece)**
  - Consumption/Population Data per consumer group (permanent residents, tourism, farmers)
  - Agricultural demand (total irrigation demand, percentage of agricultural demand covered by treated water (wastewater)
  - Consumer demographic data (number of households, household members, income, age)
  - Consumer awareness campaigns in the project
  - Energy and water bills as time series: (daily or monthly step) and tiered tariffs
  - Historical time series of temperature and precipitation (daily timestep)
  - Climate data trends: Downscaled IPCC scenarios in the form of time series of the climatic variables for the predetermined simulation horizon (i.e., 2024-2050, 2050-2099)
  - System operation rules (optional)
  - Desalination
  - Wells
  - Volume and location of reservoirs
  - Main gravity and pressure pipelines
  - Water treatment plants / water treatment 7nits
  - Wastewater treatment plants
  - Water distribution network
  - Household tanks
  - Soil texture
  - System structure

- **CS-4 - Costa Brava (Spain)**
  - Surface water volume (reservoirs, rivers within the study area)
  - Groundwater levels and flow rates (state of the aquifers)

  - Water quality (full chemical analysis: COD, BOD, TOC, N, P, TSS, 254nm absorption, contaminant analysis, pathogens) of treated wastewater effluents.
  - Data for QMRA: human related faecal pathogens (e.g. CrASS-phage, coliforms)
  - Aquifer characterization: depth, soil and water quality)
  - Temperature (historical maximum, minimum, average)
  - Precipitation (historical maximum, minimum, cumulative)
  - Water demand per sector (agricultural, industrial, touristic seasonal, urban)

- Water distribution network (length, age, losses throughout the network, population served)
- Energy demand of WWTP, DWTP, and desalination plant
- Historical time series of temperature, precipitation and evaporation
- Groundwater storage dataset (GLDAS-2.2/GRACE-GRACE-FOLLOW)
- Projection of future water demand, water availability, energy demand, water quality

These sources will serve as the foundation of the data acquisition layer (Section 0).

5.2 Data Acquisition Layer

The Data Acquisition Layer represents the starting point of the platform's workflow. Where all the data required to power models and processes is gathered. This layer plays a critical role in ensuring that the right data is available at the right time, setting the foundation for accurate and effective analysis.

Data can be acquired through several flexible methods, depending on the needs of the user or the system. In many cases, automated connectors are used to pull data from integrated sources, either on a scheduled basis or when a specific process is triggered. This helps maintain up-to-date information without requiring manual input.

Alternatively, users can choose to retrieve or input data manually through the platform's graphical interface, especially when preparing to run a specific model or analysis. For more advanced or integrated use cases, data can also be accessed or submitted programmatically by calling the platform’s API.

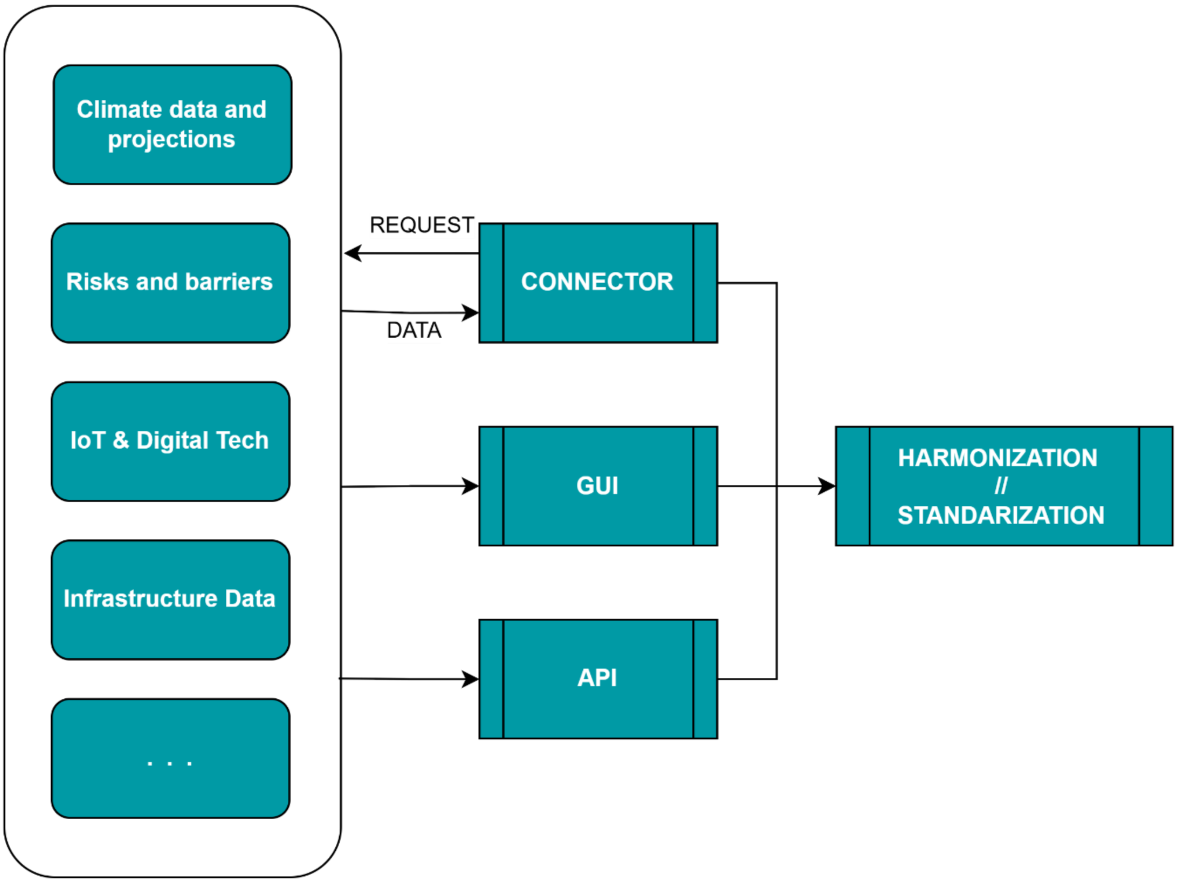

Figure 1: Data Management Schema

5.2.1 Connectors (Data Collectors)

Connectors (aata collectors) are pre-configured adapters responsible for retrieving raw data from various external systems, either on a scheduled basis or in response to specific events. These connectors serve as the platform's interface with the outside world, abstracting away the complexities of each individual data source—such as authentication methods, file system paths, network protocols, or database credentials. Their primary role is to ensure that regardless of the origin or structure of incoming data, it is converted into a consistent, platform-wide canonical schema that downstream components can readily consume.

Beyond abstraction, connectors also handle the transportation to harmonization layer. This process often includes intelligent schema mapping, type conversions, and the resolution of discrepancies such as missing fields or inconsistent naming. Moreover, as data sources evolve—by adding new fields or changing formats—connectors are designed to detect these changes dynamically and manage schema drift in a controlled way, sometimes by routing new structures to temporary or versioned landing zones until they can be formally incorporated.

Security and governance are integral parts of the connectors' architecture. Credentials and tokens are typically retrieved from secure vaults rather than being embedded in code or configuration files. Each data pull operation is logged in a detailed audit trail, capturing metadata such as timestamps, payload sizes, source systems, and success or failure status—providing both transparency and traceability for compliance purposes.

From a reliability perspective, connectors are built with fault-tolerant and idempotent mechanisms to ensure robust operation in production environments. For instance, they track offsets or watermarks to prevent duplicate ingestion during retries and apply exponential backoff strategies when encountering transient errors. In cases of persistent failure, connectors can isolate problematic sources, preventing them from degrading the overall ingestion pipeline.

Performance and scalability are also key considerations. Connectors can be deployed in parallel across distributed environments, such as container orchestration systems, to handle large volumes of data efficiently. They support both full data loads and incremental syncs—pulling only data that has changed since the last run—thereby optimizing bandwidth and processing time. To avoid overwhelming downstream systems, many connectors incorporate back-pressure mechanisms, buffering data temporarily or adjusting pull frequency based on system load.

5.2.2 Graphical User Interface (GUI)

The platform also supports manual data acquisition through a graphical user interface (GUI). This method is particularly useful in scenarios where users need to quickly test a model, run an ad hoc analysis, or work with one-off datasets that aren't available through a continuous data stream.

Through the GUI, users can easily upload data either by selecting files directly from their local system or by providing a URL pointing to a remote resource such as a hosted CSV, JSON feed, or external API endpoint. This upload interface is tightly integrated into the process execution workflow—meaning users are prompted to provide data inputs at the point they initiate a model run or analysis task. As such, this mode of ingestion is more interactive and event-driven, triggered explicitly by a user's action rather than running in the background like automated connectors.

Once the data is uploaded or fetched via URL, the platform automatically validates it against expected formats and schemas defined for the given process. Any necessary normalization or transformation is performed behind the scenes to ensure compatibility with the platform's internal data models. This makes it possible for even non-technical users to provide inputs without worrying about structure, formatting, or preprocessing requirements.

Furthermore, this method offers traceability and repeatability. Uploaded datasets are typically versioned and stored alongside metadata about the process run. Including who uploaded the data and when. This ensures that model executions remain reproducible and auditable over time, even when manually initiated.

5.2.3 Application Programming Interface (API)

The platform provides a comprehensive suite of RESTful APIs that enable programmatic interaction with all catalogued data sources. This interface is designed for developers and system integrators who need to embed data access and exchange directly into their own applications, workflows, or external services—offering a powerful and flexible alternative to manual or scheduled ingestion methods.

Through the API, external systems can securely retrieve data from the platform in a structured, on-demand fashion. Whether pulling historical datasets or querying the latest available record users have fine-grained control over what data they access and how it is delivered. In addition to simple fetch operations, the API supports event-driven interactions, allowing clients to subscribe to change notifications. This means that external applications can be alerted in real time when the model have completed his execution, enabling more reactive and dynamic data workflows.

The API doesn't just support data retrieval—it also enables publishing of new data streams into the platform. This allows third-party services, custom scripts, or external pipelines to inject data into the system in a consistent and validated manner. Each API call can be authenticated and authorized based on user roles and scopes, ensuring that data access and publishing are tightly governed and auditable.

5.3 Data Harmonization Layer

This module is a key component in the architecture of the tool, as it is responsible for transforming heterogeneous raw and specific datasets into standardized and interoperable entities that can be consumed by models and visualization tools. It acts as a bridge between the data acquisition layer (Section 0) of raw data via connectors, APIs or manual uploads and its integration into the semantic and analytical workflow of the platform, which consists on transforming this data into smart data models (Section 6.1).

Additionally, the layer is responsible for data cleaning and normalization processes, which is specially important for integrating data from many different temporal spatial scales and values.

By aligning harmonized data to NGSI-LD models and using open formats like JSON-LD, the platform ensures full interoperability with external systems, including FIWARE components (such as the Orion Context Broker), dashboards, and policy-support tools.

The harmonization layer also supports both real-time and batch processing. For example, it can process continuous streams from IoT sensors for operational dashboards, as well as large historical datasets, such as, climate scenarios used by UWOT or System Dynamics models for long-term strategic planning.

Ultimately, the outputs of the Harmonization Layer are semantically enriched, NGSI-LD compliant (FIWARE protocol) entities stored in the platform. These harmonized datasets feed directly into model execution, visual dashboards, and third-party applications through APIs, enabling robust, explainable, and reproducible insights across all regions and stakeholder groups.

5.4 Data Management Layer

This layer is responsible for the secure and scalable storage of all data within the RECREATE_WT. This layer ensures that real-time data and historical data are consistently available for analysis, visualization and reuse, and additionally guarantees data integrity, traceability ad accessibility.

This layer operates as the central point where the outputs of the data harmonization layer are stored and structured, allowing components (such as UWOT, System Dynamics, the pathway engine, or visual interfaces) to query and retrieve data in a semantically consistent way. To support this, the data management layer integrates a combination of time-series optimized storage, document stores, and relational databases, depending on the format and access of each data stream.

5.4.1 TimescaleDB

Within the RECREATE_WT platform, TimescaleDB is used to store time-series data. It is integrated through QuantumLeap, a FIWARE generic enabler that listens to context changes in the Orion Context Broker and writes them into TimescaleDB.

This database allows the platform to:

- Record the evolution of NGSI-LD entities over time, enabling historical analysis.
- Support time-based queries
- Provide reliable access to both real-time and past data for dashboards, simulations, and reports.

TimescaleDB is optimized for efficient storage and retrieval of time-series data, making it ideal for the RECREATE_WT use case where many datasets change frequently and need to be tracked over time. By combining FIWARE's context management with time-series storage, the platform ensures that all data remains traceable, accessible, and ready for analysis across all CS [7].

[7] https://fiware-tutorials.readthedocs.io/en/latest/time-series-data.html

5.4.2 File System Storage

The platform provides users with the capability to upload and manage a wide range of data types, including structured and unstructured files. Once uploaded, these files are securely stored within the platform's source storage system. This design ensures that the data remains protected and isolated, and it is only made accessible to AI models on a demand-driven basis. By restricting automatic or background access, the platform minimizes the risk of unintended inferences or data leakage, thereby maintaining a strong security and privacy posture.

In cases where AI models generate output files as part of their execution, these results are stored separately in a dedicated sile system storage. This output data is automatically indexed, enabling precise tracking and association with the specific model run that produced it. Similarly, input files used in each model execution are also indexed and linked, creating a comprehensive and traceable record of all file interactions within the system.

This structure ensures that every model execution maintains a clear separation of input and output data, supporting reproducibility, auditing, and compliance. It also enhances operational transparency, allowing users to review, reuse, or analyze data and results across different sessions or workflows with confidence and efficiency.

5.5 Model Orchestrator

The model orchestrator is a core component responsible for the comprehensive management and coordination of model execution within the system. Its primary role is to oversee and streamline the execution pipeline for various analytical and predictive models. This includes handling all preparatory and post-execution tasks associated with data and model operations.

Specifically, the model orchestrator is tasked with:

1. **Data Management:** The orchestrator determines and manages the data inputs required for each model execution. These inputs may originate from:
    a. The data acquisition layer, where raw data is collected from external or internal sources.
    b. The data harmonization layer, where acquired data is cleaned, transformed, and aligned to ensure consistency and compatibility across different models.
    c. User-provided data, which can be manually input through the User Interface (UI) for ad hoc or scenario-specific model runs.

