D3.2 - Platform design document – v2

WP Leader:	**GMV**
Revision Date:	28.02.2024
Submission Date:	29.02.2024
Document Type:	Report
Authors:	Marco Corsi (e-GEOS) Beatriz Revilla (GMV) Paloma Fonseca (GMV) Jorge Pacios (GMV)
Contributor(s):	CENTAUR Consortium
Verified by:	CENTAUR Consortium Board
Approved by:	Valerio Botteghelli (Project Coordinator, e-GEOS) and shared with the Consortium
Version	2.0
Dissemination Level:	Public (PU)
Deliverable #	3.2

Disclaimer

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

History of Changes

Date	Version	Author	Change Description
30.01.2024	0.1	GMV	Second version with a ToC for Tuned and adapted components reference D3.1 - Platform design document ([RD08]) deliverable
29.02.2024	1.0	GMV, ECM, HEN, EG, VIT	First version of the document

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Table of Content

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

LIST OF FIGURES

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

LIST OF TABLES

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

1 EXECUTIVE SUMMARY

The present document represents the deliverable D3.2 – Platform design and development of CENTAUR project and is produced under the Work Package WP3 – Service deployment, in particular, under Task 3.1 that is in charge of:

- Analysis of different services and systems
- Integration of services
- Geospatial analytics

The document includes all the theoretical background related to service design:

- Architecture design
- Platform interfaces
- Components
- A high-level concept of CENTAUR platform

The information included in this document will be the basis for the next deliverables in Task 3.2 and Task 3.3 to consolidate the design and to deliver the prototype version of the platform that allows the integration of WP2 services.

2 INTRODUCTION

2.1 CENTAUR PROJECT DESCRIPTION

Climate change is a fact and its impact on human lives and security is continuously growing. The EU understood the importance and consequences of climate change a long time ago, adopting ambitious legislation in different policy areas. The Green Deal recognises that tackling climate change and striving for climate neutrality should be placed at the centre of societal and economic transformation. Over the last 50 years, more than 11.000 reported disasters related to extreme weather and climate conditions have caused over 2 million deaths and US$ 3,64 trillion in losses. The number of disasters has multiplied by a factor of five during that period, mainly driven by climate and more weather extremes[1]. In particular, the last twenty years have seen the number of major floods more than double, from 1.389 to 3.254, while the incidence of storms grew from 1.457 to 2.034[2]. Floods and storms were the most prevalent events and floods are the most common type of disaster worldwide, accounting for 44% of total events registered in the last twenty years. A global temperature increase of the global climate is estimated to increase the frequency of potentially high impact natural hazard events across the world. This could render current national and local strategies for disaster risk reduction and climate change adaptation obsolete in many countries. In total, between 2000 and 2019, there were 3,068 disaster events in Asia, 1,756 events in the Americas and 1,192 events in Africa.

Climate change is increasingly acknowledged within the EU's integrated approach to security. The related environmental degradation is recognized as a threat multiplier and an aggravating factor for political instability with serious implications for peace and security across the globe[3]. Nowadays, climate change is already causing people to migrate, and while migration should not be directly labeled as a security problem, implicitly the link with

[1] World Meteorological Organization (2021). WMO atlas of mortality and economic losses from weather, climate and water extremes (1970–2019).

[2] UNDRR report: The human cost of disasters: an overview of the last 20 years (2000-2019).

[3] Meyer, C., Vantaggiato, F. P., & Youngs, R. (2021). Preparing the CSDP for the new security environment created by climate change. European Union.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

pressures on society and increased competition for resources are often made[4]. People living in places affected by violent conflict are particularly vulnerable to climate change and it is agreed that some of the factors that increase the risk of violent conflict are sensitive to climate change[5]. This way, it is estimated that 95 % of new displacements by conflicts in 2020 happened in countries that have high or very high vulnerability to climate change[6]. From 2008 to 2016, this represents over 20 million people per year that have been forced to migrate due to climate change effects[7].
Within Copernicus Security and Emergency Services evolution, the objective of CENTAUR is to respond to societal challenges deriving from Climate Change threats by developing and demonstrating new service components for the Copernicus Emergency Management Service (CEMS) and Copernicus Security Service - Support to EU External Action service (CSS-SEA), aiming to:

1. Improve situational awareness and preparedness around climate change and its impact on complex emergencies and multi-dimensional (security) crises;
2. Anticipate the occurrence and possible knock-on effects of crisis events, in particular those triggered by climatic extremes, thus contributing to resilience and effective adaptation.

In the emergency domain, CENTAUR will address the flood-related threats to population, assets and infrastructures in urban areas. In the Security domain, CENTAUR will address water & food insecurity. The two work streams will be connected via a cross-cutting component focusing on exposure and vulnerability to climate change, as well as resilience and societal capacity for managing environmental risks and social conflict. Across work streams, indicators and models will be validated by different methods. CENTAUR will integrate data coming from multiple heterogeneous sources, with a specific focus on those generated by other Copernicus services, and, in particular, those of the Climate Change Service). It will combine these with meteorological data, socio-economic data, and data coming from new sensors (e.g. traditional and social media). Thus, it will enhance current capacities to produce composite risk indexes and to perform multi-criteria analyses in the emergency and security domains.

2.2 SCOPE OF THE DOCUMENT

The WP 3 has the objective of designing and developing the platform, also leveraging and expanding the use of already existing common geospatial standards (e.g. OGC, STAC, openEO), open-source modules and processing chains in the respective domains.

[4] Schaik, L., Bakker, T. (2017). Climate-migration-security: Policy Brief Making the most of a contested relationship. Planetary Security.
[5] W.N., J.M. Pulhin, J. Barnett, G.D. Dabelko, G.K. Hovelsrud, M. Levy, Ú. Oswald Spring, and C.H. Vogel (2014). Human security. In: Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Field, C.B., V.R. Barros, D.J. Dokken, K.J. Mach, M.D. Mastrandrea, T.E. Bilir, M. Chatterjee, K.L. Ebi, Y.O. Estrada, R.C. Genova, B. Girma, E.S. Kissel, A.N. Levy, S. MacCracken, P.R. Mastrandrea, and L.L. White (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, pp. 755-791.
[6] University of Notre Dame. (n.d.). Country index // Notre Dame Global Adaptation Initiative // University of Notre Dame. Notre Dame Global Adaptation Initiative. Retrieved January 23, 2022, from https://gain.nd.edu/our-work/country-index/.
[7] WEF (2020). *The Global Risks Report 2020*, Insight Report 15th Edition. World Economic Forum, Geneva Switzerland, p. 102. https://www.weforum.org/reports/the-global-risks-report-2020.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

2.3 DEFINITIONS, ABBREVIATIONS AND COMPONENTS

Activation
It is an action to start a workflow that produces a product. The activation will contain the parameters needed to start the workflow, such as AOI, start date, end date.
Alert
The platform will provide the user notifications based on the risk indexes produced by Task 2.7 components allocated in the local nodes.
Distributed system
Distributed system involves problem-solving, dividing things into discrete tasks that are interconnected while performing their own operations.
Component
In the current document, component is a software package, a web service or a module that encapsulates a set of related functions. Components communicate with each other via interfaces. Reusability is an important characteristic of a high-quality software component, so that it can be used in different programs.
Event
The platform will provide the user with the updates that the processors trigger. This could be a newly generated product or an update of a product.
Interfaces
The interface refers to the services that a component offers to the rest of the system.
Node
The node is part of a distributed system that can be a computer, a physical server, or a container that can connect to the network and communicate by passing messages.
User story
A user story is an informal, general explanation of a system written from the end user's perspective. Its purpose is to articulate how a user can interact with the system.
Product
A product is either dataset (i.e., the input to an indicator), indicator (i.e., the input to an index) or index ('final' product).

