Ref. Ares(2022)6909062 - 06/10/2022

score

D8.3 – GIS-Based Early Warning and Digital Twin Platform Interface Control Document

DATE OF DELIVERY – 30/06/2022

AUTHOR(S) – Attilio Vaccaro (MBI), Antonello Casula (MBI), Andrea Rucci (MBI), Giovanni Serafino (MBI), Iulia Anton (ATU), Khurram Riaz (ATU), José P. Gómez Barrón (UCD)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101003534

DOCUMENT TRACKS DETAILS

Project acronym	SCORE
Project title	Smart Control of the Climate Resilience in European Coastal Cities
Starting date	01.07.2021
Duration	48 months
Call identifier	H2020-LC-CLA-2020-2
Grant Agreement No	101003534

Deliverable Information	
Deliverable number	D8.3
Work package number	WP8
Deliverable title	GIS Based Early Warning and Digital Twin Platform Interface Control Document
Lead beneficiary	MBI Srl
Author(s)	Antonello Casula (MBI), Andrea Rucci (MBI), Attilio Vaccaro (MBI), Giovanni Serafino (MBI)
Due date	30/06/2022
Actual submission date	30/06/2022
Type of deliverable	Report
Dissemination level	Public

VERSION MANAGEMENT

Revision table			
Version	Name	Date	Description
V 0.1	Attilio Vaccaro (MBI), Antonello Casula, (MBI), Andrea Rucci (MBI)	01/04/2022	First draft
V 0.2	Attilio Vaccaro (MBI), Antonello Casula (MBI), Andrea Rucci (MBI)	01/06/2022	Updated draft internally reviewed
V 0.3	Attilio Vaccaro (MBI), Antonello Casula(MBI), Andrea Rucci(MBI), Khurram Riaz (ATU), José P. Gómez Barrón (UCD)	09/06/2022	Updated draft after contribution from partners
V1.0	Attilio Vaccaro (MBI), Antonello Casula (MBI), Andrea Rucci (MBI), Giovanni Serafino (MBI), Khurram Riaz (ATU), José P. Gómez Barrón (UCD)	30/06/2022	Final version
V1.1	Giovanni Serafino (MBI), Antonello Casula (MBI), Andrea Rucci (MBI), José P. Gómez Barrón, (UCD) Salem Gharbia (ATU)	29/09/2022	Reviewed final version

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym / Abbreviation	Meaning / Full text
2D	2-dimensional
3D	3-dimensional
3DCityDB	3D City Database
API	Application program interface
CCLL	Coastal City Living Lab
CityGML	City Geography markup language
CRUD	Create, read, update, and delete
CSW	Catalogue service for the web
DEV	Development
DT	Digital twin
DTEW	Digital Twin and Early Warning
DTEWS	Digital twin-early warning support
EBA	Ecosystem-based approach/adaptation
ENV	Environmental
EU	European Union
EW	Early Warning
EWS	Early-warning support
FAIR	Findability, accessibility, interoperability, reusability

FUN	Functional
GeoJSON	Geographic JSON
GIS	Geographic Information System
glTF	Graphics language Transmission Format
GML	Generalised markup language
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
ICT	Information and communication technologies
INSPIRE	Infrastructure for spatial information in the European Community
IoT	Internet of things
JSON	Javascript object notation
KML	Keyhole markup language
MMI	Man-to-machine interface
MQTT	Message queue telemetry transport
NBS	Nature-based solution
OGC	Open Geospatial Consortium
OPE	Operational
OWS	OGC web services
PER	Performance
PSO	PowerSchedO
REST	REpresentational State Transfer
RPO	Research performing organizations
SCORE	Smart control of the climate resilience in European coastal cities
SEC	Security

SHP	Shapefile
SIP	SCORE ICT Platform
SME	Small-medium enterprise
SOS	Sensor observation service
SPS	Sensor planning service
STA	SensorThings API
SWE	Sensor web enablement
SYS	System
TES	Test
TBD	To be done/defined
URL	Uniform resource locator
WCS	Web coverage service
WFS	Web feature service
WFS-T	WFS-Transactional
WMS	Web map service
WMTS	Web map tile service
WP	Work Package
XML	Extensible markup language

BACKGROUND: ABOUT THE SCORE PROJECT

SCORE is a four-year European Union (EU)-funded project aiming to increase climate resilience in European coastal cities.

The intensification of extreme weather events, coastal erosion and sea-level rise are major challenges to be urgently addressed by European coastal cities. The science behind these disruptive phenomena is complex, and advancing climate resilience requires progress in data acquisition, forecasting, and understanding of the potential risks and impacts for real-scenario interventions. The Ecosystem-Based Approach (EBA) supported by smart technologies has potential to increase climate resilience of European coastal cities; however, it is not yet adequately understood and coordinated at European level.

SCORE outlines a co-creation strategy, developed via a network of 10 coastal city 'living labs' (CCLLs), to enhance coastal city climate resilience rapidly, equitably and sustainably through EBAs and sophisticated digital technologies. The 10 coastal city living labs involved in the project are: Sligo and Dublin, Ireland; Barcelona/Vilanova i la Geltrú, Benidorm and Basque Country, Spain; Oeiras, Portugal; Massa, Italy; Koper, Slovenia; Gdansk, Poland; Samsun, Turkey.

SCORE will establish an integrated coastal zone management framework for strengthening EBA smart coastal city policies, creating European leadership in coastal city climate change adaptation in line with The Paris Agreement. It will provide innovative platforms to empower stakeholders' deployment of EBAs to increase climate resilience, business opportunities and financial sustainability of coastal cities.

The SCORE interdisciplinary team consists of 28 world-leading organisations from academia, local authorities, research performing organizations (RPOs), and Small-medium enterprises (SMEs) encompassing a wide range of skills including environmental science and policy, climate modelling, citizen and social science, data management, coastal management and engineering, security and technological aspects of smart sensing research.