2. **Model Execution Coordination:** Upon defining the appropriate data inputs, the orchestrator initiates and monitors the execution of the selected models. It ensures that models are run in the correct sequence, with the right parameters, and in the appropriate execution environments (e.g., cloud-based, on-premises, or containerized infrastructure).

3. **Results Delivery and Integration**: After the model completes its execution, the orchestrator is responsible for collecting, formatting, and delivering the results. Output can be:
    a. Displayed in the user interface for user review.
    b. Accessed programmatically through the API.
    c. Pushed directly to external systems or endpoints, such as dashboards, reporting tools, or decision-support systems, depending on the integration requirements.

Additionally, the model orchestrator implements logging, monitoring, and auditing functionalities to support traceability, debugging, and compliance with data governance policies.

5.5.1 Model Coordinator Layer

The model coordinator layer serves as the control center where users define the specific setup under which a model will be executed.

One of its core responsibilities is handling **model configuration**. In this phase, users provide all necessary inputs that determine how the model should function. This includes selecting the data sources—either from previously acquired and harmonized datasets or custom datasets uploaded through the user interface. In addition, users can define execution parameters tailored to the model's requirements. These might include thresholds, time ranges, algorithmic settings, or scenario-specific assumptions. By externalizing this configuration from the model logic itself, the system supports greater flexibility and reusability.

Equally important is the **execution definition**, which outlines the operational context in which the model will run. Through this component, users can specify whether the model should be executed immediately (on-demand), on a recurring schedule, or triggered automatically by specific system events or changes in data. This approach allows for both exploratory, user-initiated runs and fully automated workflows. It also enables advanced use cases such as scenario comparisons or benchmarking against historical baselines. These strategies ensure that models are not only configured correctly but also executed in a way that aligns with business objectives and operational constraints.

Finally, the **result presentation configuration** enables users to define how and where model outputs will be delivered. Results can be displayed directly within the UI as visual elements (charts, tables, relative pathways, etc.) or sent to external systems via APIs. Users may also choose specific output formats (e.g., CSV, PDF, or JSON) and apply post-processing rules such as filtering, sorting, or annotating results. This flexibility ensures that insights are presented in a meaningful and actionable way, supporting clear communication and integration into broader decision-making workflows.

5.5.2 Execution Coordination Layer

The execution coordination layer is a critical component designed to manage and orchestrate the various executions of models or processes within a system. It acts as a centralized hub that not only initiates and tracks the status of each execution but also provides visibility and access to their respective outputs.

This layer supports multiple concurrent executions, each of which can be in a distinct state (successful, failed, or running). These statuses are dynamically updated and presented to the user through both the API and the user interface, enabling real-time monitoring and streamlined interaction with ongoing and completed processes.

Each execution is assigned a unique identifier, which serves as a persistent reference to that specific run of a model or process. This identifier is essential for traceability, allowing users to retrieve detailed information, inspect results, and analyze execution performance at a granular level. Through this mechanism, users can easily access past executions, compare outputs across different runs, and identify trends or anomalies.

Abstracting the complexity of execution management, the execution coordination layer facilitates a seamless user experience, empowering users to launch, track, and interact with model executions in a scalable and organized manner. This is particularly valuable in environments where models are frequently retrained, re-executed with different parameters, or compared for validation purposes.

5.6 Execution Workflow

At the heart of this data processing platform lies a modular and flexible architecture designed to support various analytical processes. The workflow is governed by a top-level component called the ProcessRunner, which is responsible for orchestrating and managing the complete lifecycle of data processing tasks. This involves coordinating three primary components: Source, Pipeline, and Sink.

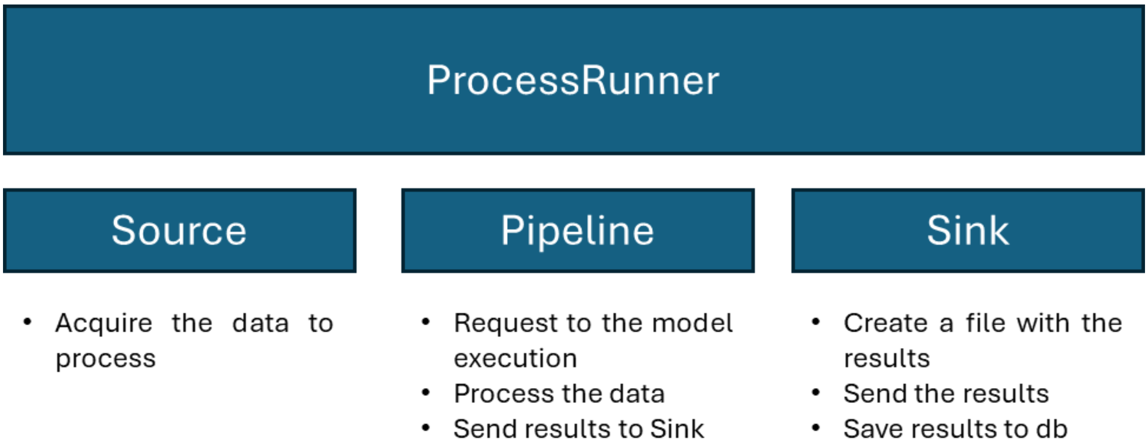

Figure 2: Process Runner Execution Workflow

5.6.1 Data Ingestion Configuration (Source)

The source represents the entry point of data into the system. It is defined by a set of configuration parameters referred to as DataSourceParams, which dictate how and from where the data is acquired. The platform supports a wide variety of data sources, including:

- Pub/Sub systems
- File systems (e.g., CSV, JSON)
- APIs (e.g., REST-based polling mechanisms)

This component ensures that data is reliably and efficiently ingested into the platform, tailored to the specifics of the input source format and access method decided on the data harmonization layer.

5.6.2 Analytical Core (Pipeline)

At the heart of the platform's data processing architecture is the Pipeline. Pipelines are a structured, step-driven mechanism through which data is transformed into valuable insights. Situated between the Source (data ingestion) and the Sink (output delivery), the pipeline is where analytical logic is executed, simulations are run, and meaningful results are generated.

The pipeline is composed of a defined sequence of steps, each responsible for a specific transformation or computation. As data flows from the source, it is passed through these steps in a predetermined order. Each step is configured with its own operational logic, enabling the pipeline to break down complex analytical processes into manageable, traceable operations.

These steps cover a broad range of tasks, including:

- Cleansing and filtering
- Enrichment with auxiliary datasets
- Simulations and predictive modeling
- Metrics aggregation and result formatting

This structured flow ensures data consistency, analytical rigor, and repeatability across all executions.

Model Integration via API: UWOT and SDM

A key functionality of the pipeline is its integration with external simulation models—specifically, the UWOT and SDM models. These models are not embedded within the platform itself but are accessed externally via API endpoints, enabling decoupled, scalable simulation processing.

- UWOT (Urban Water Optioneering Tool): Within the pipeline, a designated step is responsible for preparing the necessary input data (e.g., land use patterns, population data, weather profiles), and sending it via API to the UWOT service. The model processes the data externally and returns simulated outputs—such as projected water demand profiles and intervention impact metrics. The pipeline step then collects the

API response, parses the results, and passes them along for further processing or direct delivery to the Sink.

- SDM (System Dynamics Model): Similar to the procedure followed for UWOT, a dedicated pipeline step prepares and formats the input data required by SDM (e.g., infrastructure parameters, supply-demand relationships, policy levers) and communicates with the model via API. The SDM executes the simulation externally and returns a detailed assessment of system performance, resilience metrics, and future state projections. The results are then retrieved by the pipeline step and integrated into the data stream.

5.6.3 Result Delivery (Sink)

Upon completing all pipeline steps, including those that apply the UWOT and SDM models, the enriched and analytically processed data is forwarded to the Sink. The Sink then ensures that this output is delivered to the appropriate downstream system—whether that be a database, file, API, or messaging system—ensuring that insights are accessible, actionable, and timely.

5.6.4 Complete Workflow

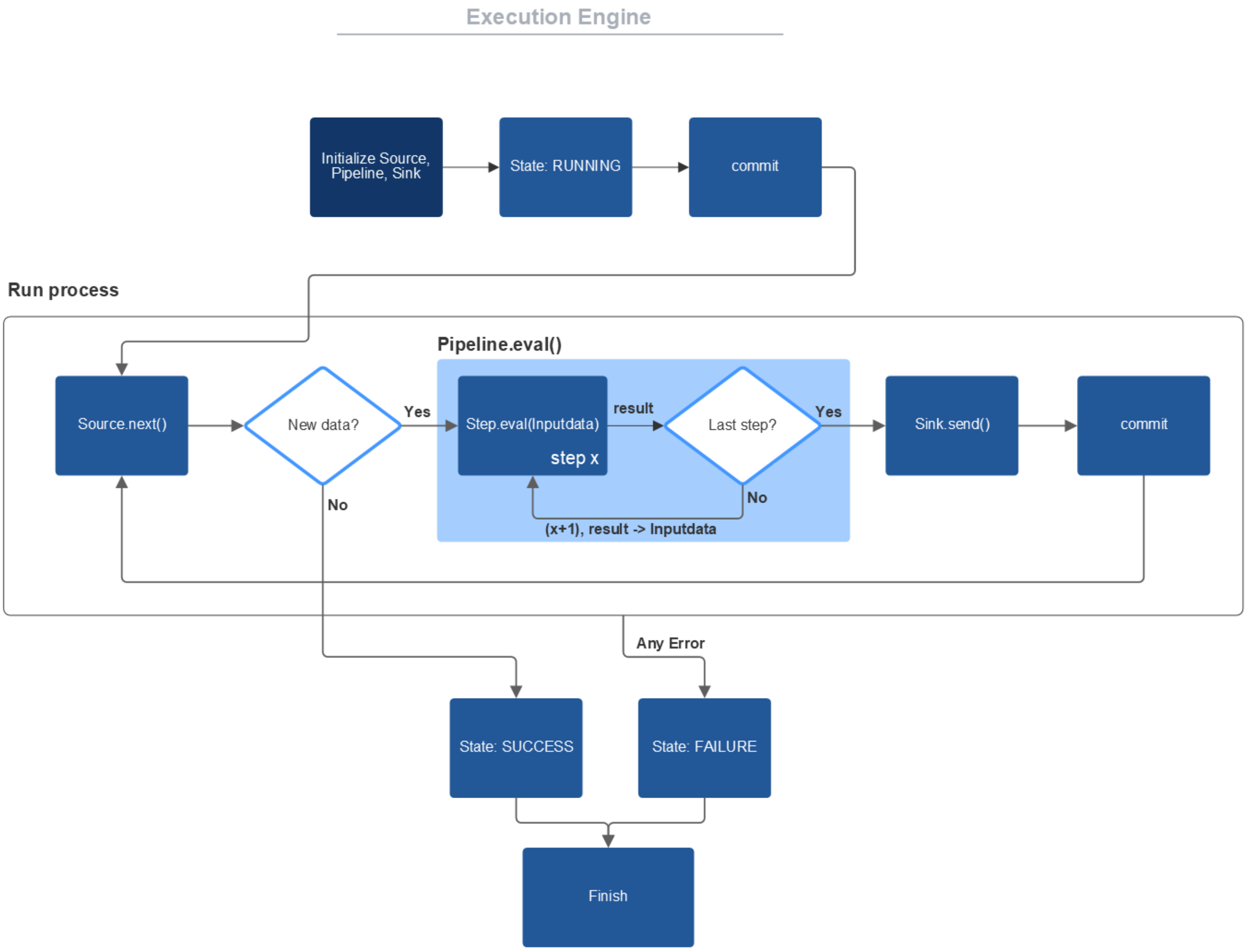

Figure 3: Execution Engine Workflow

5.6.5 Initialization and Execution

The process begins with the initialization of the Source, Pipeline, and Sink components. Upon initialization, the system transitions to a RUNNING state, after which the core processing logic begins:

1. Source fetches the next available data item.
2. The system checks whether new data is available.
    a. If no data is present, the pipeline loops back and checks again.
    b. If new data is available, it proceeds to be evaluated by the pipeline.

5.6.6 Pipeline Execution Logic

The pipeline orchestrates the stepwise evaluation of incoming data. This is the central segment of the architecture where the core analytical work is done. The pipeline is composed of a series of Steps, each designed to perform a specific transformation or computation.

Each step takes in InputData, performs its logic, and produces a result. If the current step is not the last, the result becomes the new InputData for the next step, continuing until all steps are completed.

5.6.7 Finalization

Upon reaching the last step, the transformed data is passed for output delivery. A commit operation follows to record successful data processing.

5.6.8 Error Handling and Completion

The pipeline includes robust error handling:

- Any failure during execution results in the State: FAILURE and exits the pipeline to finish.
- If the pipeline completes all data without issues, it transitions to State: SUCCESS, followed by the finish state.

6. Technical Structure of the Platform

6.1 Smart Data Models

The RECREATE_WT adopts Smart Data Models (SDMs) to ensure semantic **consistency, interoperability, and reusability** of data across all case studies and modules.

These modules, based on NGSI-LD standard provide a shared vocabulary and structure for representing entities such as water resources, infrastructure components, consumption and simulation results.

To be aligned with the **open and collaborative smart data models initiative**[8] the platform ensures that all harmonized data conforms to internationally recognized schemas. This facilitates:

- **Interoperability** between data sources, models (UWOT, SDM) and external systems (JRC tools)
- **Scalability and reuse**, as new data sources can be mapped easily to existing models.

Each data entity ingested into the system is described using standard NGSI-LD format, with the following attributes:

- **@context** to define semantic meaning.
- **@type** to indicate the entity class (e.g., WaterQualityObserved, WaterConsumptionForecast);
- **id** as a unique identifier.
- Attributes defined as property or relationship.

As described in section 5.3 the data harmonization layer is responsible for transforming raw data from sensors, datasets and APIs into NGSI-LD compliant entities based on SDMs. These entities are then available in the platform for monitoring, model execution, and visualization.

By using Smart Data Models, RECREATE_WT ensures alignment with widely adopted standards, reduces integration complexity, and improves the replicability and portability of its outputs across regions and systems.