Table 1: Abbreviations and acronyms

Acronym	Description
ACL	Access Control List
AOI	Area of Interest
API	Application Programming Interface
AWS	Amazon Web Services
CEMS	Copernicus Emergency Management Service
COTS	Commercial off-the-shelf software/services
CSS SEA	Copernicus Service in Support to EU External Action
FAIR	Findable, Accessible, Interoperable and Reusable
GIS	Geographic Information Systems
GUI	Graphical User Interface
HTML	Hyper Text Markup Language

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Acronym	Description
HTTP	Hyper Text Transfer Protocol
NDMI	Normalized Difference Moisture Index
OGC	Open Geospatial Consortium
SAR	Synthetic-aperture radar
SSO	Single Sing-on
STAC	Spatial Temporal Asset Catalog
UF	Urban Flood
WFS	Context indicator: Water & Food Security Context Web Services: Web Feature Service
WMS	Web Map Service

2.4 APPLICABLE AND REFERENCE DOCUMENTS

Table 2: Applicable and reference documents

ID	Document name
[RD01]	Copernicus Service in Support to EU External Action: https://sea.security.copernicus.eu/
[RD02]	Disaster Risk Reduction in EU external action - Council conclusions (28 November 2022): https://data.consilium.europa.eu/doc/document/ST-14463-2022-INIT/en/pdf
[RD03]	D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 15/06/2023
[RD04]	D2.1 - Catalogue of CENTAUR data and related specifications
[RD05]	D2.2 - Urban Flood and Water & Food Insecurity Design
[RD06]	D2.4 - Urban Flood and Water & Food Insecurity service pipelines
[RD07]	D2.5 - CENTAUR multi-criteria indexes design
[RD08]	D3.1 - Platform design document
[RD09]	D3.3 - CENTAUR integrated platform including Urban Flood and Water&Food Indexes v1 (baseline)

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

3 ARCHITECTURAL DESIGN

3.1 HIGH-LEVEL CONCEPT

CENTAUR platform concept is a cooperative platform with a distributed architecture based on microservices. Several components have been designed separating the capabilities of the system to fulfil the user requirements as described in D1.1 ([RD03]). The core objective of the CENTAUR platform is described in Section 3.2 General principles.

The design is based on a central node (Figure 3-1) that interacts with several local nodes. Each local node will be in charge of generating the different products as defined in D1.1 ([RD03]).

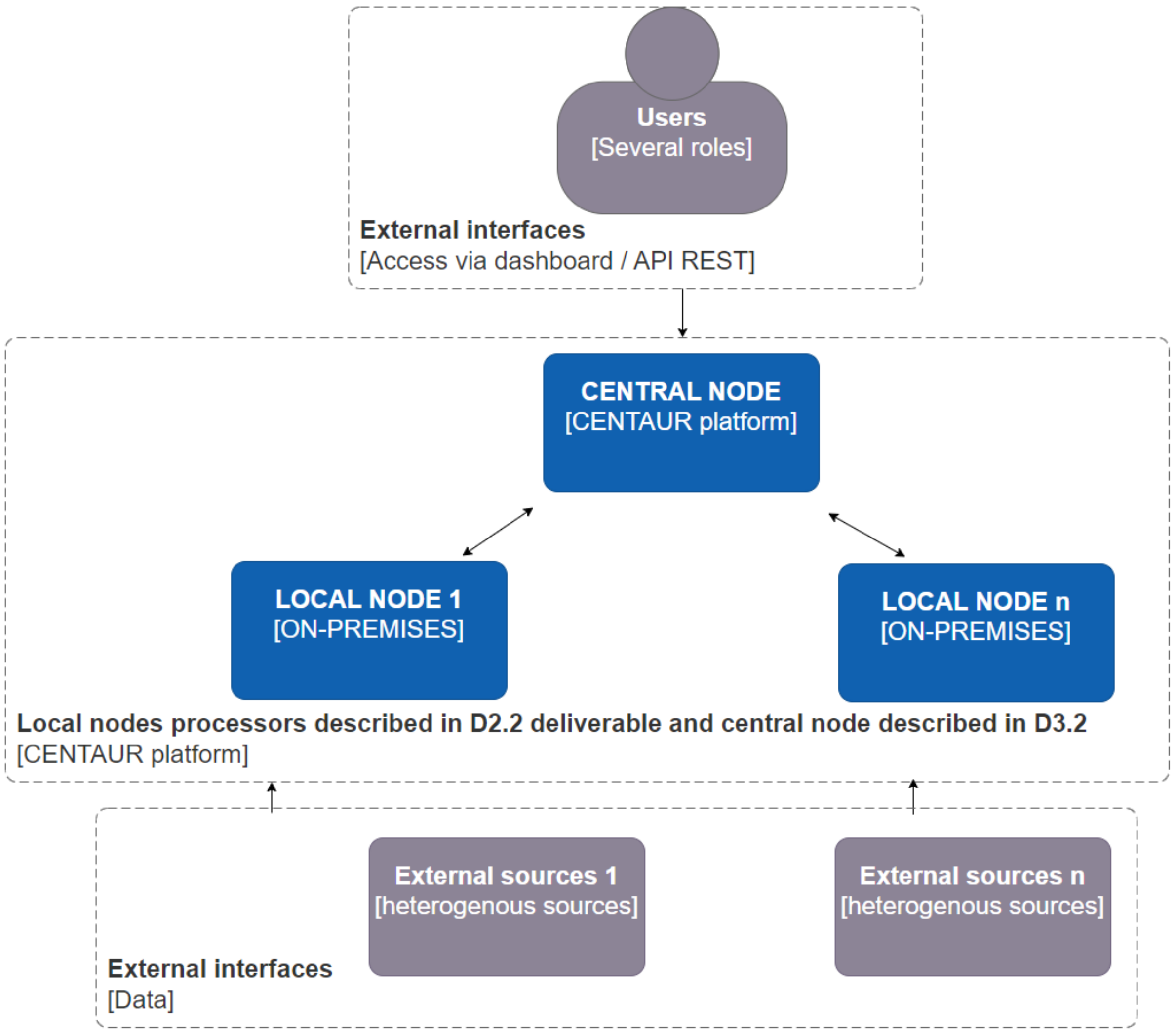


Figure 3-1: High-level concept

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

The CENTAUR platform will include all components distributed across one central node and multiple local nodes. The platform will operate in a multi-cloud infrastructure: the central node will be allocated in AWS and the different local nodes on premises. Each partner will contribute to the overall performance of the platform.

The data flow can be described as follows:

- Central node:
    - Input flow from a local node: output from the services allocated in the node or the local node can use the REST API to search for products in CENTAUR catalogue.
    - Output flow towards a local node: a request to the service allocated in the node.
    - All services of the platforms will be accessible via a dashboard.