EXECUTIVE SUMMARY

This document is a deliverable of the SCORE project, funded under the EU's Horizon 2020 research and innovation programme under grant agreement No 101003534.

The aim of this document is to identify the interfaces of the Geographic information system (GIS) Based Digital Twin and Early Warning (DTEW) Platform. This Platform is internally composed of two main parts: the backend, which deals with the Digital Twin system and the processing of several categories of data using a set of models defined in SCORE project WP3; the frontend, which is the Man-Machine Interface (MMI) that deals with user's interaction. Starting from the functional requirements and system architecture described in [2] and [3], a list of internal and external interfaces is defined and their use is described.

LINKS WITH OTHER PROJECT ACTIVITIES

This document takes inputs from the SCORE project deliverables D8.1 - GIS Based Digital Twin Platform functional requirements [2] and D8.2 - GIS Based Early Warning and Digital Twin Platform system architecture and design [3]. It is also based on the models defined in SCORE project WP3, the low-cost sensors specifications from SCORE project WP4, the SCORE Information and communication technologies (ICT) platform (SIP) platform developed in SCORE project WP5, the risk maps provided by SCORE project WP6, the list of Nature-based solutions (NBS) specified in SCORE project WP7.

Outputs of this document are relevant for the SCORE WP8 system development, integration, deployment, validation and assessment activities (project tasks T8.2, T8.3, T8.4, T8.5)

TABLE OF CONTENT

INDEX OF FIGURES

INDEX OF TABLES

1. INTRODUCTION

1.1 Scope of the document

This document is part of the technical deliverables of WP8 (Development of integrated Early Warning Support System and spatial Digital Twin solution prototypes) of the SCORE project. In particular, this deliverable is aimed at describing the control interfaces and the protocols needed by the GIS Based Early Warning and Digital Twin Platform (DTEWs) system to share and exchange information and commands between them (internal interfaces) or with external systems (external interfaces).

This document is structured as follows: in Sect.2, the high-level system architecture of the platform is described. It involves a summary of the global structure of the system, going through the definition of the inner system components (the GIS-Based Visualization tool and the DTEW part), the external systems and the available user interactions and operations. Then, Sect. 3 and Sect.4 are devolved, respectively, to the description of the internal and external interfaces, according to the general structure defined in the following paragraph. In Sec. 5 a description of the communication protocols and the design guidelines is given (REST, OGC, etc...), while in Sec. 6 details about the software repository and third-part software are reported.

1.2 Interface description

The set of interfaces used by the system and reported in this document will be described using a common pre-defined structure. Interfaces are implemented to allow data transfer between the components of the system. In the following sections we will refer to *Data Plane* as the set of interfaces devolved to the actual data exchange, while those related to the actions allowing to interact with the system will be indicated as *Control Plane*. A generic interface is described by a table in the form shown in Table 1.

Table 1 Generic interface structure

Symbolic name	-
Description	-
Source	-
Destination	-
Direction	-
Pre-conditions	-
Network Protocol	-
Data Format	-
Data Content	-
Error codes and Exceptions	-

Testing	-
Notes	-
Example	-

For each interface a unique *Symbolic Name* will be defined, with format **SCORE-DTEW-XXX-YYY.** Here XXX is a string indicating whether the interface is internal (**INT**) or relates to external components (**EXT**) e.g., between the Digital Twin and the GIS-based Visualization Tool or with the SIP, respectively. The next and final code YYY is a unique serial identifier describing the single interface and follows the code structure shown in Table 2.

Table 2. Interface Codes definition

Type	Category	Codes
Internal	Data Plane	1-100
Internal	Control Plane	101-200
External	Data Plane	201-300

The full specification of an interface requires a *Description*, which is intended to be a descriptive text that specifies the type of information that will be exchanged through the interface. *Source* and *Destination* fields indicate the endpoints, while the *Direction* specifies if the interface can be used to send (with value *Out*), receive (*In*) or exchange data (*In-Out*). If present, *Pre-Conditions* can be also specified. Most interfaces are implemented as data transfer between the components through network packets. The parameter *Network Protocol* specifies in which way the exchange of information is achieved. The fields *Data format* and *Data contents* specify how data passed through the interface are structured and the list of possible fields that are transmitted. The first one can be a reference to a well-known standard, a file format or a specification for custom or proprietary format. *Errors* and *Exceptions* specify possible error codes transmitted over the interface in case of problems e.g., bad requests. If available, an example of message exchange through the interface can be provided in the field *Example.* Finally, needed *Testing* guidelines or additional *Notes* not covered by the previous fields can also be specified.

2. HIGH LEVEL SYSTEM ARCHITECTURE

The core architecture of the SCORE Digital Twin and Early Warning Support (DTEWS) system is composed by two main parts strictly connected by each other. A summarized and schematic representation of the system architecture is shown in Figure 1 while details about its implementation and features are reported in the Deliverable D8.2. The first part is the DTEWS Backend System, which involves the objects and the logic that defines the Digital Twin itself together with the procedures related to the Early Warning Support. This platform provides the virtual representation whose visualization is devolved to the GIS based visualizer, the other component of the system.