[8] https://smartdatamodels.org/

According to the expected data to collect and the currently available datasets we have compiled a list of smart data models selected for use in the the RECREATE_WT:

PLATFORM ENTITY	SDMS	USE IN RECREATE_WT
Real-time sensor data	WaterQualityObserved	Used to monitor parameters like pH, turbidity, pathogens.
Water demand	WaterConsumption	Generated by models (e.g., UWOT, SD) under different scenarios.
Water water plant facility	WasteWaterPlant	Static infrastructure metadata and operational attributes.
Water DistributionManagement	WaterDistributionManagementEPANET	Tracks water availability by case study (e.g., Syros, Costa Brava).
Climate scenario	WeatherForecast	Used for long-term planning and adaptive pathway construction.
Energy consumption (WWTP)	EnergyCIM	Supports sustainability KPIs and operational efficiency metrics.

Simulation result (model output)	ScenarioSimulationResult*	Custom extension to store UWOT/SD simulation outputs.

Table 1 Smart Data Models used for RECREATE_WT entities

* Custom data model aligned with NGSI-LD principles, not existing but proposed where no standard SDM currently exists.

This structured approach ensures that the platform is technically robust, semantically aligned with EU standards, and ready for future integration and replication beyond the current project scope.

6.2 Authentication

The API introduces a foundational mechanism for enabling user authentication and resource ownership management. This mechanism allows each request to carry identity information, facilitating personalized access to submitted resources and processes. By incorporating an “auth” parameter in the request headers, the API can associate actions and data with a specific user or entity.

This approach ensures that resources are only accessible and modifiable by their respective owners, thereby enforcing data isolation and security. If no user identity is provided, resources are handled under a default classification, ensuring that unauthenticated interactions are still processed but clearly distinguished.

The system uses a simple identifier to denote ownership. However, the underlying concept is designed to support a more secure and scalable authentication framework, using tokens or credentials to validate and authorize user identities across the platform.

6.3 Submission Process

The platform provides a versatile endpoint designed to handle the submission of analytical processes, accommodating different execution needs through two distinct modes: asynchronous and synchronous.

When a user submits a process **asynchronously**, the request is accepted immediately, allowing the platform to begin processing in the background. This approach is particularly useful when submitting multiple tasks or when the user does not need to wait for results right away. It enables smooth, uninterrupted workflows and is ideal for high-throughput or batch processing scenarios.

Alternatively, the **synchronous** mode offers a more direct interaction. In this case, the user submits a process, and the platform holds the request open until the analysis is complete and the results are ready. This mode is well-suited for situations where immediate feedback is required, such as interactive sessions or decision-making tasks that depend on real-time outputs.

To manage and track each submitted process, the system relies on unique identifiers. Users may explicitly provide a process ID as part of their submission. However, if the provided ID is already in use, the system will reject the request to avoid conflicts, responding with a clear error code.

For greater convenience and to reduce the risk of duplication, users also have the option to let the platform handle ID generation automatically. By setting the ID field to a null-equivalent value (such as “null”, “None”, or an empty string), the platform will create a unique identifier on the user’s behalf. This ID is then returned in the response and can be used to query the process status or retrieve results at a later stage.

6.4 Process Management

The platform implements a comprehensive mechanism for managing the state and performance monitoring of analytical processes. These capabilities are essential for ensuring reliable execution, system resilience, and operational transparency.

6.4.1 Process State

Each component of an analytical process—whether a data source, transformation step, or output destination—maintains an individual process state. This state consists of a structured set of variables that represent the current execution context of the component and its position within the overall workflow.

The process state enables the platform to recover seamlessly in the event of a system interruption or failure. For instance, a source component interacting with external systems may record the identifier of the last data item processed. Upon restart, the platform can reference this stored information to resume execution from the precise point at which it was interrupted, thereby preventing data inconsistency or redundancy.

State information is persistently maintained and updated during each execution cycle. The platform stores separate state records for each process component—source, pipeline, and sink—ensuring clear segmentation and traceability. These updates are managed by the Analytical Process Execution Engine (APEE), which commits the state at every iteration. In case of failure, the stored state is automatically made available to the relevant components, facilitating a controlled and consistent recovery process.

6.4.2 Process Execution Metrics

In parallel with state management, the platform captures execution metrics generated independently by each component involved in the process. These metrics provide critical insights into the performance and operational characteristics of the analytical workflow.

Each component collects metrics relevant to its function, such as execution frequency, processing time, or resource utilization. These measurements support ongoing performance evaluation and enable informed decisions regarding system optimization and scaling.

The collected metrics are aggregated and presented to clients as part of the Process Status Report, which is available through the getStatus(link) API method. This report serves as a monitoring tool, offering a detailed view of the process's execution profile and aiding in diagnostics, auditing, and operational assessment.

7. AI Model Manager

The Analytical Process Execution Engine (APEE) is a vital component of the platform, built around the concept of a **process**. A process is defined as an independent, stateful execution instance of a pipeline. The independence of each process ensures that no data is shared between processes running in parallel, even if they are executing the same pipeline. This isolation is crucial for maintaining reliability and scalability, particularly when handling simultaneous analytical tasks or managing large-scale data streams across multiple regions.

Statefulness is a defining feature of the APEE, enabling each process to maintain its own set of attributes necessary for execution. These attributes are preserved even in the event of a system failure, providing strong fault tolerance and continuity. For example, in a process that uses a messaging system as its input, the last committed offset of the message queue is stored and maintained, ensuring that data integrity is preserved and no information is lost. By retaining such stateful information, the APEE ensures that analytical processes are resilient and can continue seamlessly after interruptions, which is critical for real-time data applications.

The APEE's primary function is to manage the lifecycle and execution of these processes, ensuring their state is consistently persisted. This capability is especially valuable for the RECREATE project, as it supports the uninterrupted operation of tools such as the Real-Time Monitoring Dashboard and the Decision Support System for AWR Management. For the Real-Time Monitoring Dashboard, the APEE enables continuous and reliable monitoring of data streams from IoT sensors, ensuring that real-time updates and threshold alerts are delivered even during technical disruptions. In the Decision Support System, the APEE allows complex simulations to run efficiently and recover gracefully if needed, ensuring policymakers receive accurate and uninterrupted insights for sustainable water management.

8. Analytical process life cycle

The platform incorporates a **Process API** that enables the management and execution of analytical processes. This API provides mechanisms for submitting new processes, monitoring their status, and terminating them when necessary. These capabilities are essential for maintaining control over the lifecycle of analytical processes, ensuring that they operate efficiently and adapt to evolving requirements.

Execution begins with a request to the API's submit endpoint, where the analytical process is defined using ProcessParams. These parameters specify critical elements, such as the input data source, the pipeline to execute, and the output destination. Upon submission, the process enters a queue with a PENDING status, awaiting the allocation of a thread. This queuing system ensures orderly execution and optimal resource utilization across the platform.

Once a thread is available, the process transitions to a RUNNING status, and execution begins. During this phase, the Analytical Process Execution Engine (APEE) orchestrates the operation by receiving input data from the defined source, processing it through the specified pipeline, and storing the results in the assigned output destination. This structured approach ensures that the process delivers consistent and accurate results.

The process runs until one of two conditions occurs: either the input source is fully consumed, or a termination request is issued through the API's terminate endpoint. This termination capability ensures flexibility, allowing processes to be halted when no longer required or when adjustments are needed.

For the RECREATE project, the Process API is a foundational component that directly supports critical tools such as the Real-Time Monitoring Dashboard and the Decision Support System for AWR Management. By facilitating efficient execution and lifecycle management of analytical processes, the API ensures that real-time monitoring systems can handle continuous data streams while maintaining reliability. Similarly, it allows complex simulation tasks to proceed without interruption, providing accurate and timely insights for long-term water resource planning. This integration of structured process management is essential for achieving the project's goals of delivering scalable, adaptable, and high-performing analytical tools.

9. Model's Management Process

9.1 Deployment & Configuration

Due to data governability requirements, both the SDM and UWOT models, will be hosted externally, outside the RECREATE platform server environment. This architectural decision aligns with data governance policies and enables independent management, scaling, and maintenance of these models while preserving data security and integrity.

The RECREATE platform is built with the capability to interact with external computational models through standardized integration mechanisms. For the UWOT model, interaction will be established via a dedicated external API. The UWOT model is maintained and executed outside the RECREATE infrastructure and is not containerized within the platform. Instead, the platform sends requests to the UWOT API, which processes the required simulations and returns the results to RECREATE for further use.

The SDM model is similarly hosted outside the RECREATE Platform environment. Depending on the integration requirements, the platform may interact with SDM using predefined interfaces or data exchange protocols to send inputs and retrieve model outputs. Like UWOT, SDM remains under separate operational control, ensuring compliance with model-specific governance and deployment considerations.

The RECREATE platform itself is fully containerized using Docker[9][10], promoting portability, modularity, and ease of deployment across diverse environments. This containerized setup supports the internal services of the platform while allowing for seamless interaction with external models like SDM and UWOT.

Once results are received from these external models, the RECREATE Platform processes and stores the output data in two primary repositories:

- A relational database, for structured, queryable data, ensuring efficient access and integration with other platform modules.

[9] https://www.docker.com/

[10] *Docker is a platform that allows developers to package applications and their dependencies into lightweight, portable containers that run consistently across different environments*

- The file storage system on the server, used for managing unstructured data, simulation files, or any additional documentation associated with the models.

9.2 Application Programming Interface

Users can interact with the module through this API, with the available endpoints detailed in the subsequent sections.

9.2.1 Authentication

The user can indicate the owner of the process with a header parameter in all the requests; the header key is "auth". The value indicates the process owner name; in future upgrades, this parameter will contain a token that will help the platform to validate the identity of the "requester".

A process/resource submitted with the header "auth", will only be accessible, modifiable, and visible by the authorized user. If no auth key is provided it will be assigned to the user "UNIDENTIFIED_OWNER".

Key	Value	Description
auth	User_1	Indicates that the request is performed by "user 1"

Table 2 Authentication API description

9.2.2 Submission process

The endpoint submits an analytical process to the platform; it can be submitted in two ways:

- **asynchronous** submissions can be useful to throw multiple processes without blocking the request.
- **synchronous** submissions can be useful to force the request to wait until the result are generated.

The id provided on the request must be unique among all the processes registered in the system, you can get the list of processes available through Get list of process status endpoint. Otherwise, the object response will contain as code attribute a 501.

Another option is to let the platform generate and associate an ID to the process, the user can do it by setting the id attribute to (“Null”, “null”, “None”, “none”, “”) in the request body, internally the platform generates an id that is available in the response body on the attribute pid.

Operation	**POST**		
Endpoint	/api/process/	/api/sync/process	
Header	Content-Type	application/json or multipart/form-data;	Multipart option allows the user to send files with the request.
Parameter	-		
Body	-	Based on ProcessParams	This will contain a ProcessParams object as json like in request body json section
Multipart Body	process_params	Based on ProcessParams	This will contain a ProcessParams object as json like in request body json section
	{name of the file}	File	User can add multiple files, just need to add more keys. Consider that the files will be saved using the key provided as its name.
Responses	**default**	{ "code": {int}, "msg": {str}, "pid": {str}, }	File created on the process

	SYNC * ResultSink	result	If using the sync variant of the endpoint and the Sink implements the “get_result_object” method, it will call it and return the result in the request.

Table 3 Submit endpoint API description

Request body

The body of the request uses a dictionary based on the model ProcessParams and can be provided using two different methods:

- **application/json** default approach, providing just the configuration json of the process the user desires to run;
- **multipart/form-data** allows the user to provide the json under the key “process_params” as in the default method, plus a key for each one of the files need.

Format:

```
{
  "id": {str},
  "type": {typeEnum},
  "pipeline": [
    {
       "cls": {cls(str)},
       "config": {StepParams}
    }
  ],
  "sink": {
    "cls": {cls(str)},
    "config": {SinkParams}
  },
  "source": {
    "cls": {cls(str)},
    "config": {SourceParams}
  }
}
```

Body example:


```
{
  "id": "id1",
  "type": "PUNCTUAL",
  "pipeline": [
      {
      "cls": "sms_analytics.test.factorial_pipeline.MultiplyStep",
      "config":{}
      }
  ],
  "sink": {
    "cls": "sms_analytics.test.factorial_pipeline.TestSink",
    "config":{}
  },
  "source": {
    "cls": "sms_analytics.test.factorial_pipeline.SequentialNumberGenerator",
    "config":{
      "start_number": 1,
      "limit_number": 3
    }
  }
}
```

RESPONSES

All responses will use the following json structure:

```
{
  "code": {int},
  "msg": {str},
  "pid": {str},
}
```

Response 200 Example: success with ID provided on request

```
{
"code": 200,
"msg": "Pid submitted properly, id: afn6mzM0VjJ8d0wcunu0tgXiJlx24RWO",
"pid": "afn6mzM0VjJ8d0wcunu0tgXiJlx24RWO"
}
```

Response 200 Example: success not providing ID on request

```
{
"code": 200,
"msg": "Pid submitted properly, id: id1",
"pid": "id1"
}
```

Response 409 Example: when ID already in use

```
{
"code": 409,
"msg": "A process is already submitted with the same id, id: \"id1\". Please try
 with another id or set it as null or \"None\" and the platform will autogenerat
e one for you",
"pid": "id1"
}
```

9.2.3 Get list of process status

Return all the process information that match with the filter parameters.

Operation	**GET**	
Endpoint	/api/process/list	
Header	None	
Content-Type	application/json	
Parameter	None	
Body	{} or ListFilterDictionary	Filtering object.
Responces	{"result":List<ProcessStatusReport>}	List of process state resumes

Table 4 Process Status List API endpoint description

9.2.4 Get logs

Return the logs file content related to the ID provided.

Operation	**GET**
Endpoint	/api/process/{pid}/logs
Header	None
Content-Type	None

Parameter	last	Int Opt.	This parameter will determine the last number of lines that will be returned. If not provided it will return all the lines from the logs file.
Body	None		
Responses	text		The response is raw text extracted from the id.logs file.