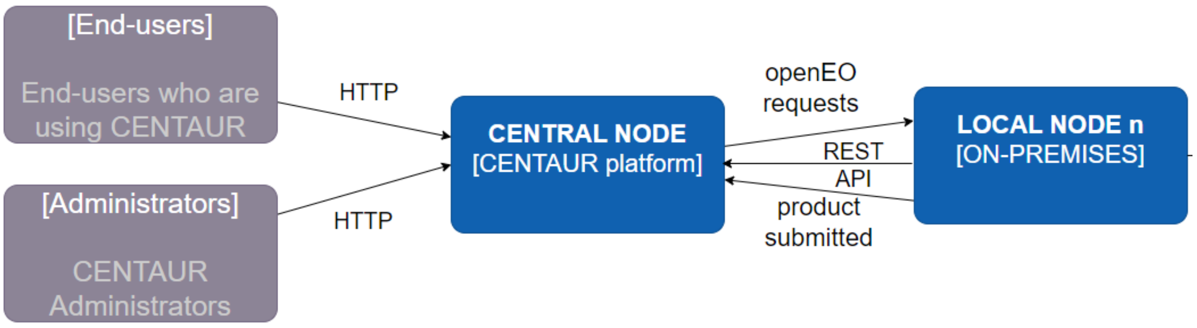


Figure 3-2: Central node interfaces

- Local node:
    - Input flow from the central node: activation request.
    - Output flow to the central node: search for a product and submit products.
    - Input from external sources or datasets that are required for the underlying services.

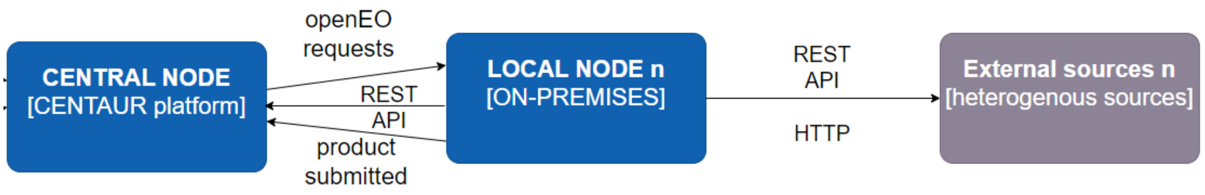


Figure 3-3: Local node interfaces

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

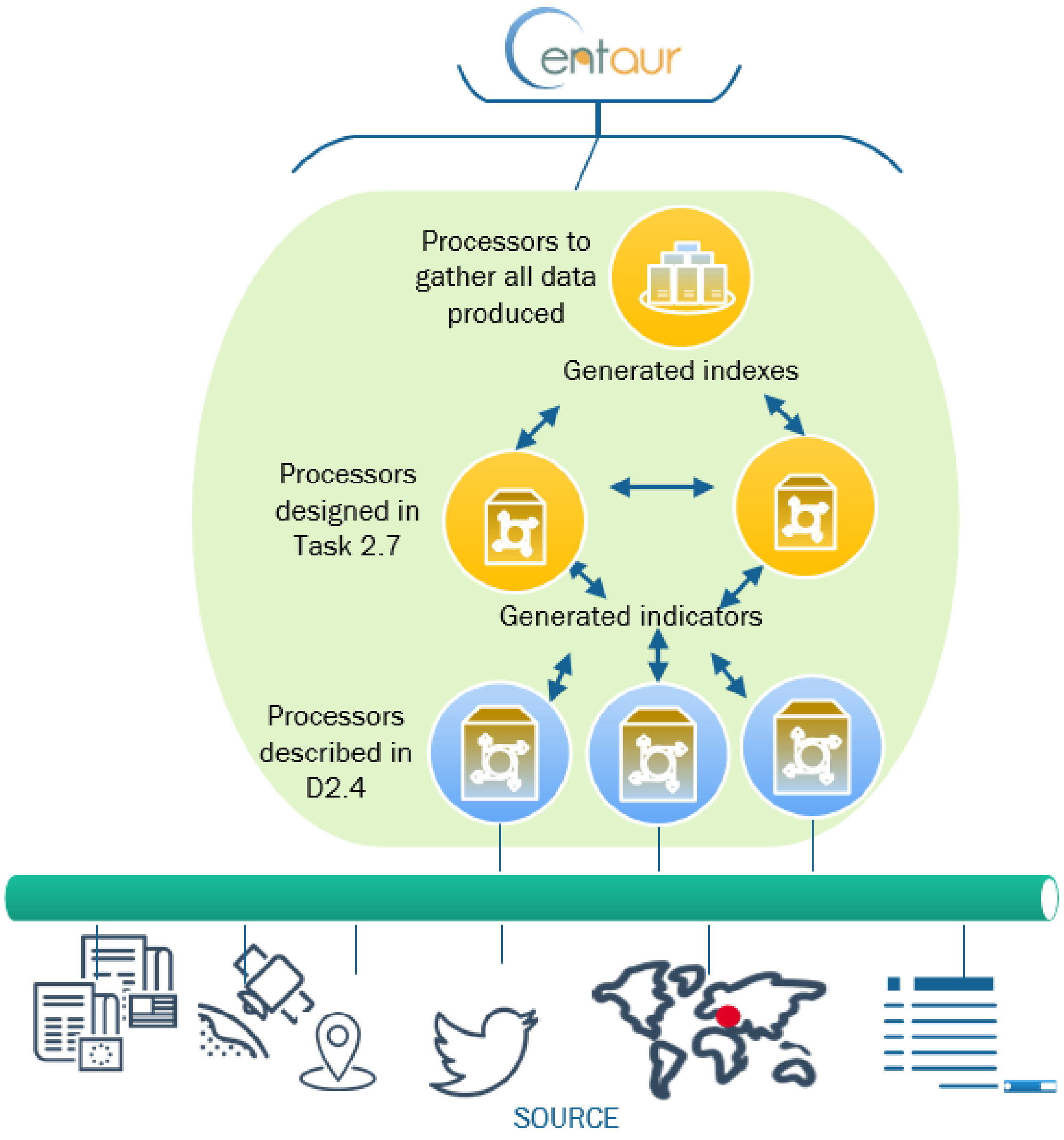

Processors to gather all data produced
Generated indexes
Processors designed in Task 2.7
Generated indicators
Processors described in D2.4
SOURCE

Figure 3-4: Connection between nodes

3.2 General principles

The following sections describe the main drivers that have been used to design the CENTAUR platform.

3.2.1 Distributed platform

The design has a central node that coordinates multiple local nodes. The number of local nodes is not fixed, the platform design is adjusted to increase or decrease the local nodes connected to the central node. The strategy to build the CENTAUR platform is an iterative process, based on this capability, pipelines from WP2 will be

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

seamlessly phased into the platform or removed in favour of a more comprehensive chain that meets the user's needs.

3.2.2 Self-standing components

The design of the components has been considered to avoid dependencies between them. A well-defined interface has been established to link between components, see section 5.

3.2.3 Scalability, Reliability and Resilience

The selection of cloud-native technologies for the design of the system implies the possibility of supporting scalability, reliability and resilience thanks to its auto-provisioning, auto-scaling, and auto-redundancy capabilities, in compliance with requirement GR-02 and PR-02 see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

Diving into the nodes that shape the platform:

- The selected COTS that build the central node are scalable, further details are in section 3.4.
- It is foreseen that the number of nodes will be dynamic, to be removed or included as needed. Further details about their design and the implementation are described in the scope of WP2 D2.4 – Urban Flood and Water & Food Insecurity service pipelines ([RD06]).