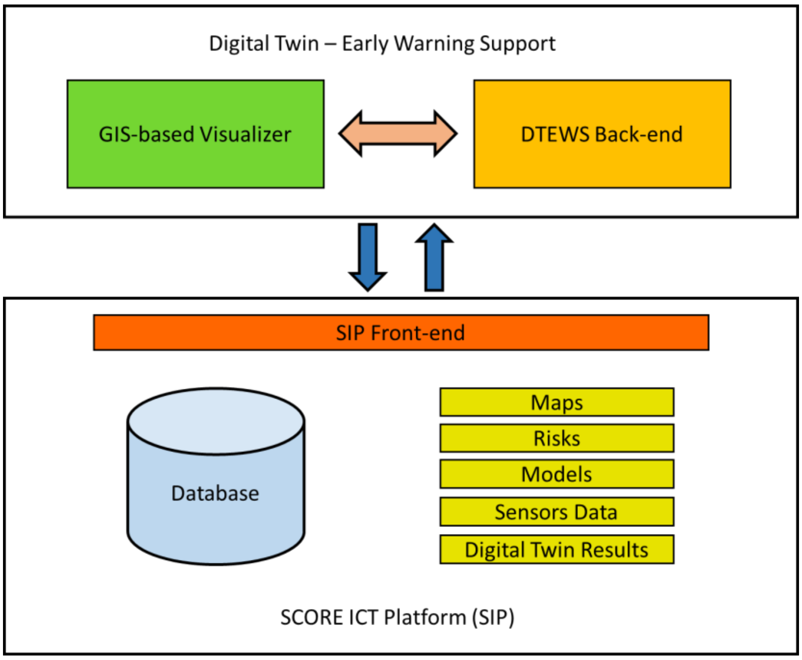

Figure 1. Architecture of the SCORE Digital Twin and Early Warning Support System (DTEWS).

The system makes use of the data stored in the SIP and shares them with several objects needed by the Digital Twin for the definition of the virtual representation of the environment e.g., maps of the city, risk maps and sensor data.

The global architecture of the SCORE system is depicted in FIg. 2 below. The architecture of DTEW system is structured on multiple layers composed by several subsystems and technological components (see Fig. 1). At the core of the backend side of the DTEW is the SCORE ICT Platform, termed as SIP. The SIP infrastructure offers a web-based management content system for geospatial data and services based on GeoNode and Geoserver open-source technologies. The SIP enables SCORE information systems, operations, and users (i.e., members, stakeholders and CCLLs) to explore, upload, manage and publish geospatial data, maps, and resources. Using Open Geospatial Consortium (OGC)-compliant standards and web services, it makes all data available as unique web resources. Through the SIP, the different content, resources, and spatial data generated and used by SCORE work packages (WPs) and members are aggregated and stored on a central database (PostgreSQL/PostGIS) that has capabilities to store and query both spatial and non-spatial attributes. The SIP enables a GeoNode Representational state transfer (REST) Application program interfaces (APIs) connected to the database to provide an import and transformation data layer to ingest different datasets from a variety of formats coming from SCORE WPs activities. Thus, thematic base line data and maps and data outputs produced by WPs and DTEW environmental modelling and simulations are stored in the database and available in the backend to be published via standard web mapping services. The SIP REST API exposes several endpoints to create, read, update, and delete (CRUD) operations available for the DTEW backend components and web client. This enables the creation (Hypertext transfer protocol, HTTP, POST method), reading (HTTP GET method), update (HTTP PUT/PATCH methods) and delete (HTTP DELETE method) of users uploads, documents, datasets, maps, resources, etc. This component of the SIP offers a catalogue and a search and the data recovery engine to exchange data with other SCORE systems.

Together with the component for web mapping services, a second SIP component is the interface with sensors. The SIP makes accessible two web services using two different OGC standards and protocols to manage sensors, their locations, observations, datastreams, and get observed properties of a feature of interest. The OGC SensorThings API facilitates the managing of sensors by proving great interoperability, making data exchange more flexible and easier for CCLLs and citizen sensing activities when there is a heterogeneity of sensor hardware technologies and observed properties and variables. Thus, the OGC SensorThings API (STA) servers are available mostly to connect and get data directly from CCLLs sensor networks. Additionally, together with the STA, the OGC Sensor Observation Service (SOS) implementation helps to standardize the observations pulled from public-available, official, or legacy sensors, and data from citizen-driven monitoring activities having an INSPIRE-compliant service. These web services help to integrate several data sources for publishing all SCORE available sensor observations. Both SIP and sensor technologies have some resources available to request data from the DTEW components. Finally, as part of the data management and publication, a third component, that interoperates with the SIP, is the availability of 3-dimensional (3D) portrayal services for the visualization of 3D geospatial data in the DTEW GIS-based client. This component aims to manage virtual 3D city models when available for a coastal city using the open source 3DCityDB (3-dimension City Database) spatial database that implements the OGC City Geography markup language (CityGML) standard. CityGML provides a standard model and mechanism for describing 3D objects including the geometry, topology, semantics, and appearance by defining different data levels of detail. Using the 3DCityDB tools the available data can be easily transformed into formats that are optimized for web visualisation and 3D mapping such as Keyhole markup language (KML) and graphics language Transmission Format (glTF). Also, it can be enabled a basic Web feature service (WFS) to directly allow web-access to the 3D city objects in the database without transforming the CityGML data.

The next core technology at the server layer, together with the components for SCORE's data management and exchange, is the PowerSchedO system (PSO) that provides a framework for environmental modelling, analysis, and simulation. PowerSchedO is the backend part in charge of creating and run city and environmental analysis and modelling scenarios and translate the results into actionable data and insights for the DTEW end-users. It has several components including a REST API connected to the PSO database for data access and transfer with other system components, that exposes API endpoints to manage the executions, processes and outcomes of the runs and simulations, including their status, configurations of scenarios and simulations, execution logs and results. The PSO readers and writers retrieve input data from the SIP, convert the data to be used in the modelling component and send back the outputs to the SIP and DTEW data repositories to be requested by the frontend visualisations tools. The PSO modelling works firsts by abstracting the real-world characteristics into a mathematical representation. Then, it executes different computational models that include the set of entities or objects existing in a city or environment, their attributes, processes, and relations. The actual application of the models is performed by processors that are triggered by a system rule or state, or by end-users. The early-warning support (EWS) processors, a Short-Term Forecaster and a Warning Generator have the tasks of evaluating, estimating, and warning about environmental hazards and risks. The Digital twin (DT) processors perform the scenario modelling of both the current situation of the environment, and the simulated or created conditions to be analysed.