Table 5 Get logs API endpoint description

RESPONSES

Response 200 example

```
ProcessRunner initialization start
Initialising state...
Trying to load ProcessState for pid:id1 from persistence folder
Can't load ProcessState for pid:id1 doesn't exist in persistence folder
State Initialised, initial ProcessState: {'pid': 'id1', '_status': 'PENDING',
'_submissionDate': '', '_executionStartDate': '', '_executionEndDate': '',
'source': None, 'pipeline': None, 'sink': None, 'config': {'id': 'id1',
'pipeline': [{'config': {}, 'logger': 'Not printing'}], 'sink': {'cls':
'sms_analytics.test.factorial_pipeline.TestSink', 'config': {}, 'logger': 'Not
printing'}, 'source': {'cls':
'sms_analytics.test.factorial_pipeline.SequentialNumberGenerator', 'config':
{'limit_number': 3, 'start_number': 1}, 'logger': 'Not printing'}, 'type':
'PUNCTUAL'}}
Initialising process objects:
Initialising source...
Source Initialised, state after source creation:

{
    "last_generated_value": 1,
    "last_commited_value": 1
}
Initialising pipeline...
Init Pipeline from config
Initializing all the steps of the pipeline
All steps initialized successfully
Pipeline initialised
Initialising sink...
...
...
...
```

9.2.5 Get notifications

Return the notifications related to the ID provided. Two endpoints exist, one that return all the notifications (/notifications) and a second one that returns and removes the notifications (/notifications_pop) from the process.

Operation	**GET**	
Endpoint	/api/process/{pid}/notifications	/api/process/{pid}/pop_notifications

Header	None	
Content-Type	None	
Parameter	None	
Body	None	
Responses	{"notifications": List<ProcessNotification>}	The response is a dictionary with all the notifications.

Table 6 Get notifications API endpoint description

RESPONSES

Notifications

```
{
    "notifications": [
        {
            "level": "ERROR",
            "msg": "Step 2 detected 5 errors",
            "timestamp": 1716450651.710516
        },
        {
            "level": "INFO",
            "msg": "id "DQWEEQWEWQE" processed",
            "timestamp": 1716450651.713973
        }
    ]
}
```

Response 409


```
{
"code": 409,
"msg": "A process is already submitted with the same id, id: \"id1\". Please try
 with another id or set it as null or \"None\" and the platform will autogenerat
e one for you",
"pid": "id1"
}
```


Response 200 Example: no notifications


```
{
    "notifications": []
}
```

9.2.6 Get status

Return a report of the process related to the ID sent on the URI, this report will contain useful information of all parts of the process and some information about the current state.

Operation	**GET**		
Endpoint	/api/process/{pid}		
Header	None		
Parameter	return_structure	Bool Opt..	Type bool, default True, can be set to false if want to avoid the "structure" attribute content, that will be return with a None.
Body	None		
Responses	-	ProcessStatusReport	Process state resume

Table 7 Get Process Status API endpoint description

9.2.7 Get process result

Get process result.

Operation	**GET**
Endpoint	/api/process/{pid}/result
Header	None
Parameter	None

Body	None		
Responses	-	Text (depending of Sink used)	Process result

Table 8 Get Process Result API endpoint description

RESPONSES

The response is raw text extracted from the result file.

Response 200 Example: successful return without extra_info

```
{
    "files": [
        "standard_result.csv",
        "bucket_result.csv"
    ],
    "pid": "dIM6QofuA3Fh7v6nkf2wQC6Pd2SBK3mL"
}
```

RESPONSES

Response 200 Example: successful return of the file

```
File content....
...
...
...
```

Response 400 Example: id not found

```
{
    "error": "The provided pid doesn’t exist"
}
```

Response 400 Example: process fail

```
{
    "error": "The process related to this id end with FAIL status"
}
```

Response 400 Example: no file result

```
{
    "error": "The process related to this id have not a file as result"
}
```

Response 500 Example: can’t read the result file

```
{
    "code": 500,
    "pid": "id",
    "msg": "Process end successful but we can’t read the result file "
}
```

9.2.8 Kill process

Operation	**DELETE**		
Endpoint	/api/process/{pid}		
Header	None		
Parameter	None		
Body	None		
Responses	-	{“msg”: {string}}	Dictionary with status msg.

Table 9 Delete Process API endpoint description

RESPONSES

Response 200 Example: process deleted

```
{
    "msg": "Process deleted"
}
```

Response 500 Example: Some Error while trying to delete process

```
{
    "msg": " Some Error raised while trying to delete process files"
}
```

10. User Interface

This chapter outlines and guides the co-design process of the RECREATE_WT visualisation tool, ensuring it effectively incorporates insights from the Working Group (WG) exchanges conducted under WP2. The objective is to define the appropriate level of information detail and identify the most effective visualisation types to support data-driven decision-making in water governance. By prioritising usability and accessibility, the platform will empower stakeholders with intuitive tools for interpreting complex data and selecting strategic water management options. The task just commenced in March 2025 (M15) and will evolve through an iterative development process until its completion in June 2027 (M42).

The stakeholder engagement process in WP2 was structured into three phases to support the establishment of Communities of Practice (CoP) and WGs across all CS. Initial efforts focused on mapping and identifying key stakeholders (M10 - M14), followed by a rapid analysis of their influence and interest to inform WG formation (M13 - M15). Concurrently, activities were contextualized within local spatial policy frameworks to ensure relevance and alignment. Early CoP meetings were held in Kalundborg and North Holland, with upcoming sessions in Costa Brava and Syros. This process lays the groundwork for inclusive co-design and capacity building aligned with regional priorities.

Once all CoPs are established and the WG are fully operational, the project can initiate the ad-hoc WG focused on the user interface (UI, see FRONT-END Layer of Figure 4: RECREATE_WT architecture) design of the RECREATE_WT tool. This group will drive the co-creation of the visualisation environment in Task 3.1 by incorporating user needs and expert feedback into the platform's development.

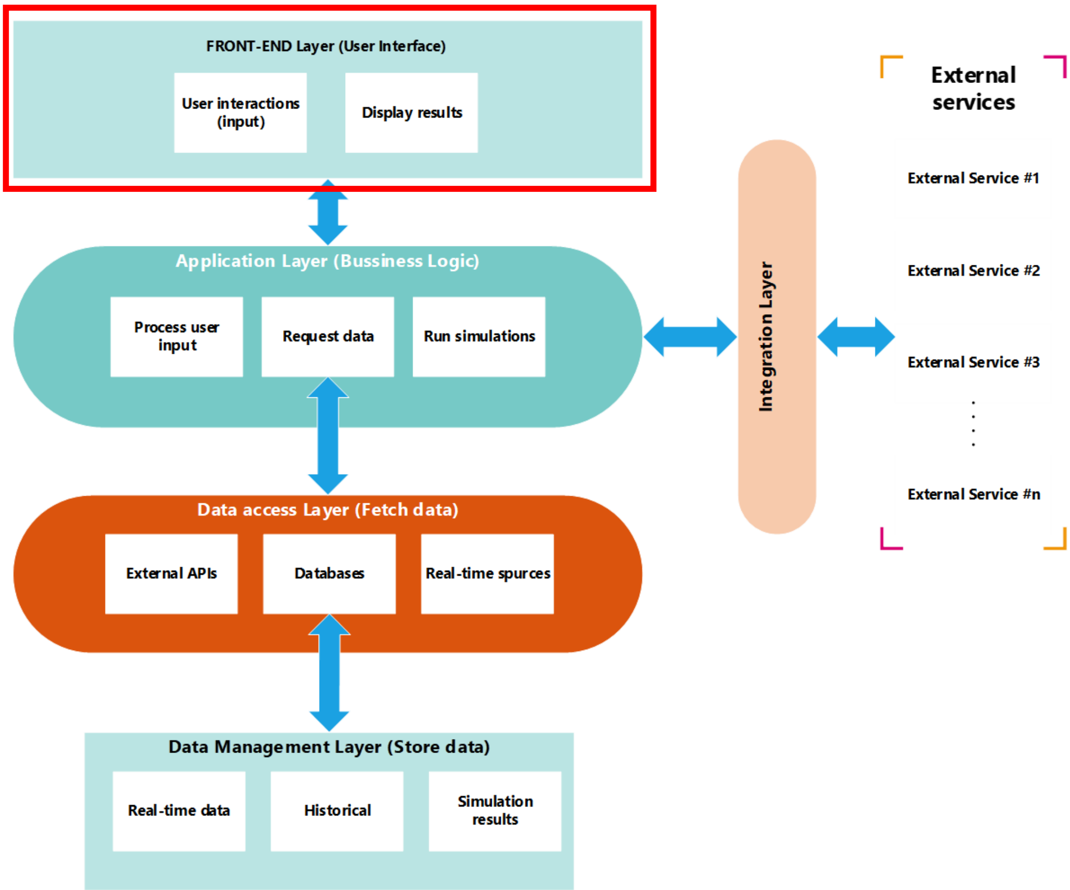

Figure 4: RECREATE_WT architecture

The first UI workshop, was scheduled for the end of May 2025 (M17), was 60-minute collaborative session led by ICLEI with support from EUT. It engaged eight participants, including internal and external experts from the case study regions and CoPs. The workshop's objective was to gather structured input on the required information levels and preferred visualisation types, ultimately enhancing the platform's usability and decision-support functionality.

As this is an iterative process continuing until June 2027 (M42), this chapter will be revisited and expanded upon as part of the development for Deliverables D3.2 and D3.3. Insights gathered through the user interface WG and its workshops will inform ongoing refinements to the RECREATE_WT tool, ensuring the platform evolves in alignment with stakeholder feedback and project goals for interoperability, usability, and decision support.

10.1 Review of Best Practices in Data Visualization for Water Management

Effective data visualization in water management plays a critical role in turning complex datasets into actionable insights for policymakers, utility operators, and community stakeholders. Based on recent literature, platform reviews, and successful implementations in the water sector, the following best practices have emerged:

10.1.1 User-Centric Design

- Tailor visualizations to the needs, roles, and data literacy levels of different users (e.g; water managers vs. policymakers vs. the public).
- Use user stories and personas to guide which information is shown, how it is presented, and in what level of detail.

10.1.2 Dashboard-Based Interfaces

- Combine multiple views (e.g; real-time data, forecast models, KPIs) into a single dashboard to support holistic decision-making.
- Ensure modularity: allow users to configure views based on their role or area of responsibility.

10.1.3 Clarity and Simplicity

- Prioritize clarity by avoiding visual clutter and selecting intuitive chart types (e.g; line charts for time trends, bar charts for comparison, heatmaps for intensity).
- Use consistent scales, units, and colors across views to avoid misinterpretation.

10.1.4 Temporal and Spatial Context

- Integrate time-series and geospatial visualizations to show when and where events occur (e.g; water quality fluctuations, reservoir levels, leakages).
- Tools like Leaflet, OpenLayers, or Mapbox are commonly used for map-based interactions.

10.1.5 Forecasting and Scenario Tools

- Visualize both real-time and projected data to support both short-term operations and long-term planning.
- Scenario sliders or toggles help users interactively explore different climate or demand futures.

10.1.6 Alerts and Thresholds

- Use visual indicators (e.g., red-yellow-green color coding, warning icons) to signal when thresholds are breached (e.g., low reserves, high demand).
- Allow users to set customizable thresholds for automated notifications.

10.1.7 Interactivity and Drill-Down

- Allow users to interact with data through tooltips, filtering, zooming, and drill-down capabilities.
- Enable users to download reports or data extracts directly from the interface for deeper offline analysis.

10.1.8 Open Standards and Accessibility

- Apply web standards (e.g; WCAG) to ensure accessibility for all users.
- Use open-source tools (e.g; D3.js, Highcharts, Plotly) and standardised APIs for interoperability and future expansion.

10.1.9 Real-Time Integration

- Where possible, integrate real-time data from sensors, SCADA systems, or IoT platforms for timely insights.
- Ensure backend systems can handle the performance, and latency demands of real-time dashboards.

10.1.10 Evaluation and Iteration

- Conduct regular usability testing with stakeholders (CoP, CS leads) to refine dashboard design and functionality.
- Monitor platform usage and adapt visuals based on real-world feedback and evolving user needs.

10.2 Dashboards

The RECREATE_WT platform will feature a standardized, user-friendly visualisation environment designed to support effective water governance. At its core are interactive dashboards that provide stakeholders with comprehensive access to:

1. Key Performance Indicators (KPIs),
2. Analytical and advisory services derived from raw data,
3. Economic, legislative, and risk-related information stored in the database, and
4. Dedicated interfaces for configuring and fine-tuning various decision-support tools.

This visual framework will enhance both the usability and accessibility of the platform. By adopting open-source technologies and leveraging free, standardized interfaces, the dashboards will enable transparent, data-driven decision-making and improved water consumption management across diverse user groups.

The following chapter will summarize the outcomes of the user stories analysis, which directly inform the design of these dashboards and the overall user interface. User stories are concise, user-centric descriptions of specific features or functionalities from the perspective of stakeholders. They capture the goals, needs, and actions of users, offering a clear understanding of expected outcomes. This structured overview of user stories serves as the foundation for designing the "white boxes" of the RECREATE_WT tool, as illustrated in Figure 4: RECREATE_WT architecture.

10.3 Required level of information

As part of the user interface design process in Task 3.1, this section defines the required level of information to be presented in the RECREATE_WT tool, ensuring clarity, usability, and relevance for diverse stakeholder needs. To support this, a series of structured User Stories has been created. These concise, goal-oriented descriptions capture specific features from the perspective of various user types, outlining what they aim to achieve and how they will interact with the platform. The interface development follows a phased approach: first, the co-design of low-fidelity prototypes with input from the CS regions, followed by the definition of visual identity elements such as logos, colour schemes, symbols, and typography.

These elements are then integrated into high-fidelity prototypes representing the final user experience. Upon completion, the design assets and specifications will be handed over to the development team, who will implement the operational interface and conduct all phases of testing – from early prototype validation to full tool deployment – ensuring technical alignment with the intended user experience.

To ensure a seamless transition from co-design to implementation, the defined user stories help establish the required level of information for each functional module of the RECREATE_WT platform. These narratives describe the types of data users expect (e.g. water quality, energy consumption, scenario outputs), the temporal context (real-time vs. long-term), and the necessary decision-support outputs (e.g. alerts, forecasts, dashboards). This structured understanding is essential to inform developers about what data must be collected, processed, and visualised - ultimately shaping the backend data architecture, real-time processing needs, and front-end interface components. By capturing these requirements early, the RECREATE_WT tool will remain aligned with stakeholder needs while ensuring technical feasibility and scalability.