3.2.4 Prioritize open-source usage

The use of open source has been prioritized in the design as much as possible accounting for existing concepts, standards, and software.

3.2.5 Supporting “what if” analysis

Supporting “what if” analysis is provided in the CENTAUR platform. This functionality is designed and implemented in the local nodes, see WP2 deliverables for further information, D2.4 – Urban Flood and Water & Food Insecurity service pipelines ([RD06]).

3.2.6 Microservices approach

The design of all components has been designed using a microservice approach to be deployed in a containerized form on a Kubernetes cluster, with each service required for public-facing operations running in a distributed form designed to provide scalability, high availability and rolling restarts during upgrades.

The use of a container orchestration engine as Kubernetes provides the connection to the external world, schedules tasks, load balance between the available infrastructure, scalability, and resilience, in compliance with requirement GR-02, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

In addition, this type of architecture provides the following advantages:

- Velocity: the deployment of a new application version takes minutes and according to the replacement policy could enable service zero-downtime.
- Immutability: The software is packed in with a specific structure that should not be modified thanks to the possibility of signing it. In this way, it is not possible to modify a deployed service.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- Scalability: thanks to the microservices design approach, the number of instances associated with a specific service can be increased to solve a peak demand of a specific activity.
- Infrastructure abstraction: the container abstracts the development of the service from the infrastructure making the software deployment between different platforms easier.

3.2.7 Security and Authentication

The central node is the component of the CENTAUR platform that provides the user a single access to the high-functionalities in compliance with requirement PR-08, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]). This node is deployed in AWS infrastructure, for access to the cloud environment it is using AWS Identity and Access management[8].

The central node has a GeoNode component in charge of managing: access, service catalogue and viewer capabilities, its architecture is based on Oauth2 Security[9] and it has a GeoServer Security Backend and GeoNode Security Backend[10], see further details in section 6.1.2.

GeoNode can be configured to run on production mode which enables SSL, encrypting traffic between GeoNode server and client browsers[11], in compliance with requirement GR-01, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

3.2.8 Sensitive data handling

The CENTAUR platform is not available for registration. An operator with an administration role will manage the creation of the user and assign it to the relevant group, see section 3.3. The personal data stored related to the end-user has the purpose of notifying the availability of data, in compliance with requirement IR-01, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]), no further personal data will be stored assuring the compliance with requirement GR-03, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

The products are generated based on AOI that are not sensible or military zones from France, Germany, Italy, Mali, Mozambique, Somalia, and Sudan.

3.2.9 Interoperability standards and protocols

With regards to the external interfaces to submit notifications or dissemination of the products are chosen implementation based on standards to enhance communication efficiency and facilitate seamless integrations and interoperability between different systems.

- Notifications to the end-user are based on SMTP protocol, to handle the transmission of emails, in compliance with requirement IR-01, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).
- Dissemination of products is based on GeoNode and GeoServer RESTful services that are chosen as components in the central node, see section 3.4, in compliance with requirement IR-02, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

[8] https://aws.amazon.com/es/iam/
[9] https://docs.geonode.org/en/master/advanced/components/index.html
[10] https://docs.geonode.org/en/master/advanced/index.html
[11] https://cs-geonode.readthedocs.io/en/2.8_a/tutorials/advanced/geonode_production/ssl.html

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- Through the end-user interface deployed in the central node, products can be downloaded and uploaded following standard formats, such as tiff, kml, geojson, in compliance with requirement IR-03, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).
- The CENTAUR platform is generating enriched products based on external providers, see WP2 deliverables for further information, D2.4 – Urban Flood and Water & Food Insecurity service pipelines ([RD06], in compliance with requirement IR-04, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

The CENTAUR platform uses standards for internal interfaces as described in sections 5 and 6.3.

3.2.10 Data format inconsistencies between the different input data systems

The CENTAUR platform manages a wide range of data to generate enhanced products, more precisely through the components allocated in the local nodes, see section 6.2. Furthermore, generated products in the CENTAUR platform have been analysed and principles have been established to follow standards, see section 6.2, as GeoServer[12] has a wide variety of types of stores this can be increased if needed.

3.3 USERS AND PRIVILEGES

Based on the requirements defined in D1.1 ([RD03]), two types of users were identified, that will interact with the CENTAUR platform:

- **End-users** will have restricted access to data and limited capabilities, such as performing scanning, filtering, subscribing to events, or using predefined areas of interest. End-users will be created by an administrator.
- **Administrators** will have access to all data and perform advanced functionalities, such as creating areas of interest or managing users.

3.4 TECHNOLOGIES AND PRE-EXISTING COMPONENTS

The use of COTS-based (Commercial off-the-shelf software/services) solutions is increasingly widespread.

This section includes the list of COTS that will be used during the project lifecycle.

3.4.1 Docker

Open-source containerization technology for building and containerizing applications. Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.

3.4.2 Kubernetes

It is an open-source system for automating deployment, scaling, and management of containerized applications (also deployable as an AWS service).

[12] https://docs.geoserver.org/2.23.x/en/user/data/webadmin/stores.html

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

3.4.3 FastAPI[13]

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. Used in Data Requester.

3.4.4 Django[14]

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Used in Access management.

3.4.5 Rasterio[15]

Library that reads and writes GIS rasters and provides a Python API based on Numpy N-dimensional arrays and GeoJSON.

Used in:

- Urban inundation probability maps and water depth.
- Urban flooding map based on geomorphological / InSAR approach for enhanced damage assessment.
- Hazard web sources indicator.

3.4.6 GeoPandas[16]

The goal of GeoPandas is to make working with geospatial data in Python easier. It combines the capabilities of pandas and shapely, providing geospatial operations in pandas and a high-level interface to multiple geometries to shapely. GeoPandas enables operations in Python that would otherwise require a spatial database such as PostGIS.
Used in:

- Urban inundation probability maps and water depth
- Urban flooding map based on geomorphological / InSAR approach for enhanced damage assessment
- Hazard web sources indicator

3.4.7 GeoNode[17]

This is one of the central node components that will manage: access, service catalogue and viewer capabilities. As part of GeoNode is installed GeoServer[18], high-availability, scalability and performance are provided[19].

[13] https://fastapi.tiangolo.com/
[14] https://docs.djangoproject.com/en/3.2/topics/auth/
[15] https://rasterio.readthedocs.io/en/stable/
[16] https://geopandas.org/en/stable/
[17] https://geonode.org/
[18] https://geoserver.org/
[19] https://docs.geoserver.geo-solutions.it/edu/en/clustering/clustering/introduction.html

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

3.4.8 openEO[20]

Standard used to exchange information between nodes.

3.4.9 STAC[21]

Standard used to exchange information on data from UF/WFS processing pipelines to the central node.

4 STATIC ARCHITECTURE

The CENTAUR platform follows the next static diagram (Figure 4-1). There are 2 types of nodes, the central node and the local nodes.