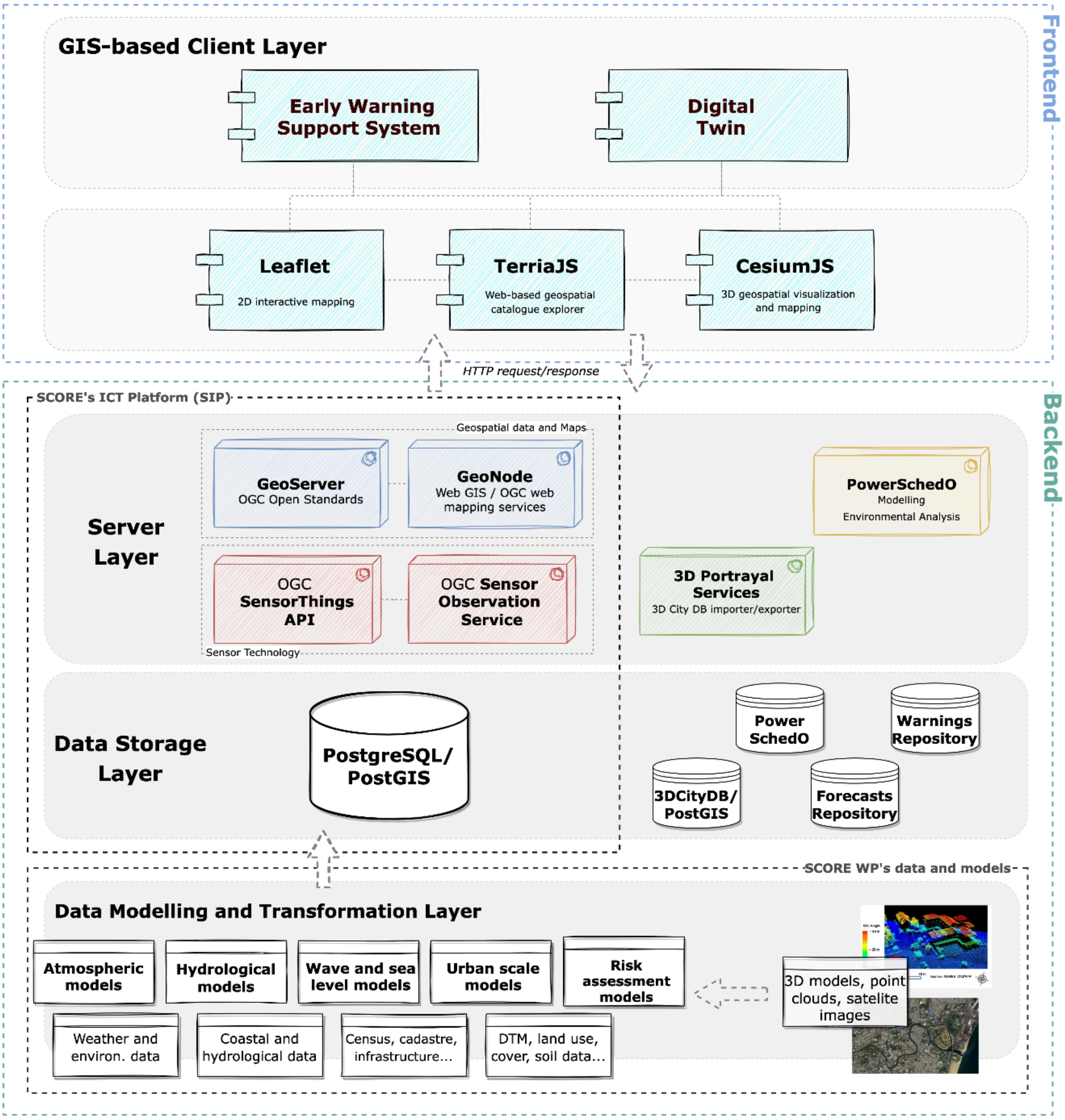

Figure 2. Global architecture of the SCORE system

The last components of the DTEW system architecture are those of the GIS-based client layer. This layer comprises the frontend interface to communicate with the backend resources and capabilities introduced above via a HTTP request/response model and OGC communication protocols. The DT and EWS system offer several web GIS functionalities for data visualisation based on open-source geospatial technologies. Leaflet is used to enable 2-dimensional (2D) interactive and dynamic mapping functionalities. It makes easy to work with map tiled layers, feature vector layers and data formats for easily retrieving and displaying the different base maps and thematic layers served by SIP OGC standards. CesiumJS client library is used for 3D geospatial visualisation and mapping. It facilitates the integration of 3D tiles to stream imagery, 3D terrain data, and 3D models and vector data to load geometries and attributes. Finally, TerriaJS is used as a web GIS development framework for building geospatial apps that combine 2D and 3D web mapping features. It facilitates the assembly of a geospatial catalogue explorer from various data source providers, supporting easy integration of different geospatial data formats and visualization tools.

2.1 System Components

2.1.1 GIS-based visualization system

The frontend part of the DTEWS is based on three web mapping technologies, Leaflet for 2D interactive mapping, CesiumJS for 3D geospatial visualisation, and TerriaJS as a web GIS development framework that facilitates the integration of a geospatial data catalogue explorer for connecting to diverse data formats and providers, and for 2D and 3D mapping capabilities. The main components of the GIS-based frontend are the i) Geospatial Catalogue Explorer; ii) Map viewer; iii) Map editor; iv) Data Charting and time-series; v) Alerts, Warning and Hazards Data Aggregator. These components are presented in more detail in the Derivable D8.2.

2.1.2 Digital Twin and Early Warning Support system

The backend part of the DTEWS is based on the PowerSchedO application server, a processing engine for business analytics and optimization solutions developed by MBI s.r.l. [1]. It relies on an object-oriented approach in which the virtual representation of the cities, the physical models and all the processes related to them are implemented using a set of specific classes. The framework is designed to handle a large class of problems and is based on a stack of many abstraction layers used to translate the real/business world in a mathematical problem. In particular, the specialization of PowerSchedO within the context of SCORE involves the realization and integration of several elements:

Internal State Management Control API

Interface libraries with external data sources

Set of rules for the realization of the DTEWS

Dedicated workers for the execution of the DTEWS models

For details about the system architecture, we refer to Deliverable D8.2 [3].