Location	Title	User Type	Purpose / Need	Expected Outcome	Dependencies	Acceptance Criteria	Priority
North Holland	System Dynamics Model (SDM)	Water personnel, research-ers	Run scenario analyses on AWR options under socio-economic and climate changes	Insights on resilience and impact of AWR strategies	Water demand scenarios, chloride inputs	Standalone VENSIM model; ideally visual dashboard	Low
North Holland	Serious Game	Policy-makers, water personnel	Multi-user policy simulation of inter-ventions	Understand effects of AWR policies; improve collabora-tion	Platform access, multiplayer setup	User-friendly online or in-person game interface	Medium
Kalund-borg	Climate Scenario Simulation	Strategic planner	Assess future water	Adaptive water strategy	Climate/environ mental data	User-friendly tool in English;	Medium

			resources for planning	under climate/demand change		continuous use	
Kalund-borg	Evaluation of Treatment Trains	Strategic planner	Compare treatment options for future scenarios	Optimal reuse solution based on cost, perfor-mance	Innovation data, treatment defaults	Default values; English interface; easy to use	Medium
Syros	Real-Time Water Availability	Water manager	Monitor reserves and detect leakages	Minimise water loss; timely supply redirection	IoT sensors, real-time processing	Real-time alerts, leakage detection, operational insights	Low
Syros	Real-Time Energy Consumption	Water manager	Track energy use vs. water output	Optimize system sustainability and efficiency	Smart power meters	Real-time monitoring and system under-standing	Low
Syros	Near-Future Climate Scenarios	Water manager	Plan desalination and reserves based on forecasts	Manage production based on seasonal/weather projections	Climate/ weather data	Seasonal and short-term clarity, strategic usability	Medium
Costa Brava	Water Availability & Quality	Water manager	Monitor system status and trigger alerts	Emergency response, reduce losses, ensure safety	Flowmeters, water sensors	Accurate dual monitoring, real-time processing	High
Costa Brava	Reuse Water Treatment	Treatment plant operator	Ensure water meets reuse standards	Safe water reuse for indirect potable use	Quality sensors, pathogen analysis	Continuous water quality monitoring	—
Costa Brava	Near-Future Climate Scenarios	Water manager	Plan production and infrastructure needs	Improved resource and reserve planning	Climate/ weather data	Accurate forecast use; decision-support interface	—
Costa Brava	Energy Consump-tion	Water manager	Monitor energy use across water systems	Support infrastructure upgrade and sustain-ability	Smart meters	Energy-water data integration; improve efficiency	—

Table 10 Summary of User Stories Across Case Studies

Find below in Table 11the synthesized User Stories in a structured matrix across four regions which summarizes core insights. The first two entries in bold letters have been taken up upon and are described in paragraph “0

Technical Justification" as crucial functions for two high priority tools, the Real-Time Monitoring Dashboard and the Decision Support System for AWR Management.

Region	User Type	Desired Functionality	Data Dependency	Priority
North Holland	**Policy Advisor**	**Adaptive planning tools**	**Climate projections, DEMs**	**High**
Costa Brava	**Water Manager**	**Real-time drought forecasting**	**Live weather, reservoir**	**High**
Kalundborg	Infrastructure Eng.	Risk assessment visualizations	Coastal elevation, rainfall	Medium
Syros	Planner	Scenario-based land use simulation	Socioeconomic and land use	Medium

Table 11 Summary of core insights of User Stories

Across the regions, several common themes emerged regarding user needs. Policymakers expressed a preference for simplified visualizations that clearly communicate risks and recommended actions. Technical users, on the other hand, emphasized the importance of having the ability to drill down into model assumptions and underlying data layers. Meanwhile, water managers prioritized access to real-time alerts and intuitive trend visualizations. These diverse needs informed key design choices, such as implementing modular dashboards, offering dual-view options for both basic and advanced users, and integrating context-aware tooltips to enhance clarity and usability.

10.4 Types of visualisations

To support data interpretation and user interaction, the RECREATE_WT platform will offer a variety of visualisation types tailored to different stakeholder needs and use cases. These visual components are essential to communicate analytical outcomes, support decision-making, and enhance understanding of system behaviours under various scenarios. Below, we describe the main types of visualisations that the platform will provide.

10.4.1 First design concept: Water Management and Adaptation Pathways

The initial draft in Figma is structured around two complementary modules that support strategic and operational decision-making in water resource management: long-term adaptation and short-term management. These modules are designed to enhance the RECREATE_WT platform's capacity to address both future planning and real-time operational needs. Please find below in Figure 5 an indicative screenshot of the UI screen after logging in.

Figure 5: Figma Prototype – Screen following a log-in with short-term and long-term option

1. Long-Term Adaptation Module

The Long-Term Adaptation module is focused on supporting strategic water resource planning over a 30-year horizon. It enables users to assess the impact of varying climate change scenarios on water availability and to explore adaptive strategies accordingly. This forward-looking functionality is intended to guide infrastructure investment and policy development. Please find below in Figure 7 an indicative screenshot of the UI screen.

Key Features:

- Climate Scenarios: Visual and textual explanations of multiple climate trajectories, ranging from pessimistic to optimistic, allowing users to understand the range of potential environmental futures.
- Water Supply Forecasting: Long-term projections of groundwater and surface water availability under each selected climate scenario.
- Adaptive Pathways: Decision-support outputs suggesting infrastructure and management strategies that align with projected water conditions.
- Consumption Trend Analysis: Forecasted patterns of water use to support planning for sustainable consumption and demand management.

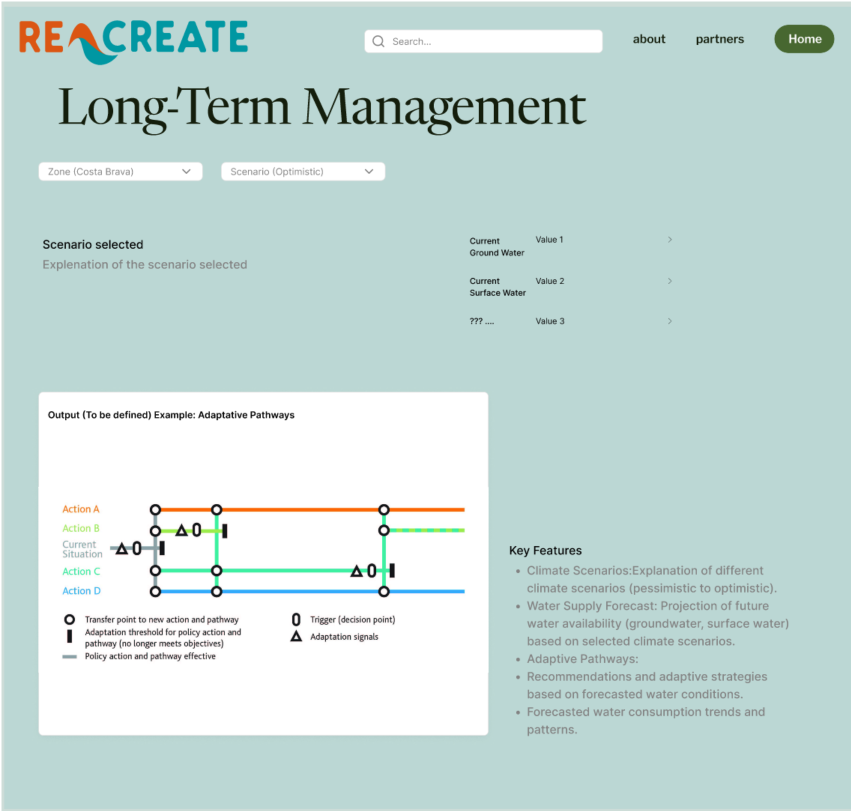

Figure 6: Figma Prototype - Screen option long-term management.

2. Short-Term Management Module

The Short-Term Management module is tailored for operational decision-making within a 15-day window. By integrating real-time meteorological data and water system metrics, this tool supports dynamic resource allocation and crisis response planning across different economic sectors. Please find below in Figure 7 an indicative screenshot of the UI screen.

Key Features:

- Weather Forecast Integration: Continuous access to 15-day forecasts for key parameters like temperature, precipitation, and humidity.
- Real-Time Water Availability Analysis: Live data on groundwater and surface water levels, including quality assessments, enabling informed day-to-day decisions.
- Resource Optimization Tools: Scenario simulation engines that help predict short-term water use and recommend efficient, sector-specific allocation strategies.

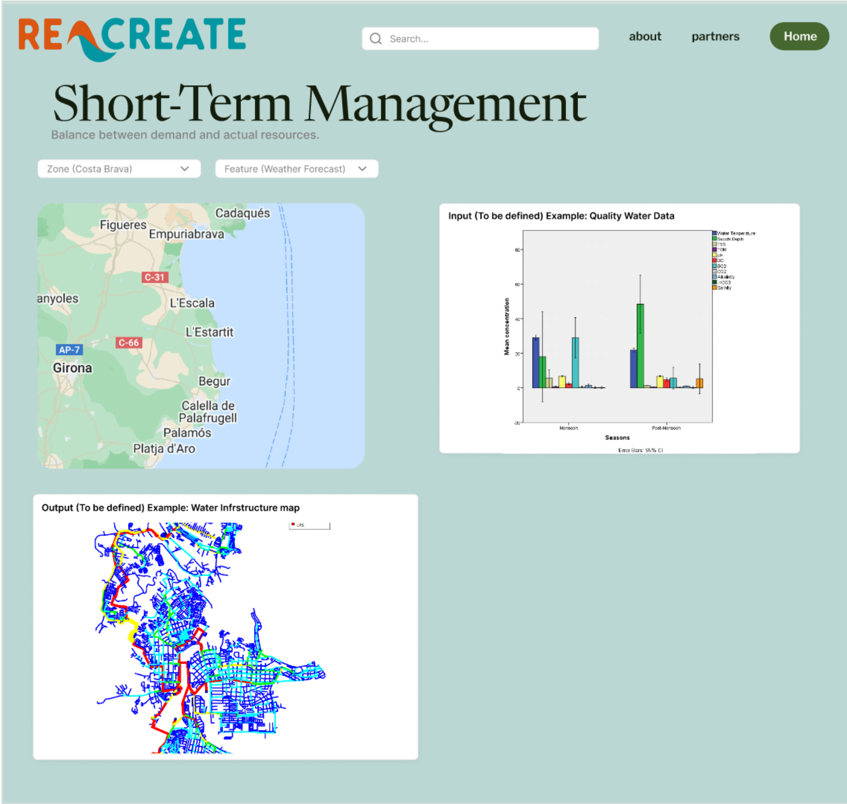

Figure 7: Figma Prototype - Screen option short-term managemet.

This dual-pathway approach ensures that RECREATE_WT addresses both the long-term challenges of climate resilience and the immediate demands of daily water resource management. The Figma draft visually reflects this architecture, offering a clear, user-centric interface that supports intuitive navigation and action-oriented insights.

A possible third module - Mid-Term Management - may be added to bridge the gap between operational response and strategic foresight. This option could include monthly or seasonal planning tools, such as water allocation simulations or drought preparedness measures.

10.4.1.1 Evolution and development through CoP workshop integration

A dedicated workshop with the CoP will play a central role in evolving and refining this conceptual framework. The workshop will gather insights, priorities, and use-case feedback from regional stakeholders to ensure the design meets practical needs and fosters co-created solutions. These collaborative sessions will guide the iterative development of both modules, ensuring they remain user-focused and regionally adaptable. For example, please find below in Figure 8, Figure 9, and Figure 10 first screenshot of the Alpha-version UI screen to run different jobs against two scenarios.