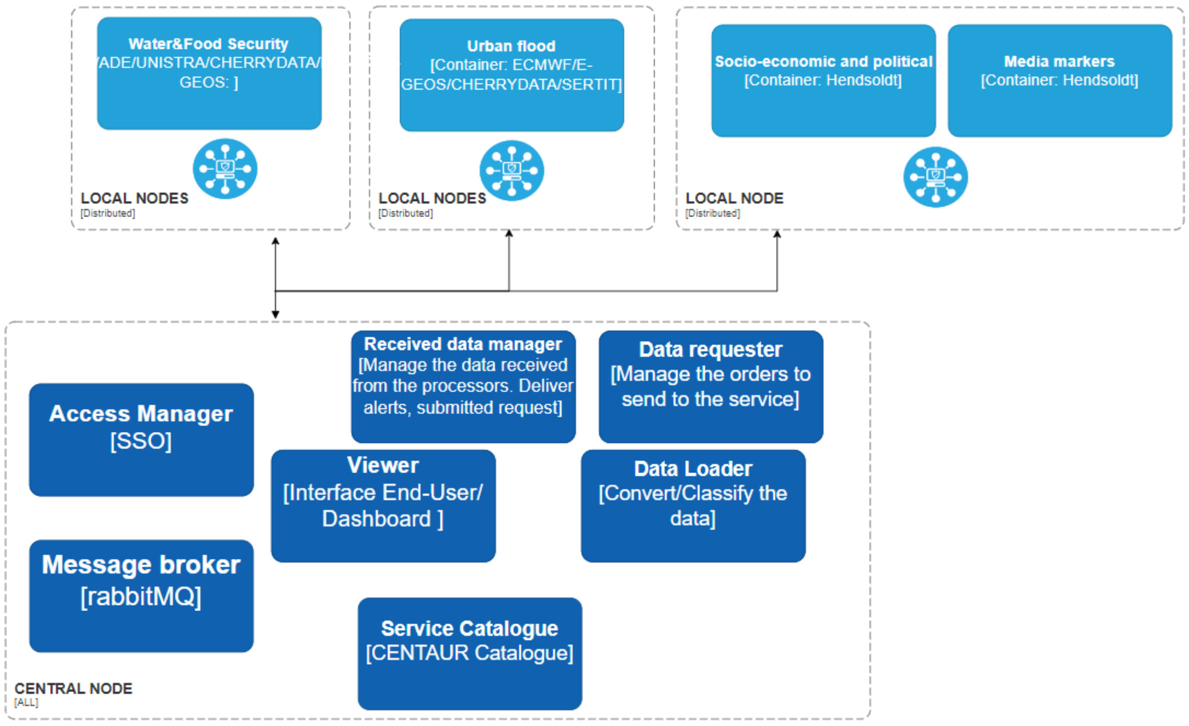


Figure 4-1: Static architecture

There is one central node and multiple local nodes. Local nodes are meant to be scalable, to add or remove them without interfering with the performance of the platform.

The central node includes the following capabilities:

- **Access manager** verifying the user identity. All groups and users will be managed by an administrator.
- **Data requester**, managing the requests to specific services (allocated in a local node.)

[20] https://openeo.org/documentation/1.0/python/#collections
[21] https://stacspec.org/en

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- **Data loader,** managing how the data is ingested in the platform.
- **Service catalogue**, the catalogue which includes the relevant information for Water & Food security and Urban Flood domains.
- **Viewer,** viewer component that implements the user interface, this is the entry point to the end-user.
- **Received data manager** receives the product generated on-demand by the local nodes and manages the notifications.
- **Message broker,** to exchange messages between components.

The local nodes are responsible for running the services and delivering the results to the central node. All data (required as input for or generated by the corresponding services) is listed in deliverable D2.1 - Catalogue Centaur.

5 PLATFORM INTERFACES

The platform interfaces are mainly divided into the following:

- Internal interfaces between the components in CENTAUR platform. These interfaces will be implemented using openEO API that allows:
    - Unified and straightforward access to multiple Earth observation datasets.
    - Scalable and efficient processing capabilities.
    - A standardized system that works across different platforms.
    - Independence from underlying technologies and software libraries.
    - Reproducibility through transparent workflows, supporting principles of FAIR (Findable, Accessible, Interoperable, and Reusable) data and Open Science.

  The openEO API will be used for the connection between components as described in Section 6.2. The endpoints that will allow to manage processes, data and job execution are the following:
    - Capabilities will provide the capabilities supported.
    - Data Discovery will discover available data collections and their metadata.
    - Data Access will access data or metadata from a specific collection.
    - Process Graph Execution will execute a process graph which is a sequence of operations to be applied to the data.
    - Jobs management to retrieve the status of submitted jobs.
    - Job results to access the result of a specific job.
- External interfaces to external data providers as listed in D2.1 – Catalogue of CENTAUR data and related specifications ([RD04]) and the end-user via HTTP, using the platform dashboard. An administrator will access the platform with advanced privilege using the AWS interface (SSH, HTTP…)

6 COMPONENTS DESCRIPTION

6.1 CENTRAL NODE COMPONENTS

6.1.1 Introduction

The central node allocates components to interface with the end-user, coordinate the request the activation of the processors needed, and the central repository of the generated products.

An instance of GeoNode will be the kernel of all the capabilities that the central node provides. Following is a list of components that make up the central node:

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- GeoNode will provide:
  - **Viewer**: User interface to the end user implemented by MapStore.
  - **Access manager**, included in GeoNode based on Django framework.
  - **Service catalogue**: GeoServer as the geospatial backend server, which will be CENTAUR catalogue.
  - Postgres database to store GeoNode information: such as resources, and users.
  - REST API to manage GeoNode information and expose data.
  - **Message broker**, based on RabbitMQ to exchange messages between components.
- **Data loader**, service to ingest products: based on Python. This component will use GeoNode REST API to manage the CENTAUR catalogue.
- **Data requester**, service to request the activation of the processors allocated in the local nodes. It will be based on Python for the request action and Postgres to store the data.
- **Received data manager**, service to manage the events and the alerts. An event will be triggered by the calculation of a processor and the alert will be triggered by the generation of indexes.