2.2 External Systems

The system interacts only with the SIP for storing and retrieving the data used by the GIS-based Visualizer and the DTEWS.

2.2.1 The SCORE ICT Platform SIP

One of the main objectives of the SCORE project is the development of the SCORE ICT platform (SIP), featuring the SCORE database. It will contain all the heterogeneous data and models generated from all the SCORE WPs, together with relevant interfaces which will be employed to collect, store, and share the data acquired and processed during the project, while ensuring Findability, Accessibility, Interoperability, and Reusability (FAIR) features. Moreover, the SIP platform will ensure long-term data storage and will therefore represent an efficient tool for sharing knowledge on EBAs and on their efficacy against extreme events, sea level rise, and coastal erosion risks. Furthermore, SIP will provide data supporting both the development of digital twin solutions to smart instant monitoring and control of climate resilience and the development of the smart city early warning support system against extreme events.

The SIP will contain all the heterogeneous data collected during the project: on one hand, it will store the raw data coming, e.g., from meteorological models or historical records, from environmental risk maps, or raw data gathered by the already established environmental sensor networks, as well as by the low-cost sensors operated by citizens. On the other hand, the SCORE ICT platform will collect and make available all the relevant results of data processing by the DT and EWS systems.

The data management approach adopted by the SIP is specified in the Data Management Plan drawn in Task 5.2. As the baseline, only the members of the SCORE consortium will be enabled to upload data into the SCORE database and access to the data will be restricted to the partners of the project, to CCLLs and to the stakeholders. A software interface for data up-and down-load will be developed, too.

2.3 User's interactions and operations

2.3.1 User Authentication

A user signup and login interface are available in the frontend for users to register and start a session in the DTEW system. The DTEW available functionalities depend on the current logged user and are based on different user categories and permissions.

2.3.2 Map selection

The map selection component uses the Geospatial Catalogue Explorer based on OGC Catalogue service for the Web (CSW) and connected to the available geospatial data resources in the SIP. This component is used by the frontend system to discover datasets and live querying of online data services, to select and add map layers (baseline and simulation thematic layers) and to change the base maps. The geospatial catalogue groups and manages catalogue items from different data using various geospatial formats and existing Web map service (WMS), WFS, STA, SOS, Geographic Javascript Object Notation (GeoJSON), glTF layers, etc.

2.3.3 Map visualisation

The map viewer component is used by the front-end system to contain and display different layers of geographic data and services. The map view can display map layers and graphics in 2D, controlling the area or extent of the map, location and zoom levels that best fits the data that a user has selected. It supports user interactions such as pan and zoom to facilitate the navigation and data exploration. Also, it is possible to enable 3D scenes allowing to present visualizations of landscapes, 3D terrain and, when available, buildings and other 3D objects. The component enables falling back to 2D when 3D data are not available, and easily switching between 2D and 3D on the fly, based on Leaflet and Cesium JavaScript libraries for interactive mapping and 3D geospatial visualization.

2.3.4 Risk Map visualisation

The available risk maps produced by WP6 and those coming for DTEWS modelling outputs will be visualized using GeoTIFF format, WMS, Web map tile service (WMTS), and Web coverage service (WCS) protocols for raster data, and formats as GeoJSON, Generalised markup language (GML), Shapefile (SHP), KML, and the WFS protocol for vector data.

2.3.5 Visualisation of statistics collected by R scripts provided by task T3.3

The visualisation of the statistical outputs is supposed to be produced by the WP3 in their task 3.3. Currently, they have created an inventory of some R packages and scripts as shown in Fig 3. These scripts are to be used as tools to perform statistical analysis, but the feasibility of their integration into the DTEWS system is still under evaluation. Indeed, the DTEWS development is based on another programming language and merging codes of different languages in the same system can be a too high additional effort.

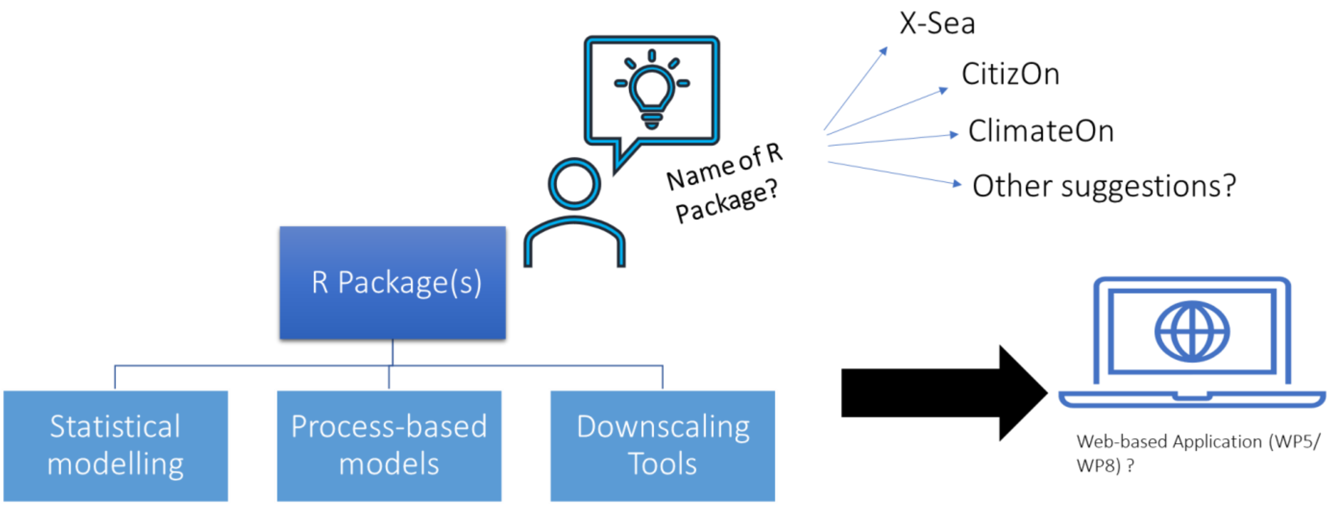

Figure 3. R packages suggested by WP3 for visualisation of the statistical outputs

2.3.6 Editing

The map editor component of the DTEW frontend offers editing tools for authorized users to update map layers and metadata of available sensors. A WFS transactional (WFS-T) service, enabled through the SIP, is used for editing geographic features. WFS-T supports transaction requests to edit data (geometries and attribute data) of a selected map feature layer.