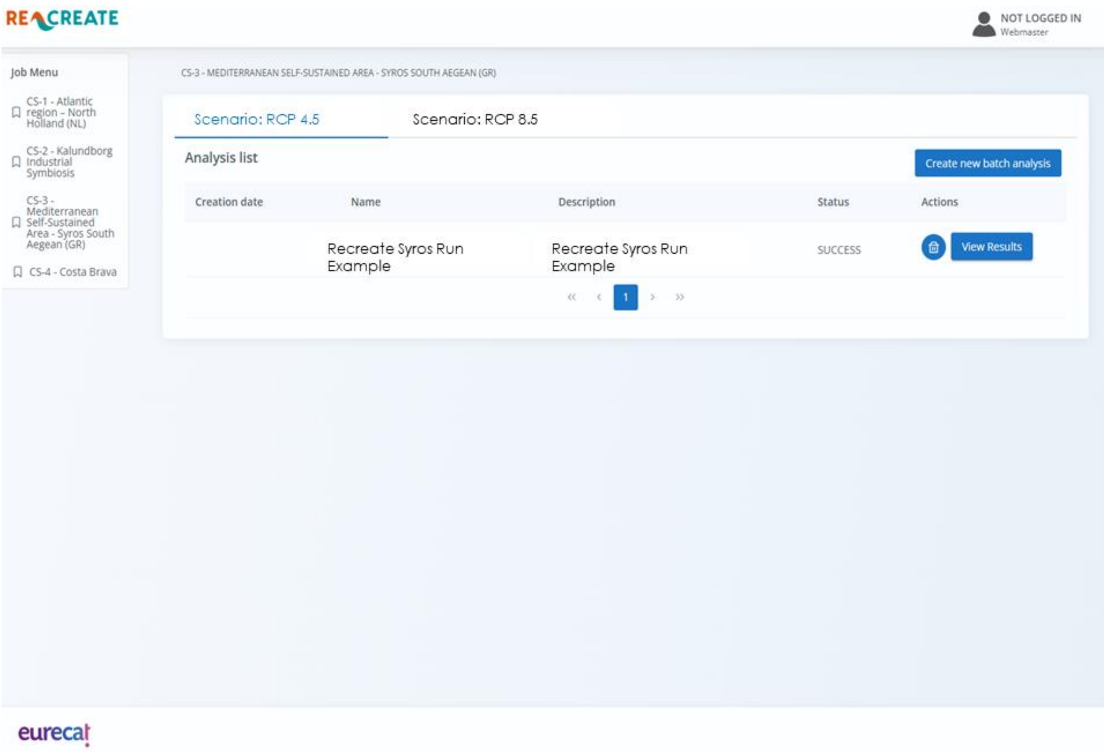

Figure 8: First Alpha-version of UI screen development - scenario selection

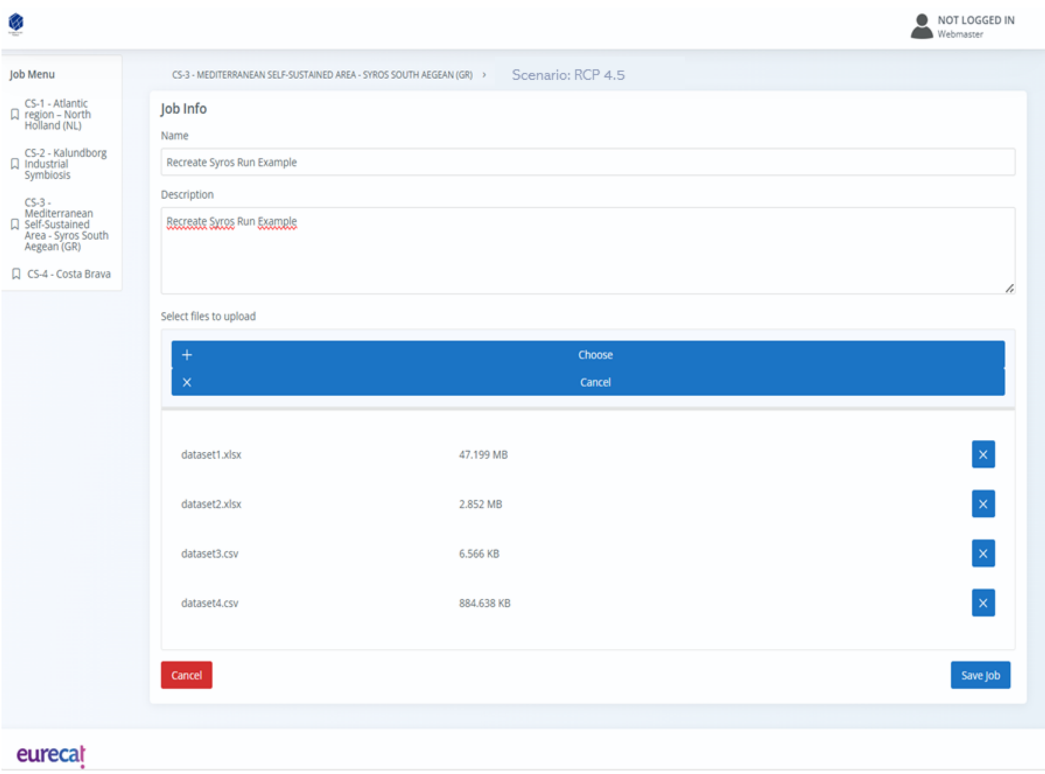

Figure 9: First Alpha-version of UI screen development - job creation

Figure 10: First Alpha-version of UI screen development - presentation of results

10.4.1.2 Further Prototype Concepts

The initial prototype was developed using Figma and demonstrates a dual-module layout corresponding to long-term and short-term planning functions. Key features include an interactive navigation structure, map-based dashboards overlaying key climate and hydrological indicators, and scenario comparison panels with sliders and dropdown filters. Real-time widgets allow filtering by time period, spatial extent, and system component, while alert mockups illustrate drought and flood notifications. Feedback from early presentations emphasized the need for customizable colour schemes, clear legends, and downloadable summaries, all of which are under iterative development.

10.4.1.3 Integration with Communities of Practice

The regional CoPs play a central role in shaping the interface design. While the CoP workshops have not yet been conducted, their preparation is underway, and they are scheduled for summer 2025. Feedback from upcoming stakeholder sessions will be used to further refine and validate interface components. This user-centred, iterative approach ensures that the UI remains aligned with regional needs and stakeholder expectations as it evolves.

10.4.1.4 Outlook and Action Plan

Task 3.1 officially began in March 2025 and will continue iteratively until the project concludes in 2027. In the immediate term, the focus lies on finalizing the analysis of user stories and consolidating findings into the D3.1 deliverable. Although the draft and adaptation of a new prototype UI could not be completed for inclusion in D3.1, it remains a high priority ahead of the first interactive workshop.

Looking ahead:

- **May**: Complete and internally review the D3.1 report. The document must be finalized by the end of May, with internal review starting in early June.
- **June**: Plan and announce the first interactive meeting with CS leads. Prepare and send invitations for this internal workshop and the subsequent four CoP workshops. Begin a new round of prototype development. Announce the CS meeting at the WP5 meeting.

- **July**: Host the first CS leads workshop to reflect on the UI prototype and gather input. Document stakeholder activities and complete participation records within two weeks post-event.
- **August**: Conduct the four CoP stakeholder workshops. Ensure all engagement activities are documented, with participation lists submitted no later than two weeks after each workshop.

This timeline ensures that iterative UI design and stakeholder validation proceed in alignment with technical development milestones, supporting the creation of a responsive and user-oriented interface for the RECREATE_WT tool.

ANNEX 1: User Stories and Use Cases

This document aims to outline the functionalities that will be implemented in the RECREATE platform for all CS. It serves as a precursor to the front-end design and provides a comprehensive overview of the back-end requirements. The process will involve multiple iterations to ensure an optimal design that considers the feasibility of each functionality according to the available technologies, data, models and needs of the final users.

This document will focus on the requirements for each specific CS but will also be used to detect commonalities between them in order to optimize the design and development of each functionality.

Each CS provides a detailed analysis based on real-world usage and experiences, offering valuable insights into how the platform performs in various contexts (socio-economic context, climate projections, user profiles, etc.).

1. Document overview and instructions

This document will focus on the User Stories which are concise, customer-centric description of a specific feature or functionality from the perspective of a user or stakeholder. The emphasis is on the user's goals, needs, and actions, providing a clear understanding of what the user expects to accomplish. These User Stories are meant to be the first step to design the white boxes in the RECREATE_WT tool described in Figure 11.

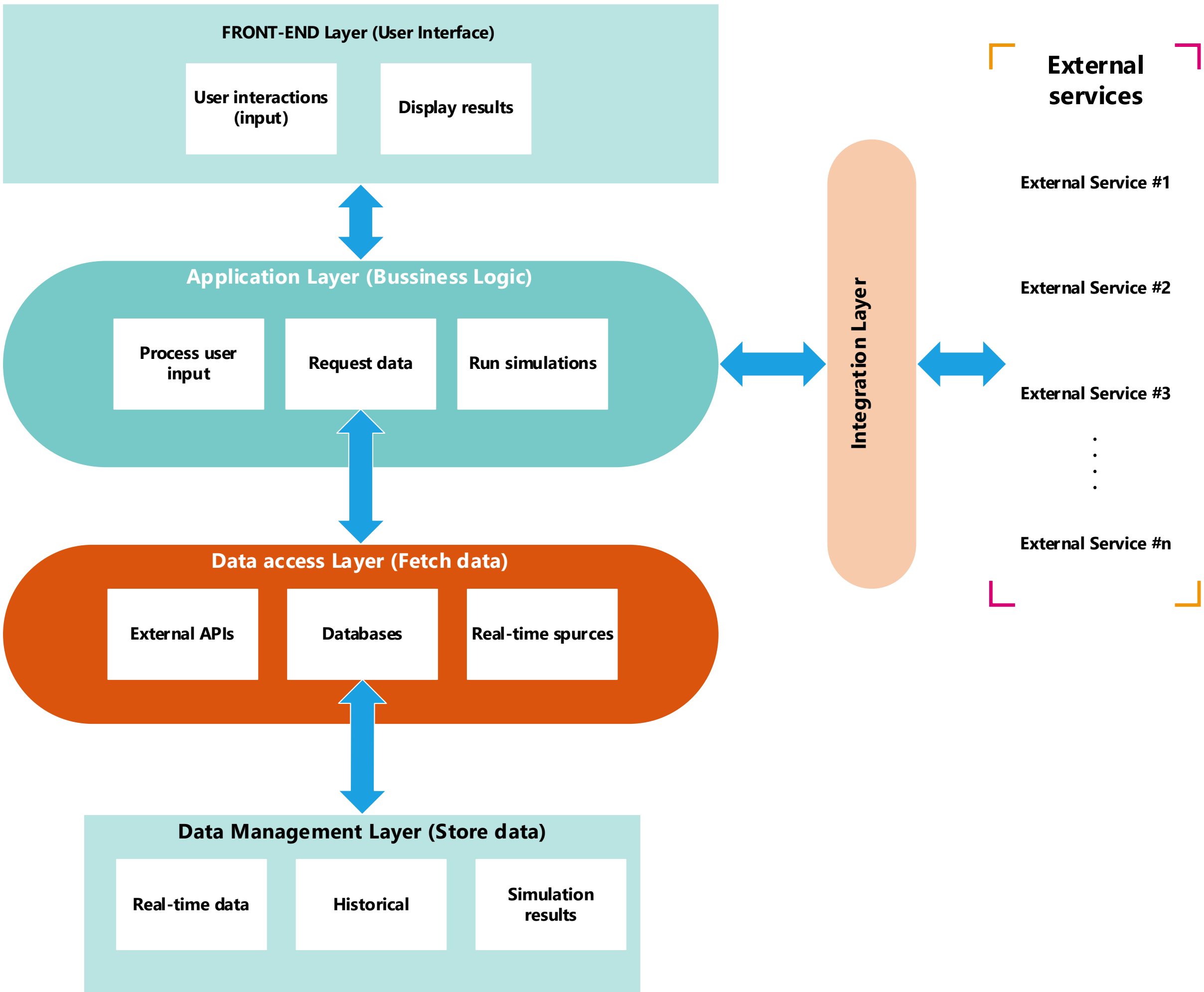

Figure 11 RECREATE_WT architecture

Each CS leader will be responsible to define one user story for each of the specific functionalities related to their CS by fulfilling a template table provided in Section 5.
An exhaustive analysis of the User Stories will be performed by Task 3.2 and discussed with all the actors involved in the definition and implementation of these functionalities to ensure their feasibility and alignment with the Grant agreement Section 6 will discuss all technical requirements in terms of

available data and back-end, and Section 7 will discuss all technical requirements in terms of front-end requirements to be implemented in the RECREATE_WT.

The following steps explain the overall process proposed to design the platform and the role of the definition of the user stories.

2. Steps for Designing the Platform

Step 1: General Requirements

- Identify technical constraints or dependencies.
- Determine data needs.
- Define all types of user profiles.
- Outline user journeys (what specific users want to achieve with the tool).
- Design the architecture:
  - Logical architecture
  - Back-end services to be used by the front end
  - Front-end architecture
  - Infrastructure
  - Technical choices (e.g., Docker, Angular)

Step 2: Analyze Previous Prototypes or Existing Similar Tools (if available).

Step 3: Functional Requirements (Continuous Iteration with Step 3.5) (Cooperation with CSs)

- Identify epics (thematic groups of user stories based on the user journey).
- Complete user stories tables.
- **Review General Requirements (Continuous Iteration with Step 3)**
  Redefine technical constraints or dependencies.
  Review the feasibility of user stories and alignment with available data.

Step 4: Non-Functional Requirements

- Consider aspects like responsiveness, language, security, and performance.

Step 5: User Interface Design (Cooperation with CSs)

- Create low-fidelity prototypes (initial wireframes without design).
- Define branding and design elements (logos, colours, symbols, typography).
- Develop high-fidelity prototypes (detailed prototypes including visual design).
- Hand over to developers.

Step 6: Testing

- Test low-fidelity prototypes.
- Test high-fidelity prototypes.
- Test the operational tool.

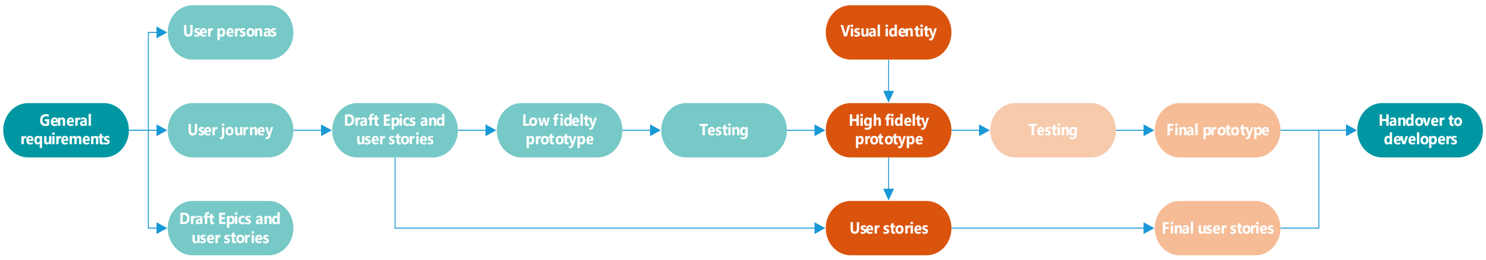

Figure 12 Design process flow

3. User Personas

It is essential to consider the user profiles that will utilize each functionality. While any user might have access to any functionality, it is crucial to ensure that each one is designed to meet the specific requirements of the primary user. For instance, tools intended for scientists will likely require access to more detailed data than those intended for citizen education. If a functionality is designed with two layers of complexity for two types of users, this should be stated in the acceptance criteria section of the user story.

Based on the description provided in the existing documentation, the following is a list of potential user profiles for the RECREATE_WT tool, considering the different layers of complexity and primary user needs:

Water Management Professionals:

- **Primary Users:** Engineers, planners, and utility managers responsible for water distribution and treatment.
- **Needs:** Access to real-time and historical water data, simulation tools, and decision-support frameworks for planning and optimizing water resources.

Scientists and Researchers:

- **Primary Users:** Environmental scientists, hydrologists, and researchers involved in studying water systems and climate impact.
- **Needs:** Detailed, complex datasets for climate projections, water quality analysis, and advanced simulation models for research and experimentation.

Policy Makers and Regulators:

- **Primary Users:** Government officials, EU representatives, and local authorities involved in setting policies for water management.
- **Needs:** Simplified dashboards that provide key insights, policy impact analysis, and scenario planning tools for decision-making and regulatory compliance.

Environmentalists and NGOs:

- **Primary Users:** Environmental advocates and organizations focused on sustainable water management and conservation efforts.
- **Needs:** Access to environmental impact data, AWR adoption rates, and visualization tools that highlight sustainability metrics.

Local Communities and Citizens:

- **Primary Users:** General public, community leaders, and citizens interested in understanding local water issues and contributing to water conservation.
- **Needs:** Easy-to-use interfaces that provide simplified information about water quality, availability, and conservation tips.