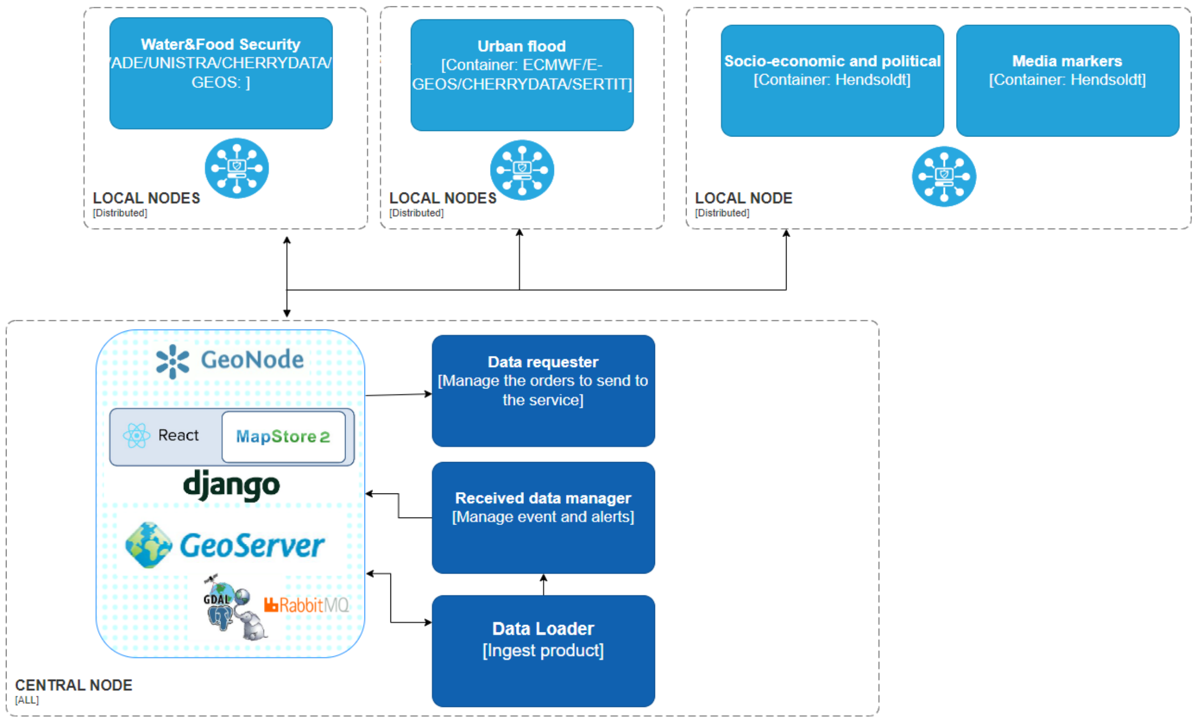


Figure 6-1: Central node components

6.1.2 GeoNode

The CENTAUR platform is a Django project based on GeoNode. GeoNode is an open-source project, based on Django framework for the backend, MapStore for the user interface and GeoServer as a geospatial server.[22]

Once an instance of GeoNode is created, to build CENTAUR platform it is needed to customize it by adding functionality and updating the end user interface. GeoNode's default language is English, but it can be changed[23],

[22] GitHub repository (https://github.com/GeoNode/geonode).
[23] https://cs-geonode.readthedocs.io/en/2.8_a/tutorials/admin/default_lang/index.html?highlight=language

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

in compliance with requirement PR-11, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

GeoNode dockers to run the instance are:

- Django docker that includes the set-up of the front-end related to the CENTAUR project. MapStore client for GeoNode has a user-friendly graphical user interface, in compliance with requirement PR-01 see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).
- GeoServer docker that includes GeoServer service.
- Postgis docker that manages the database.
- Nginx docker that provides HTTPS server capabilities and is used to reverse proxy, load balancing, and caching.
- Celery docker for asynchronous task queue/job queue based on distributed message passing.
- RabbitMQ docker that facilitates communication and data exchange between components.

6.1.2.1 GeoServer

GeoServer publishes data from any major spatial data source using open standards. GeoServer implements several Open Geospatial Consortium protocols including Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) and Web Map Tile Service (WMTS). Additional extensions are available for Catalogue Service (CSW) and Web Processing Service (WPS).

In addition, extensions[24] or community modules can be integrated to add functionality to the CENTAUR platform such as:

- STAC data store[25]: Create a vector data store that can connect to a STAC API, delivering collections as feature types and items as features.

6.1.3 Data requester

This component has the capability to request the activation of a service. It is made up of:

- Data requester front-end integrated into the user interface GeoNode, where the user could request the activation of a service. The form to be filled will contain the necessary parameters to be sent to the services, for example: AOI, start date, and end date.
- Data requester back-end based on Python. Using openEO for the activation will be sent.

It will be a new capability of the GeoNode instance.

6.1.4 Received data manager

This component is in charge of managing events and alerts. It will be made up of:

- Received data manager front-end component: this will include the needed user interfaces to display the events and alerts.
- Received data manager back-end component: this will implement the logic of the events and alerts. Events and alerts are generated by the components in the local nodes. These elements will be stored in the database managed by this component. The features to be handled by this component will be:

[24] https://docs.geoserver.org/2.23.x/en/user/extensions/index.html#extensionsl
[25] https://docs.geoserver.org/2.23.x/en/user/community/stac-datastore/data-store.html

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- Store an event: An event will be stored with the following attributes, Figure 6-4.
- Trigger predefined events. The event has to be defined by each component allocated in the local nodes. The event will be sent through RabbitMQ as a message. A preliminary set of messages will be defined and triggered to the end-user automatically, such as the update of a dataset or new dataset available. The automatic message involves the workflow between Received data manager and Data Loader in charge of the ingestion of the products in the catalogue, see Figure 6-2.

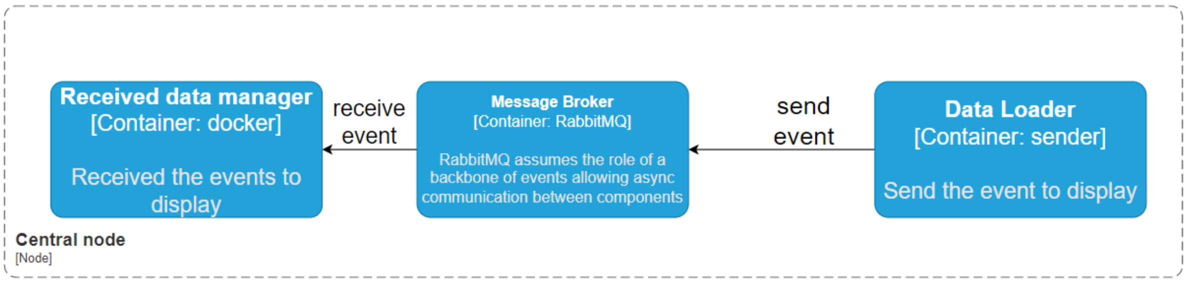

Received data manager
[Container: docker]
Received the events to display
receive event
Message Broker
[Container: RabbitMQ]
RabbitMQ assumes the role of a backbone of events allowing async communication between components
send event
Data Loader
[Container: sender]
Send the event to display
Central node
[Node]

Figure 6-2: Received data manager and Data Loader interface

- Trigger events. Events will be sent by components allocated in local nodes through Message Broker Figure 6-3, a RabbitMQ docker.