2.3.7 Simulation scenario

The map editor component of the DTEW frontend provides tools for the creation and configuration of simulation scenarios. Simulations can be run employing historical or synthetic (i.e. artificial, ad-hoc prepared) data sets.

2.3.8 Simulation

Information about the execution process, like its duration and status, together with generated alerts and outcomes of a simulation are available in the map editor frontend component for feedback and visualisation.

2.3.9 Events

A user interface within the map editor component of the frontend shows a list of events to select and run event-based simulations as part of the scenario and of the configuration editor.

3. INTERNAL INTERFACES

This section describes the interfaces between the GIS-based visualizer and the Digital Twin and Early Warning Support Backend systems. The Backend is based on PowerSchedO: it exposes a REST interface that allows the GIS-based visualizer to configure, manage and monitor the simulations executed by the Digital Twin. Through this interface, the visualizer creates a scenario providing all the references to the data stored in the SIP that must be used in the simulation. Then it can activate the run of the simulation and monitor its status. When the simulation is completed, the visualizer can retrieve the result or the error and show it to the users.

3.1 Control Plane

3.1.1 Simulation

The following API describes how to create a new simulation and list the existing ones. When created, a simulation is in the NEW status: the scheduler component of PowerSchedO decides when the system has enough resources to start a new simulation. A simulation execution can be described with an ID and a STATUS.

Symbolic name	SCORE-DTEW-INT-001
Description	This interface is used to control the execution of a simulation
Source	Frontend
Destination	Backend
Direction	In-Out
Pre-conditions	
Network Protocol	REST
Data Format	JSON
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

Simulations execution are managed using the endpoint: /api/score/simulations_executions

Method	Action
POST	Create new simulation execution request

GET	List existing simulations executions
PUT	Not Available
DELETE	Not Available

Symbolic name	SCORE-DTEW-INT-002
Description	This interface is used to define, modify, retrieve and delete information related to a simulation
Source	Frontend
Destination	Backend
Direction	In-Out
Pre-conditions	
Network Protocol	REST
Data Format	JSON
Data Content	Input: Simulation ID, Output: simulation status: see table below
Error codes and Exceptions	TBD
Testing	
Notes	

A single simulation is managed using the endpoint: **/api/score/simulation_executions/ID**

Method	Action
POST	Not Available
GET	Retrieve specific simulation execution
PUT	Update specific simulation execution status
DELETE	Remove specific simulation execution

The possible statuses are:

NEW	A new simulation, ready to be executed when resources are available
RUNNING	The simulation is running
FINISHED	Simulation completed without errors
INTERRUPTED	Simulation interrupted by the user, can be removed or the status must be changed to NEW in order to be scheduled for a new run
ERROR	An error has occurred and the simulation cannot be completed

3.2 Data Plane

3.2.1 Scenarios

A scenario can be described by:

ID

Name

Description

Baseline map ID

List of ID of changes to baseline map

List of risks maps ID, if available

List of simulation events ID

Symbolic name	SCORE-DTEW-INT-101
Description	This interface is used to create a new scenario and to list existing ones
Source	Frontend
Destination	Backend
Direction	In-Out
Pre-conditions	
Network Protocol	REST
Data Format	JSON
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

Scenarios are managed using the endpoint **/api/score/scenarios**

Method	Action
POST	Create new scenario
GET	List existing scenarios
PUT	Not Available
DELETE	Not Available

A single scenario is managed using the endpoint **/api/score/scenarios/ID**

Method	Action

POST	Not Available
GET	Retrieve specific scenario
PUT	Update specific scenario
DELETE	Remove specific scenario

3.2.2 Simulation

A simulation can be described by:

ID

Scenario ID

Creation Timestamp

Symbolic name	SCORE-DTEW-INT-102
Description	This interface is used to define, modify, retrieve and delete information related to a scenario
Source	Frontend
Destination	Backend
Direction	In-Out
Pre-conditions	
Network Protocol	REST
Data Format	JSON
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

Simulations are managed using the endpoint **/api/score/simulations**

Method	Action
POST	Create new simulation
GET	List existing simulations
PUT	Not Available
DELETE	Not Available

A single simulation is managed using the endpoint **/api/score/simulation/ID**

Method	Action

POST	Not Available
GET	Retrieve specific simulation
PUT	Update specific simulation
DELETE	Remove specific simulation

3.2.3 Simulation Results

A simulation result can be described by:
ID
Simulation ID
Start Timestamp
End Timestamp
Status
Logs
Warnings
List of output ID of GIS-based results stored into SIP Database

Symbolic name	SCORE-DTEW-INT-103
Description	This interface is used to retrieve and delete information related to a result of a simulation
Source	Frontend
Destination	Backend
Direction	In-Out
Pre-conditions	
Network Protocol	REST
Data Format	JSON
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

Simulations are managed using the endpoint **/api/score/simulations_results**

Method	Action
POST	Not Available
GET	List existing simulations results
PUT	Not Available

DELETE	Not Available

A single simulation is managed using the endpoint **/api/score/simulation_results/ID**

Method	Action
POST	Not Available
GET	Retrieve specific simulation result
PUT	Not Available
DELETE	Remove specific simulation result

4. EXTERNAL INTERFACES

At the present stage of the SCORE project development, according to the proposal, the only external system with which the DTEWS will interact is the SCORE ICT platform. This section describes the public interfaces of the DTEWS.