Data Analysts:

- **Primary Users:** Data scientists and analysts involved in processing and interpreting large datasets from water monitoring systems.
- **Needs:** Access to raw data, analytical tools, and customizable dashboards for generating reports and insights.

Infrastructure Developers and Technologists:

- **Primary Users:** Developers working on digital infrastructure for water management and integration of IoT technologies.
- **Needs:** Integration tools, API access, and technical documentation for building and maintaining digital water management systems.

Health and Safety Experts:

- **Primary Users:** Public health officials and safety experts monitoring the impact of water quality on human health.
- **Needs:** Tools for health risk assessment, real-time monitoring of water contaminants, and alerts for public safety issues.

Each of these user personas can be interested to interact with the RECREATE_WT tool at different levels of complexity based on their specific needs, ensuring that functionalities are tailored to provide the necessary insights and usability for each profile.

4. Common functionalities: similar User Stories across different case studies

This section lists the common tools and functionalities that have been identified across the CS of the RECREATE project. These tools serve multiple regions and address the shared needs of different user personas. The goal is to ensure consistent functionality while allowing flexibility for specific regional adjustments where necessary.

First approach of the common functionalities and expected tools based on existing documentation:

Title	**Real-Time Monitoring Dashboard**	**Priority**	High
General info	A dashboard that provides real-time data on water quality and quantity, including alerts for threshold breaches. Integrated with IoT sensors for live updates across different CS regions.		
As	A water manager	**Size/effort**	Large (due to real-time data integration and customization per region)
I need to	Monitor water quality and quantity in real-time, and receive alerts when thresholds are breached.	**Dependency**	IoT sensors, data integration tools, real-time processing capability
So that	I can make timely decisions for water distribution and optimize resource allocation.	**User story ID**	US1.1
Case studies	Costa Brava (Spain), Syros (Greece), Kalundborg (Denmark), North Holland (The Netherlands)		
Common Functionalities	• Data integration from IoT sensors across different cities. • Real-time updates and alerts for threshold breaches. • Consistent UI/UX for ease of use in all locations.		
Acceptance criteria	• The system offers real-time data for water quality and quantity. • Configurable alert mechanisms for threshold breaches are provided. • Dashboard should be user-friendly, responsive, and accessible across devices.		

Title	**Decision Support System for AWR Management**	**Priority**	High
General info	A tool that simulates the long-term effects of using Alternative Water Resources (AWR) for policy planning. It helps policy makers evaluate sustainable water management strategies.		
As	A policy maker	**Size/effort**	Medium (due to simulation modeling and data processing requirements)
I need to	Simulate the effects of AWR implementation on water management to ensure sustainable policies.	**Dependency**	Climate models, water usage projections, scenario analysis tools
So that	I can make data-driven decisions on water management and formulate effective, long-term policies.	**User story ID**	US2.1
Case studies	Costa Brava (Spain), Syros (Greece), Kalundborg (Denmark)), North Holland (Netherlands)		
Common Functionalities	• The system integrates data from climate models and water usage projections across regions. • Scenario analysis tools for simulating the impact of AWR on long-term water supply and demand. • Consistent UI/UX to ensure ease of use for policy makers.		
Acceptance criteria	• The system provides accurate simulations of AWR effects on water supply and demand. • Enables scenario-based analysis and long-term projections. • User-friendly and intuitive interface, accessible to non-technical users.		

4.1 Action plan for finalizing the section

- **Conduct Workshops:** Organize workshops with stakeholders from each CS (Kalundborg, Costa Brava, the Netherlands, Syros) to confirm and refine the identified common functionalities.

- **Iterative Refinement:** Continue to update this section as new user stories emerge or are refined.

- **Align with Development:** Ensure the development team focuses on creating reusable and scalable solutions for the common functionalities across these CS areas.

5. Specific functionalities: Case Studies User Stories

This section is designed to guide the development of tools and functionalities for the RECREATE_WT platform by outlining specific user stories based on the needs of stakeholders across different CS. Each tool or functionality must be tailored to meet the requirements of the primary users involved in water and AWR management integration. To ensure that the platform delivers effective and usable solutions, the user stories tables will be filled out based on the following key and required fields:

Title

- **Description**: This field represents the name of the tool or functionality being discussed. It should briefly describe the purpose or primary feature of the tool.
- **How to Complete**: Provide the name of the specific tool or dashboard, such as "Real-Time Monitoring Dashboard" or "Decision Support System for AWR Management."

General Info

- **Description**: A brief explanation of what the tool or functionality does, including its primary purpose and how it helps the user.
- **How to Complete**: Summarize the main features of the tool, highlighting what kind of data or functionality it provides and why it is useful for the user. Example: "A dashboard that provides real-time data on water quality and quantity, including alerts for threshold breaches."

As

- **Description**: The user or persona who will primarily use the tool.
- **How to Complete**: Describe the primary user of the tool, such as "A water manager," "A policy maker," "A researcher," or other relevant stakeholders.

I Need To

- **Description**: Describes the specific action or goal that the user needs to accomplish using the tool.
- **How to Complete**: Clearly describe what the user wants to achieve with the tool. Example: "Monitor water quality and quantity in real-time."

Dependency

- **Description**: Lists any external systems, data sources, or technologies that the tool relies on to function.
- **How to Complete**: Identify the systems or data integrations required for the tool to operate. For example, "IoT sensors, real-time data processing capabilities, environmental models."

So That

- **Description**: Explains the user’s motivation or the reason why the action (from "I Need To") is necessary.
- **How to Complete**: Clearly state why the user wants to achieve their goal. Example: "So that I can optimize water distribution and prevent shortages."

Acceptance Criteria

- **Description**: Specific conditions or requirements that must be met for the tool to be considered complete and usable.
- **How to Complete**: List the criteria the tool must meet, such as "The system provides real-time data," "The system allows for threshold breach alerts," or "The tool is user-friendly and available in multiple languages."

5.1 North Holland (Netherlands) – Likely focusing on integrated water management in a densely populated area, addressing innovative water management solutions.

Leader: PWN
Contact person: Koen Zuurbier
Email: koen.zuurbier@pwn.nl
Other Institutions involved: KWR

Title*	**System Dynamics Model (SDM)**	**Priority**	Low
General info*	The SDM encompasses a high level model of the drinking water system, sewage system, and service water system of North Holland.This model will provide insights into water demand, water supply, and the use of alternative drinking water souces. Scenario analysis and output from this model will support decision making and future development of the systems.		
As*	Water Personnel and researchers	**Size/effort**	
I need to*	Be able to set up scenarios of different alterative water resource options and analyse these with different socio-economic (water demand) and climate scenarios	**Dependency***	Chloride scenarios of IJsselmeer Water demand scenarios Other dependencies are programmed within the SDM.
So that*	I can compare different alternative water resource strategies, test the systems resilience, and get insights into how alternative water solutions propagate through the water system.	**User story ID**	
Acceptance criteria*	The SDM is designed as a standalone model with VENSIM software. Modellers or other technically skilled personnel can work with the model and perform scenario analysis. Ideally, we would like to have a dashboard that would be more user friendly.		
• **Long Term**			

<table>
<tr><td colspan="4">*No need, finally only ill include information, video and a link to the serious game</td></tr>
<tr><td>Title*</td><td>Serious Game</td><td>Priority</td><td>Medium</td></tr>
<tr><td>General info*</td><td colspan="3">The aim of the serious game is to help bridge the gap between science and policy by translating complex modelling results into an user friendly format. This game will be a multi-user, interactive tool which allows participants to experience the decision making process in the water sector. This game will be platform-based and powered by a high-level system dynamics model of the regional water cycle.</td></tr>
<tr><td>As*</td><td>Policy makers, water personel, (potentially general public)</td><td>Size/effort</td><td></td></tr>
<tr><td>I need to*</td><td>Be able to play with multiple players in a user-friendly environment. Have the ability to take (pre-defined) AWR interventions within the regional system and produce results quickly in a user-friendly interface.</td><td>Dependency*</td><td>Users’ ability to access operating platform (i.e. location of supporting model). User access to game platform (i.e. board game, website, download key)</td></tr>
<tr><td>So that*</td><td>We can see how different interventions effect the regional system, while learning about the aims and constraints of the different stakeholders. Ultimately, learning to work together for the development of AWR policies.</td><td>User story ID</td><td></td></tr>
<tr><td>Acceptance criteria*</td><td colspan="3">User-friendly online platform game.
Ability to play with multiple users online, but at minimum ability to play with multiple users gathered for an in-person meeting.</td></tr>
</table>

5.2 Kalundborg (Denmark) – Representing the Atlantic region, focusing on water management and treatment optimization.

Leader: Preben Thisgaard
Contact person: Preben Thisgaard and Anne Kleyböcker
Email: prth@kalfor.dk; anne.kleyboecker@kompetenz-wasser.de
Other Institutions involved: KCR, KWB

Title*	**Climate Scenario Simulation**	**Priority**	medium
General info*	The tool provides information on how the climatic and environmental conditions will change in the future. The output shall be used to adapt the water management strategy for the region around Kalundborg.		
As*	Strategic planner	**Size/effort**	
I need to*	Assess the availability of the future water resources in order to plan and design new infrastructure for water and wastewater treatment and management.	**Dependency***	Climate data, etc.
So that*	I can adapt and optimize the current water strategy. I need a tool, that can be used continuously. If the water demand of the region will change due to e.g. a further industrial expansion or the acceleration of the climate change, I want to use the tool to adapt the strategy.	**User story ID**	Not known
Acceptance criteria*	• The tool should be user-friendly, so that the project partners in Kalundborg can use it themselves. • Language of the tool should be English.		
• **Long term**			

Title*	**Evaluation of treatment trains**	**Priority**	medium

General info*	The tool is able to evaluate treatment trains in order to rank the best suitable treatment solutions for the expected future situation regarding water and wastewater management.		
As*	Strategic planner	**Size/effort**	
I need to*	To compare different treatment options in terms of performance, energy efficiency, costs, etc.	**Dependency***	Data from demonstrated innovations (other EU projects?)
So that*	I can decide which treatment is best for alternative water resources not yet used in Kalundborg	**User story ID**	
Acceptance criteria*	• The tool should contain default values for the innovative technologies. • The tool should be user-friendly, so that the project partners in Kalundborg can use it themselves. • Language of the tool should be English.		
• **Long term**			

5.3 Syros (Greece) – Mediterranean region case study (island), focusing on water scarcity and the demonstration of AWR in a climate-vulnerable area.

Leader: DEYAS
Contact person: Dimitrios Vakondios
Email: dvakondios@deyasyrou.gr
Other Institutions involved: NCSRD, NTUA

Title*	**Real time water availability dashboard**	**Priority**	Low
General info*	This dashboard will utilise the existing water level sensors and digital flowmeter sensors (at key locations) to monitor water availability and identify potential network damages and leakages		
As*	A water manager	**Size/effort**	Large (due to real-

			time data integration and customization per region)
I need to*	Monitor water quantity and quantity in real-time, and receive alerts when thresholds are breached (low reserves level, potential leakage detected)	**Dependency***	IoT/water level sensors, data integration tools to link existing systems to the toll, real-time processing capability
So that*	I can make timely decisions for redirecting resources to the parts of the network needing additional wate supply. Minimize wate loss due to leakages.	**User story ID**	
Acceptance criteria*	• The system provides accurate monitoring of water reserves and identified potential leakages • Enables data acquisition (consumption-location relationship) for improving network understanding and improve operational management.		
• **Real time**			

Title*	**Real time energy consumption**	**Priority**	Low
General info*	This feature will inform the user to measure in real time the energy consumption of all pumping stations, desalination plants, and wastewater treatment facilities.		
As*	A water manager	**Size/effort**	Large (due to real-time data integration and customization per region)
I need to*	Monitor power consumption in real-time. Link energy consumption with water production and distribution	**Dependency***	Smart power meters
So that*	I can estimate power consumption/water	**User story ID**	

	production relationship. The data are needed for the upgrade strategy of the water production and distribution network, to increase sustainability.		
Acceptance criteria*	• The system provides accurate monitoring of energy consumption • Enables data acquisition (consumption-water production relationship) for improving network understanding and improve energy efficiency and sustainability.		
• **Real time**			

Title*	**Near future climate scenarios**	**Priority**	Medium
General info*	A dashboard showing short-term and long-term (seasonal) weather projections. This will allow water managers to plan desalinated water production, manage reserves, and plan potential desalination plants expansions in the long term.		
As*	A water manager	**Size/effort**	Small
I need to*	Short-term water production planning and resources management	**Dependency***	Climate data / Weather reports
So that*	Adjust and plan water production. Increase reserves, for near future use,	**User story ID**	
Acceptance criteria*	• The system provides accurate monitoring of energy consumption • Enables data acquisition (consumption-water production relationship) for improving network understanding and improve energy efficiency and sustainability.		

5.4 Costa Brava (Spain) – A Mediterranean region case study dealing with water scarcity and the use of Alternative Water Resources (AWR).

Leader: ICRA
Contact person: Nikolaus Klamerth
Email: nklamerth@icra.cat
Other Institutions involved: Eurecat

Title*	Real time water availability and water quality	**Priority**	high
General info*	A dashboard showing the available water and its quality in the system through water level sensors, flowmeter sensors, pipe pressure sensors and water quality sensors (in pipe analysis).		
As*	Water Manager	**Size/effort**	High, due to the combination of quantity and quality
I need to*	Monitor water quantity in real time and receive alerts when certain thresholds are breached (low water reserves, infrastructure damage). Monitor water quality in real time and receive alerts when certain thresholds are breached (pollutants, pathogens and overall water quality)	**Dependency***	Water level sensors, pressure sensors, flowmeter sensors, data integration tools into the existing system or upgrade the existing system. The water manager needs to access the data in real time and needs real time processing capability. Water analysis data, inline water monitoring, Analysis of pollutants and pathogens

So that*	I can make timely decisions for redirecting resources to the consumers I can monitor the distribution system and reduce losses I can make short time decisions in case of emergencies, which could affect consumers. I can monitor the distribution system and ensure overall water quality.	User story ID	
Acceptance criteria*	The system will provide accurate monitoring of water resources and reserves and identifies losses Data acquisition (consumer location relationships) to improve networks, network understanding and improve operational management. The system will provide accurate monitoring of water quality and will provide data in case of pollution. Data acquisition to monitor water quality at start and end points of the network.		