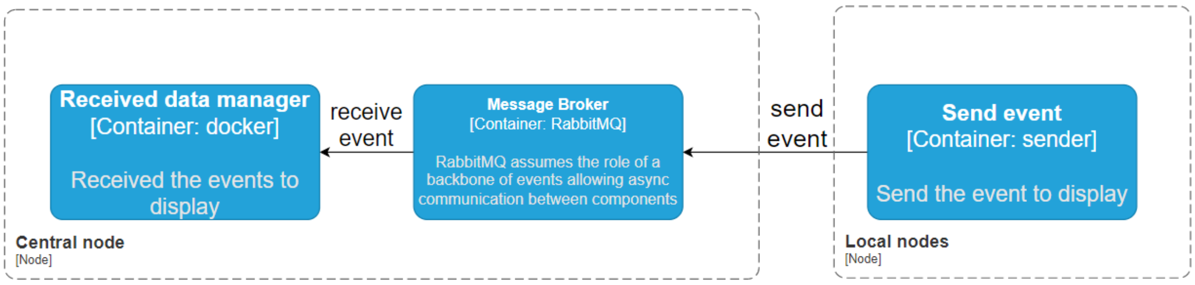


Figure 6-3 Event flow

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

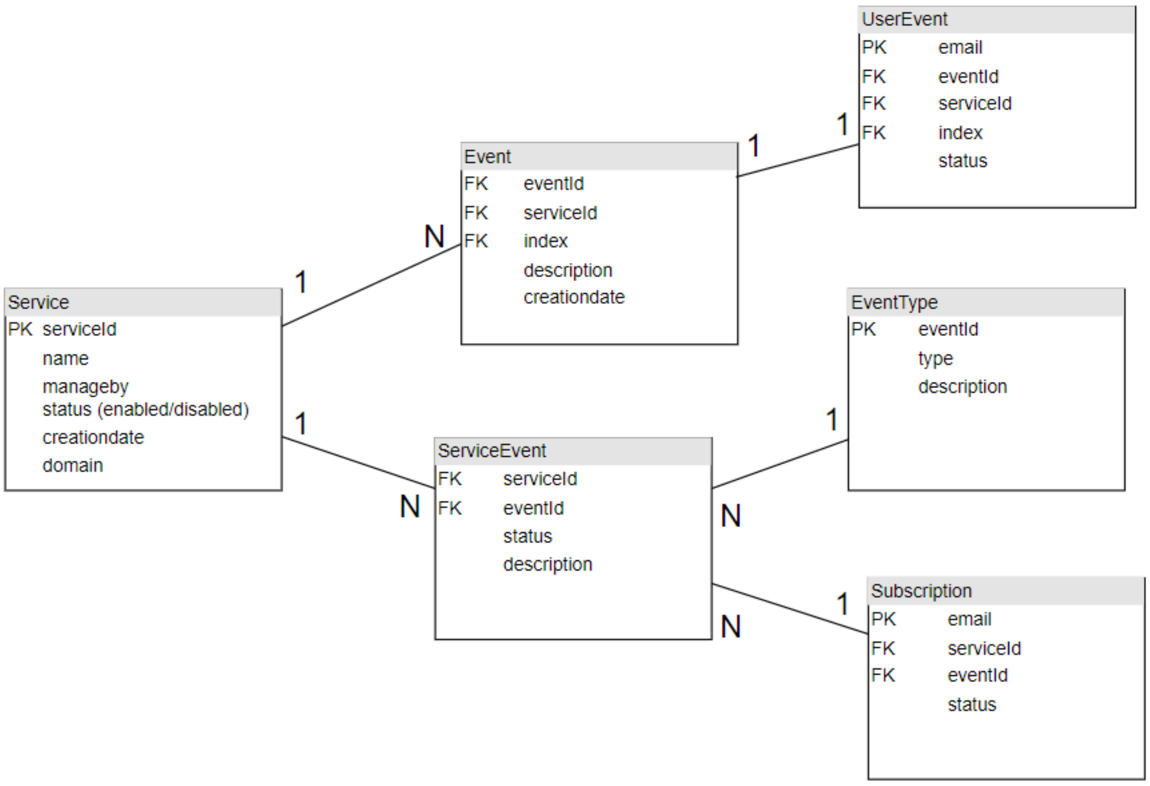
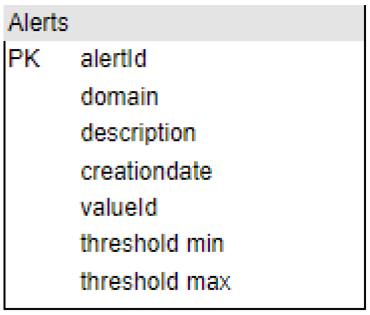


Figure 6-4: Event attributes

- Store alerts: An alert will be stored with the following attributes Figure 6-5.
- Manage alerts: Alerts will be sent from local nodes designed in WP2, Task 2.7 (D2.5 - CENTAUR multi-criteria indexes design, ([RD07])), created by the combination of indicators.



Alerts	
PK	alertId
	domain
	description
	creationdate
	valueId
	threshold min
	threshold max

Figure 6-5: Alert attributes

- Subscription to events. User will be available to display events and alerts. A user will be able to receive via email events by subscription, see Figure 6-4.
  - Service will store the information about the services available in the CENTAUR platform.
  - ServiceEvent will store the list of events that a service can notify.
  - EventType will store the available types of events.
  - Event will store the launched event from a service.
  - UserEvent will store the status of the event notified to an email.
  - Subscription will store the emails that has to be notified by a service event.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Alerts and events will be subject to updates in the final deployment as Task 2.7 is running in parallel in the edition of this document.

It will be a new capability of the GeoNode instance.

6.1.5 Data Loader

This component will be developed in Python using GeoNode REST API to manage CENTAUR catalogue. It will be made up of:

- Monitoring component: it has access to S3 bucket to acquire the products and has an interface with Message broker to receive notification from local nodes.
- Ingestion handler component: it will use GeoNode REST API to insert or update CENTAUR catalogue, in compliance with requirement DP-01, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).

It will be a docker with access to GeoNode instance.

6.2 LOCAL NODE COMPONENTS

6.2.1 Introduction

Local node components are several processors that will be in charge of producing the indicators and indexes to enhance Urban flood and Water & Food security domains in the AOI, which are the scope of CENTAUR project: France, Germany, Italy, Mali, Mozambique, Somalia and Sudan, in compliance with requirement DP-02, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]). The pipelines and workflows that are performed inside each local node are explained in detail in the deliverables of WP2, which includes the models used to process data using AI-trained algorithms and tools needed, in compliance with requirements DP-07, DP-08, and DP-09, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]). This document is focused on explaining the method to acquire, store and display to the user the output of the processors. The local nodes are split into 3 main domains:

- Water & Food Security
- Urban flood
- Socio-economic and political

6.2.2 Water & Food Security

Water & Food Security indicators will be generated by six data processing components. Dependencies between these components might exist (e.g. meteorological forecasts produced by ECMWF are required by the VITO processing component), processing chains are described in D2.2 – Urban Flood and Water & Food Insecurity Design ([RD05]) deliverable.

Meteorological drought products (i.e., WFS–ID–1, WFS–ID–2, and WFS–ID–3) will be made available via a generic openEO API, allowing users to retrieve global products. Going forward, we aim to include ECMWF generated indicators within the ECMWF Product Generation System that could be made available as a customizable product through the Product requirements catalogue. This product could then be automatically delivered when available. Given the complexity of the calculation and the data required, we do not foresee the possibility of using a docker for the production of these layers.

For further description of the processors allocated in the local nodes, see D2.2 – Urban Flood and Water & Food Insecurity Design ([RD05]) where the workflows of Water & Food security are described in more detail.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

6.2.3 Urban Flood

Urban Flood processors and workflows are detailed as part of WP2, these processors will be allocated in local nodes detailed in section 3.1. Urban flood indicators in CENTAUR are calculated from autonomous components hosted in on-premises environments. The central node forwards indicator processing requests to the corresponding components, which in turn notify the central node about the availability of new outputs in the central storage.

6.2.4 Socio-economic and political indicators

Indicators of socio-economic stress and vulnerability relate to both thematic areas of the project - urban floods (UF-ID-8 to UF-ID-14) and water and food security (WFS-ID-11 to WFS-ID-25). The corresponding analytical services will be made available preferably as self-contained dockerized components accessible via a standardized REST API interface to allow for easy integration and high scalability. The output (indicators) is expected to be of a quantitative and robust nature facilitating follow-up fusion with other services/indicators.

6.2.5 Local Nodes Description

As described in previous sections, local nodes will allocate the components described in D2.2 – Urban Flood and Water & Food Insecurity Design ([RD05]) deliverable. Each local node, as a set of components, will produce the output autonomously. Hereunder are detailed features that apply to all local nodes:

- Component, see 6.2.5.1.
- Activation, see 6.2.5.2.
- Output, see 6.2.5.3.