4.1 Data Plane

The SIP exposes a REST interface that allows both GIS-based visualizer and the Digital Twin and Early Warning Support Backend systems to interact with the data storage. All data are managed using the /dataset service: the resources are distinguished by type. Using the information retrieved by this interface, the system can interact with the data using the OGC protocol (for further details, see Sect. 5.2).

4.1.1 Maps

The baseline maps of the coastal cities are a set of layers that describe the actual status of the city. Each system can retrieve only the needed layers.

Symbolic name	SCORE-DTEW-EXT-201
Description	This interface is used to retrieve baseline maps
Source	Frontend
Destination	SCORE SIP
Direction	In
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

4.1.2 Modified maps

The modified maps are a specific dataset that describes the changes of the referred layer of the baseline map. The purpose of the modified map service is to allow users to create a layer over the baseline map that contains the modifications to be used in a simulation of a change in the city map.

Symbolic name	SCORE-DTEW-EXT-202
Description	This interface is used to manage the modified maps
Source	Frontend
Destination	SCORE SIP
Direction	In-Out
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

The available actions are described in the following table:

Action	Description
Create	Used by the GIS-based visualizer to create a new layer with the modifications to the baseline map done by the user.
Retrieve	Used by the GIS-based visualizer to show the available modifications and to represent the selected modifications Used by the DTEW to perform simulations
Update	Used by the GIS-based visualizer to apply the changes done by the users
Delete	Used by the GIS-based visualizer to remove the layers that are not useful anymore

4.1.3 Risk Maps

The risk maps can be retrieved as dataset related to a specific set of maps. The GIS-based visualizer retrieves the list of available risk maps for the selected map and show this information to the users. The DTEWS retrieves the maps to use it during the process of warnings generation.

Symbolic name	SCORE-DTEW-EXT-203
Description	This interface is used to retrieve the risk maps
Source	Frontend
Destination	SCORE SIP
Direction	In

Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

4.1.4 Models

The results of the long-term models are dataset that can be filtered by location, time extension, and type.

Symbolic name	SCORE-DTEW-EXT-203
Description	This interface is used to retrieve the data produced by the long-term models
Source	Frontend
Destination	SCORE SIP
Direction	In
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

4.1.5 DTEW data

The DTEW data are dataset related to a specific execution, both Early Warning and simulation. Each dataset is related to a specific set of inputs.

Symbolic name	SCORE-DTEW-EXT-204
Description	This interface is used to manage the DTEW data
Source	Frontend
Destination	SCORE SIP

Direction	In-Out
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

The available actions are described in the following table:

Action	Description
Create	Used by the DTEW to create a new layer with the results of the computation, both Early Warning and simulations.
Retrieve	Used by the GIS-based visualizer to show the results of the computations. Used by the DTEW to retrieve previous computation results to use in the Early Warning context
Update	Not used, a computation result is calculated as new when required
Delete	Used by the GIS-based visualizer to remove the results that are not useful anymore

4.1.6 Sensor data

Sensor data can be retrieved from SIP using the OGC STA. Data can only be retrieved by the system for the following purposes:

Show actual data in the visualization tool

Perform a short-term forecast using both actual and historical data

Symbolic name	SCORE-DTEW-EXT-205
Description	This interface is used to retrieve sensor's data
Source	Frontend
Destination	SCORE SIP
Direction	In
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD

Error codes and Exceptions	TBD
Testing	
Notes	

4.1.7 Weather Forecast

This service is used by the Early Warning Support system as a base to generate the warnings. The forecast is a dataset that can be filtered by location, time extension, and type.

Symbolic name	SCORE-DTEW-EXT-206
Description	This interface is used to retrieve weather forecast
Source	Frontend
Destination	SCORE SIP
Direction	In
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

4.1.8 Weather Historical Data

This service is used to retrieve weather historical data: this data is used by the Early Warning Support system in addition to weather forecasts and data to generate short-term forecasts, along with the related analysis. Data are a dataset that can be filtered by location, time extension, and type.

Symbolic name	SCORE-DTEW-EXT-207
Description	This interface is used to retrieve weather historical data
Source	Frontend
Destination	SCORE SIP
Direction	In
Pre-conditions	
Network Protocol	REST + OGC

Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

4.1.9 Simulation event

This service is used to manage the available simulation events to be used in a simulation.

Symbolic name	SCORE-DTEW-EXT-208
Description	This interface is used to manage events that can be used in simulations
Source	Frontend
Destination	SCORE SIP
Direction	In-Out
Pre-conditions	
Network Protocol	REST + OGC
Data Format	TBD
Data Content	TBD
Error codes and Exceptions	TBD
Testing	
Notes	

The available actions are described in the following table:

Action	Description
Create	Used by the DTEW to create a new layer with the results of the computation, both Early Warning and simulations.
Retrieve	Used by the GIS-based visualizer to show the results of the computations. Used by the DTEW to retrieve previous computation results to use in the Early Warning context
Update	Not used, a computation result is calculated as new when required
Delete	Used by the GIS-based visualizer to remove the results no more useful

4.1.10 Other public interfaces

If requested by some of the project stakeholders, the Early Warning Support can be connected to an external system with the purpose of notifying the generated alerts. The feasibility and the details of the implementation will depend on the considered external system, and the development of the interfaces and eventual data conversion operations will be on the requesting institution.

5. COMMUNICATION PROTOCOLS

In this section, we briefly introduce the protocols and design guidelines that will be used to communicate between the systems involved in the DTEWS platform.