Title*	Reuse water treatment	Priority	
General info*	A dashboard showing the quality of the water passing through the Reuse Treatment Plant.		
As*	Reuse Water treatment Plant Operator	Size/effort	
I need to*	Monitor water quality of the Reuse plant to be sure that the treated water complies with all the norms for indirect potable reuse.	Dependency*	Online and inline water quality sensors, daily analysis of pathogens

So that*	I can make timely decisions to assure the water quality and react in case of failures. Can monitor the water quality and provide safe water for indirect water reuse.	**User story ID**	
Acceptance criteria*	The system will provide accurate monitoring of water quality for indirect potable reuse		

Title*	Near future climate scenarios	**Priority**	
General info*	A dashboard showing short-term and long-term (seasonal) weather projections. This will allow water managers to plan reuse water treatment plants, desalinated water production, manage reserves, and plan potential desalination plants / reuse plants expansions in the long term.		
As*	A water manager	**Size/effort**	
I need to*	Short-term water production planning and resources management	**Dependency***	Climate data / Weather reports
So that*	Adjust and plan water production. Increase reserves, for near future use,	**User story ID**	
Acceptance criteria*	The system provides accurate monitoring of energy consumption Enables data acquisition (consumption-water production relationship) for improving network understanding and improve energy efficiency and sustainability.		
Title*	Energy consumption	**Priority**	
General info*	A dashboard showing the energy demand of all water related stations including pumping stations, wastewater treatment plants, water treatment plants, water reuse plants and desalination plants.		
As*	A water manager	**Size/effort**	Large

I need to*	Monitor power consumption. Link energy consumption to water treatment and distribution.	**Dependency***	Smart power meters
So that*	I can estimate power consumption/water production relationship. The data are needed for the upgrade strategy of the water production and distribution network, to increase sustainability.	**User story ID**	
Acceptance criteria*	The system provides accurate monitoring of energy consumption Enables data acquisition (consumption-water production relationship) for improving network understanding and improve energy efficiency and sustainability.		

6. Detected data and back-end requirements

This section outlines the key data and back-end requirements necessary for the successful development and deployment of the RECREATE_WT platform. It focuses on the data sources, data processing needs, and back-end architecture that will support the platform's functionalities. This section ensures that all technical components are aligned to meet the requirements of the user stories and CS. The backend must be robust, scalable, and capable of handling real-time data processing and integration across various CS and functionalities.

Data Sources

- **Environmental Data**:
    - Climate models and historical weather data.
    - Real-time environmental sensors providing data on air and water quality, temperature, and precipitation.
    - Water availability metrics (groundwater levels, river flows, reservoir levels).
- **Water Quality and Quantity Data**:
    - IoT sensor data from water treatment plants, desalination plants, and recycled water systems.
    - Data on water contaminants, physicochemical concentrations, biological compounds, and temperature.
- **Socio-Economic Data**:
    - Population growth projections.
    - Water usage patterns by sector (agriculture, industry, domestic).
- **Regulatory Data**:
    - Water usage regulations and policies.
    - Legal frameworks for implementing AWR solutions.
- **Geospatial Data**:
    - Geographic Information System (GIS) data for mapping water infrastructure.
    - Land use data for water catchment areas.

6.1 Data description table

Below is a table template (Table 12) to describe the data required for the RECREATE_WT platform, providing essential information about the datasets, their nature, time frames, output formats, and how they will be managed in the backend:

Please fill out the following table with the required data that your case study or institution will provide for the RECREATE_WT platform. Each field has a brief explanation to guide your input. Add as many rows as is required.

Dataset Name	Static/Live	Dates	Output Format	Backend Format	Data access method and requirements	Observations
The name or category of the dataset (e.g., climate models, water quality data).	Indicates whether the dataset is static (fixed data) or live (updated in real-time).	Specifies the date range covered by the dataset (e.g., historical data, real-time updates).	The format in which the dataset is produced (e.g., CSV, JSON, GIS files).	This field **will be completed by the developer**, but it refers to the format in which the data will be stored and processed on the backend (e.g., JSON, XML, databases like PostgreSQL).	Specify how the data can be obtained: • **Method:** Indicate if the data is accessible via an API, a database, or another source. • **Credentials:** Note if login credentials, API keys, or special access permissions are needed. • **Specific Data:** Detail the exact data fields or parameters to collect (e.g., water temperature, pH, historical climate records).	Additional notes or any relevant comments, such as the specific tool or case study where the dataset will be used, data collection methods, or special processing requirements

Table 12 Data description table

Example of a completed entry:

Dataset Name*	Static/Live*	Dates*	Output Format*	Backend Format	Data access method and requirements*	Observations
Climate models	Static	Historical + Projections (1980 – 2050)	Maps, tables	JSON, CSV	Access through Copernicus Climate API, API key required, retrieve temperature, precipitation, and sea level rise projections.	Used for scenario analysis
Water Quality Data	Live	Real time + Historical (2020 – present)	Sensor readings	JSON, XML	Access via local IoT sensor database, credentials provided by utility operator, collect pH, turbidity, and contamination levels.	Integrated into the real-time dashboard

North Holland

Dataset Name*	Static/Live*	Dates*	Output Format*	Backend Format	Data access method and requirements*	Observations
Delta Scenarios	Static	Projections	Report Tables	TBD	API from Deltares	Climate scenarios (SSP1 & SSP5) merged with socio-economic projections in the Netherlands. Open data. River Rhine Discharge data used to derive chloride predictions.
Chloride Prediction	Static	Historical/ Projections	Tables	TBD	Access through email. Potentially API from PWN	Used for SDM. Derived from Delta Scenarios and PWN Ijsselmeer model.
Water Demand (Drinking)	Static	Historical	Table	TBD	Access through email.	Confidential data
Water Demand Scenarios	Static	Historical/ Projections	Tables	TBD	Access through email/ generated within the project based on historical	Potentially restricted data.

(Drinking water and Industry)					data and future projections with socio-economic data.	
WWTP Effluent Discharge	Static	Historical	Tables	TBD	Access through email	Potentially restricted data

Kalundborg

Dataset Name*	Static/Live*	Dates*	Output Format*	Backend Format	Data access method and requirements*	Observations
Climate models	Static	Historical + Projections (1980 – 2050)	Maps, tables	JSON, CSV	Access through Copernicus Climate API, API key required, retrieve temperature, precipitation, and sea level rise projections.	Used for T1.1
Reasonable estimations	Static	Historical and estimates for future	tables	CSV	Send by e-mail	Used for UWOT modelling

Syros

Dataset Name*	Static/Live*	Dates*	Output Format*	Backend Format	Data access method and requirements*	Observations
Climate modes	static	Historical (1980-2004) Projections (2025-2049 and 2075-2099) RCP4.5 and Rcp8.5	Based on request (NetCDF of csv)		Sent upon request	
Water demand and production	static	Historical and realtime (but not possivle to share via API)	tables	Csv or excel	Upon request	
Water network	static	Under develoment, partially ready	maps	shapefile	Upon request	Used for UWOT modelling
Water quality data	Real-time	tdt	tables	csv	Sensors to be installed	

Costa Brava

Dataset Name*	Static/Live*	Dates*	Output Format*	Backend Format	Data access method and requirements*	Observations
Climate models	static	Historical + Projections (1980 – 2050)	Maps, tables	JSON, CSV	Access through Copernicus Climate API, API key required, retrieve temperature, precipitation	Used for 1.1
Water demand	static	Historical and projections for future	tables	CSV	CSV, email and statistical data	UWOT modelling

Data Integration

- **Real-Time Data Processing**:
  - The backend must support real-time data streaming from IoT devices and sensors across the case studies.
  - Data must be collected, processed, and stored in near-real-time for immediate use in dashboards and decision-support tools.
- **Data Standardization**:
  - Ensure that all data sources are compatible with the RECREATE_WT platform through standardized formats (e.g., JSON, CSV, XML).
  - Data harmonization is needed to integrate data from different case studies and regions seamlessly.

Back-End Infrastructure

- **Cloud-Based Architecture**:
  - To be defined based on requirements
- **Database Management Systems (DBMS)**:
  - Use relational databases (e.g., PostgreSQL, MySQL) for structured data such as water usage metrics, sensor logs, and policy data.
  - NoSQL databases (e.g., MongoDB) can be used for unstructured data like sensor streams and real-time monitoring data.
- **APIs for Data Access**:
  - Develop RESTful APIs to enable seamless data access and integration between the platform's front-end and back-end.
  - APIs should support data querying, fetching, and updates in real-time.

7. Detected front-end requirements

This section focuses on collecting essential information regarding the front-end requirements needed for the RECREATE_WT platform. The front-end components will directly interface with end-users and must be tailored to the specific functionalities and design preferences for each CS. The table below (**Error! Reference source not found.**) will help stakeholders provide detailed requirements for how the user interface and user experience (UI/UX) should function, ensuring that the platform meets the needs of its various users.

7.1 Data description table

Please fill out the following table with the front-end requirements needed for your CS or institution. This includes functionality, user interaction, design preferences, and specific tools or visualizations that are crucial for the user experience. Pay special attention to describing user personas, required visualizations, and any interactivity.

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
Name the feature or interface element (e.g., real-time monitoring dashboard, data visualization tools, reporting panel) that users will interact with	Define the primary user for this front-end feature. Different personas could include: • **Water Manager**: Needs access to operational data and real-time monitoring • **Policy Maker**: Requires scenario-based visualizations to assist in decision-making • **Researcher**: Needs detailed data analysis tools and simulation capabilities	Specify what the feature must do for the user. Example functionalities include: • Monitoring water quality in real-time • Simulating long-term effects of water managemen2t decisions • Generating reports based on live data	Indicate what data or tools need to be shown on the front-end. This might include: • Data visualizations such as graphs, charts, or maps • Live updates, alerts, or notifications • Input forms for user interaction or scenario creation	Describe the interactive features required, such as: • User controls (e.g., buttons, sliders, toggles) • Zoomable maps, draggable elements, or interactive charts • Real-time data refresh, dynamic content updates	Rank the feature's priority based on its importance to the platform: • **High**: Critical to core platform functionality • **Medium**: Important but not essential for the initial version • **Low**: Can be added in later iterations	Add any specific design preferences, accessibility requirements, or UX elements that need to be considered (e.g., "User-friendly for non-technical users," "Should be available in multiple languages," or "Customizable dashboards")

Table 13 Table User Interface

Example of a completed entry

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
Real-Time Monitoring Dashboard	Water Manager	Monitor real-time water quality and quantity data	Real-time sensor data (pH, flow rate, turbidity), alert thresholds	Graphs, gauges, real-time notifications, filter options for different parameters	High	Should support real-time data refresh every 10 seconds, must be available in both English and local languages
Scenario Simulation Tool	Policy Maker	Simulate long-term effects of AWR on water resources	Climate projections, water usage data, policy scenarios	Interactive charts, scenario comparison tables, downloadable reports	Medium	Needs an easy-to-use interface for non-technical users, and must be available in English

North Holland

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
SDM	Water personnel and researchers	Generate strategic management guidance. Help define alternative water resource options.	Provide scenario data on water resource availability, demand scenarios, and alternative water resource options.	Model with user interface.	Low	Developed for technical user.
Serious Game	Policy Makers, Water personnel, and general public (potentially)	Generate knowledge about (alternative) water management and strategic planning.	To be designed together with Community of Practice of North Holland.	Serious game with backend system dynamics model	Medium	Developed for non-technical users. Language (English/Dutch) to be decided.

Kalundborg

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
Climate scenario simulation	Strategic planner	Generate knowledge in order to adapt the water management strategy	Data to assess the availability of groundwater and surface water as resource for alternative water reuse in future scenarios	Interactive charts, scenario comparison tables, downloadable reports	Medium	User-friendly for non-technical users, should be available in English
Evaluation of treatment trains	Strategic planner	Generate knowledge in order to adapt the water management strategy	Data to compare and rank different innovative treatment trains	Interactive charts, scenario comparison tables, downloadable reports	Medium	User-friendly for non-technical users, should be available in English

Syros

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
Real time energy consumption	Water Manager	Monitor real-time power consumption data	Real-time sensor data powermeters	Graphs, gauges, real-time notifications,	Low	Should support real-time data refresh every 10 seconds, must be available in both English and local languages
Real time water availability dashboard	Water Manager	Monitor real-time quantity data	Real-time sensor data (flow rates, water level in reservoirs)	Graphs, gauges, real-time notifications, filter options for different parameters	High	Should support real-time data refresh every 10 seconds, must be available in both English and local languages
Near future climate scenarios	Water Manager	Plan as accurate as possible water reserves	Expected rainfall (dates, expected rainfall rate)	Graph	Medium	Should be updated daily based on the modelling results. 3-day forecast, 7-day forecast, seasonal

Costa Brava

Front-End Feature*	User Persona*	Functionality*	Data/Tools to Display*	Visual/Interactive Elements*	Priority*	Observations
Real-Time Monitoring Dashboard	Water Manager	Monitor real-time water quality and quantity data	Real-time sensor data (pH, flow rate, turbidity), alert thresholds	Graphs, gauges, real-time notifications, filter options for different parameters	High	User-friendly for non-technical users, should be available in English
Scenario Simulation Tool	Policy Maker	Simulate long-term effects of AWR on water resources	Climate projections, water usage data, policy scenarios	Interactive charts, scenario comparison tables, downloadable reports	Medium	Needs an easy-to-use interface for non-technical users, and must be available in English

11. References

IEEEXplore. Jan. 610-1990 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. *doi: 10.1109/IEEESTD.1991.106963*, IEEE Std 610 , vol., no., pp.1-217. 10.1109/IEEESTD.1991.106963

Rocha, H. d., Espirito-Santo, A., & Abrishambaf, R. (2020). Semantic Interoperability in the Industry 4.0 Using the IEEE 1451 Standard. IEEE, doi:10.1109/iecon43393.2020.9254274.

In case of any questions, please contact:

Eloy Hernandez
Researcher, EURECAT
eloy.hernandez@eurecat.org

Didac Colominas
Researcher, EURECAT
didac.colominas@eurecat.org

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101136598. This document reflects only the views of RECREATE consortium, neither the European Commission nor any associated parties are responsible for any use that may be made of the information it contains.