5.1 REST

Representational state transfer (REST) is an architectural style for software design, oriented to network applications, that has been initially conceived as a set of guidelines for the development of applications for the World Wide Web. The REST style has been adopted by the software industry because of its scalability, transparency to the type of exchanged data and interacting components, portability, back-compatibility, security, and reduced latency [4]. It is important noticing that REST is not a protocol specification, but rather a style that, in the last decades, has been largely employed for the design of APIs. The so-called RESTful APIs are compliant with the following architectural constraints:

Client-server architecture: a REST architecture is composed of clients, servers, and resources, and it handles requests through the HTTP.

Uniform interface: This constraint is crucial for the design of RESTful APIs and includes 4 features:

Resource identification in requests: Resources are identified in requests and are separate from the representations returned to the client

Resource manipulation through representations: Clients receive files representing the resources, containing enough information to allow modification or deletion

Self-descriptive messages: Each message sent from the server to a client contains enough information to allow the client processing the resource

Hypermedia as the engine of application state: After the access to a resource, the REST client should be able to discover through hyperlinks all other available actions.

Statelessness: Between requests, no client content is stored on the server, and every information about the session state is held with the client.

Cacheability: Caching is employed to reduce the number of client-server interactions.

Layered system: Client-server interactions can be mediated by additional layers. These layers could offer additional features like load balancing, shared caches, or security.

RESTful web APIs are typically loosely based on HTTP methods to access resources via Uniform resource locator (URL)-encoded parameters and the use of Javascript Object Notation (JSON), hypertext markup language (HTML), or extensible markup language (XML) to transmit data.

5.2 OGC

The Open Geospatial Consortium (OGC) is an international standards organization, active from 1994 [5]. Today, it involves around 500 research, nonprofit, commercial, governmental organizations that are technology providers or users. The OGC members work together on the development, implementation, and dissemination of open interface and encoding standards for geospatial content and services, sensor web and Internet of Things (IoT), GIS data processing and sharing. The collaboration among the OGC members is based on voluntary consensus, to define standards enabling information systems that can easily exchange geospatial information and instructions with other systems.

The scope of OGC activities is very wide and it is divided in Domain Working Groups, ranging from simple geotagging in map images on the Web, to complex control of satellites for Earth observation. The workgroups issue some generalized, high-level description documents called Abstract Specifications, in which basic models for geographic features representation. On the other hand, Implementation Standards documents are also released, where detailed design and implementation technical specifications are reported, always with the aim of allowing interoperability among different technologies and software components employing geospatial data. A list of the current standards and their related documents can be found at [6]. It is worth noticing that most of the OGC standards developed in recent years are OGC Web Services (OWS), defined for the Web services environment.

5.3 STA

The OGC SensorThings API (STA) provides an open, geospatial-enabled and unified way to interconnect to the Internet of Things devices, data, and applications over the Web [7]. The STA is an open Standard, which means it is non-proprietary, platform-independent, and perpetually royalty-free. Although it is a new Standard, it builds on a rich set of proven-working and widely-adopted open standards, such as the web protocols and the OGC Sensor Web Enablement (SWE) Standards, including the OGC/ISO Observation and Measurement data model [8]. This makes the OGC STA also extensible and applicable to simple, as well as to complex use cases.

At a high level, the OGC STA provides two main functionalities and each function is handled by a part. The two parts are the Sensing Part and the Tasking Part. The former provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The latter, on the other hand, provides a standard way for parameterizing, or "tasking", of task-able IoT devices, such as sensors or actuators.

The Sensing part provides functions similar to the OGC SOS and the Tasking part provides functions similar to the OGC Sensor Planning Service (SPS). The main difference between the STA and the OGC SOS and SPS is that the STA is designed specifically for the resource-constrained IoT devices and the web developer community. As a result, the STA follows the REST principles, the use of an efficient JSON encoding, the use of MQTT (Message Queue Telemetry Transport) protocol, and the use of the flexible OASIS OData protocol and URL conventions.

6. SOFTWARE AND TOOLS

6.1 Software repository

A collaborative development approach will be followed for the DTEW frontend development, participating ATU, UCD, MBI, and UNIPI. The code repository is hosted by the UCD Spatial Dynamics Lab in the following GitHub project: https://github.com/SpatialDynamicsLab/Web-GIS-Digital-Twin-SCORE-EU

6.2 Third party software used in the system

The GIS-based frontend component development will be based on three main open-source JavaScript libraries. Leaflet for the 2D digital mapping capabilities, CesiumJS to enable a 3D globe and 3D geospatial data visualisation, and TerriaJS will be used as web GIS development framework to have a web-based geospatial catalogue explorer, for integrating all the geospatial data resources and GIS features and capabilities needed for end-users interaction with the DTEW system.

The virtual representation of real-world environment and its processes is achieved by using specific models. In particular, the DTEW platform system will integrate the hydraulic models provided by the WP3 of SCORE and used to simulate urban flooding scenarios. Specific information and details about the implementation, the input and the output of these models will follow throughout the concomitant development of the DT (see Task 8.2: Development of the Early Warning Support and Digital Twin Platform) and the short-term hazard modelling (see Task 3.4: Short term hazard modelling).

7. REFERENCES

[1] Business Analytics and Optimization—PowerSchedO. (n.d.). MBI. [Internet]. [cited 2022 Jun 06]; Available from:

http://www.mbigroup.it/en/business-analytics-and-optimization/ (Last access 30/07/2022)

[2] SCORE D8.1 - GIS Based Early Warning and Digital Twin Platform functional requirements

[3] SCORE D8.2 - GIS Based Early Warning and Digital Twin Platform system architecture and design

[4] R.T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", PhD Defense Dissertation (https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf) (last access 30/07/2022)

[5] https://www.ogc.org/ (last access 30/07/2022)

[6] https://www.ogc.org/docs/is (last access 30/07/2022)

[7] https://www.ogc.org/standards/sensorthings (last access 30/07/2022)

[8] https://www.iso.org/obp/ui/#iso:std:iso:19156:ed-1:v1:en (last access 30/07/2022)