6.2.5.1 Component

A component is a set of processors that produces products. There are 2 types of components. Both of them submits products: ones are indicators, these processors are described in detailed in D2.2 – Urban Flood and Water & Food Insecurity Design ([RD05]), and the others are indexes, processors designed as part of Task 2.7 activities, see D2.5 - CENTAUR multi-criteria indexes design, ([RD07]).

6.2.5.2 Activation

The local nodes will produce products in two different ways depending on how the processors allocated in the node work. While some processors are working continuously and delivering products to the central node, other processors will need an order to be activated.

- Continuously: Local node that allocates processors that do not need to be activated to start generating products. The products generated will be submitted to an S3 bucket and notified to the central node for its ingestion.
- On-demand: Local node that allocates processors that need to be activated to start generation products. The central node will request the activation using openEO, see 6.3

6.2.5.3 Output

The products generated by the local nodes are:

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- Indicators, produced continuously or on-demand.
- Indexes, indicators will be used combined to generate integrated crisis indexes, detailed in deliverables related to Task 2.7, see D2.5 - CENTAUR multi-criteria indexes design, ([RD07]).

Both indicators and indexes will be available in the platform, through the central node. The visualization of each product will depend on the format. Deliverable D3.3 - CENTAUR integrated platform including Urban Flood and Water&Food Indexes v1 (baseline) ([RD09]) and in a final version, deliverable D3.4 will contain how each product is displayed to the end-user.

The output is submitted to S3 bucket as a temporal repository where Data Loader will ingest it, see 6.1.5.

6.3 NODE INTERFACES

Two use cases are defined to interact between the central node and the local nodes:

- To activate a processor allocated in a local node, using openEO, see. 6.3.1, in compliance with requirement DP-09, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]).
- Temporal repository to allocate the product generated, as an asynchronous response to an order, the product generated will be submitted in an S3 bucket by the processors, see 6.3.2. Along with the product, an event could be triggered using Message Broker component.

Between local nodes, the interfaces are defined in WP2 where the pipelines are described, see deliverable D2.2 – Urban Flood and Water & Food Insecurity Design ([RD05]).

6.3.1 openEO interface

The internal interfaces will be based on openEO standard, Table 3 lists the endpoint that the central node will use to required information.

Table 3: Component interfaces

Endpoint	Description	Comments
/	GET method	(basic openEO endpoint)
	Information about openEO	
/.well-know/openeo	GET method	(basic openEO endpoint)
	Support openEO versions	
/collections	GET method	Based on STAC API
	List of all available data collections and their metadata	
/collections/{collection_id}	GET method	Based on STAC API
	Access data or metadata from a specific collection	
/processes	POST method	This endpoint will return only custom processes for CENTAUR project (not the ones usually defined in openEO)
	List all processing services	

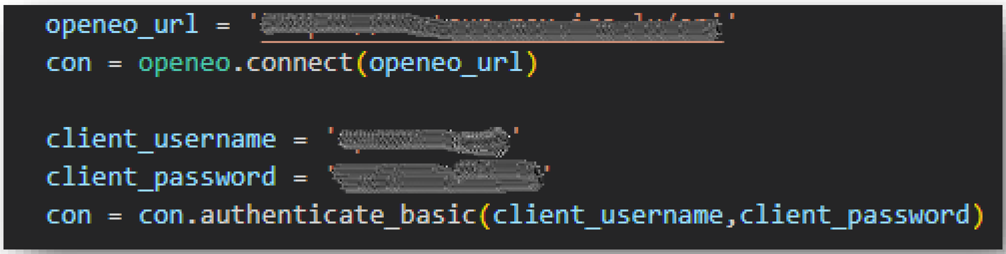
Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

/jobs	GET method	
	List all batch-processing jobs submitted	
/jobs	POST method	
	Create a new batch-processing job	
/jobs/{job_id}	GET method	
	Full metadata for a batch-processing job	
/jobs/{job_id}/results	GET method	
	List batch-processing job results	
/jobs/{job_id}/results	POST method	
	Start a batch-processing job	
/credentials/basic	GET method	This endpoint could be replaced with the SSO of the platform
	HTTP Basic Authentication	

It established a common pattern to request information between central node and local node, the next steps identified are:

- Create connection



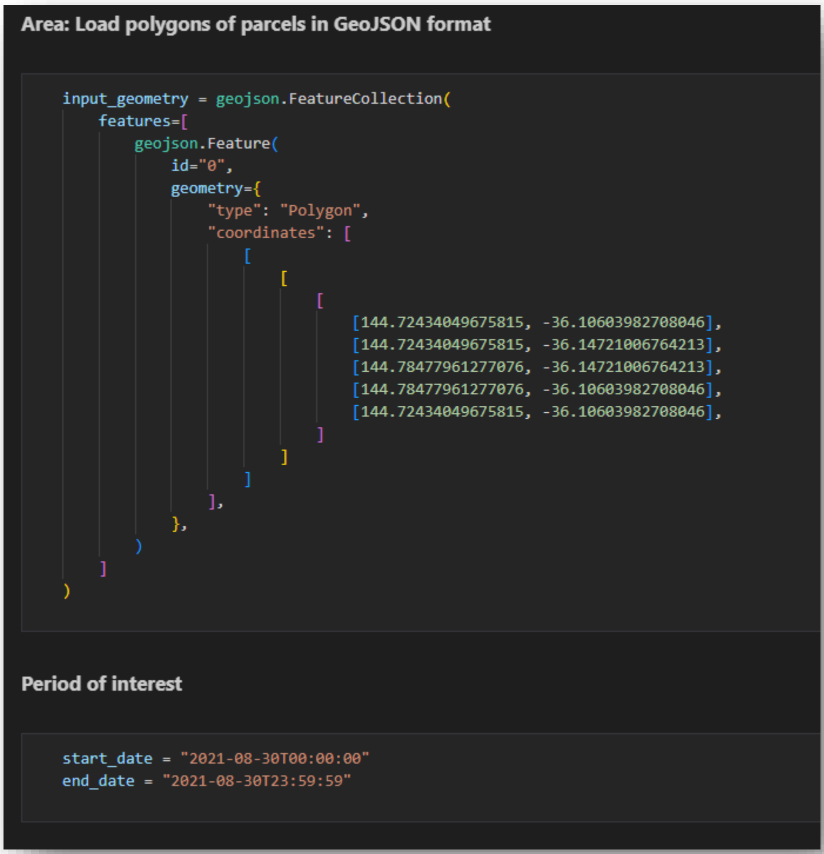
```
openeo_url = '[redacted]'
con = openeo.connect(openeo_url)

client_username = '[redacted]'
client_password = '[redacted]'
con = con.authenticate_basic(client_username,client_password)
```

Figure 6-6: openEO – Create a connection

- Define the parameters of the request

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR



Area: Load polygons of parcels in GeoJSON format

```
input_geometry = geojson.FeatureCollection(
    features=[
        geojson.Feature(
            id="0",
            geometry={
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            [
                                [144.72434049675815, -36.10603982708046],
                                [144.72434049675815, -36.14721006764213],
                                [144.78477961277076, -36.14721006764213],
                                [144.78477961277076, -36.10603982708046],
                                [144.72434049675815, -36.10603982708046],
                            ]
                        ]
                    ]
                ],
            },
        )
    ]
)
```

Period of interest

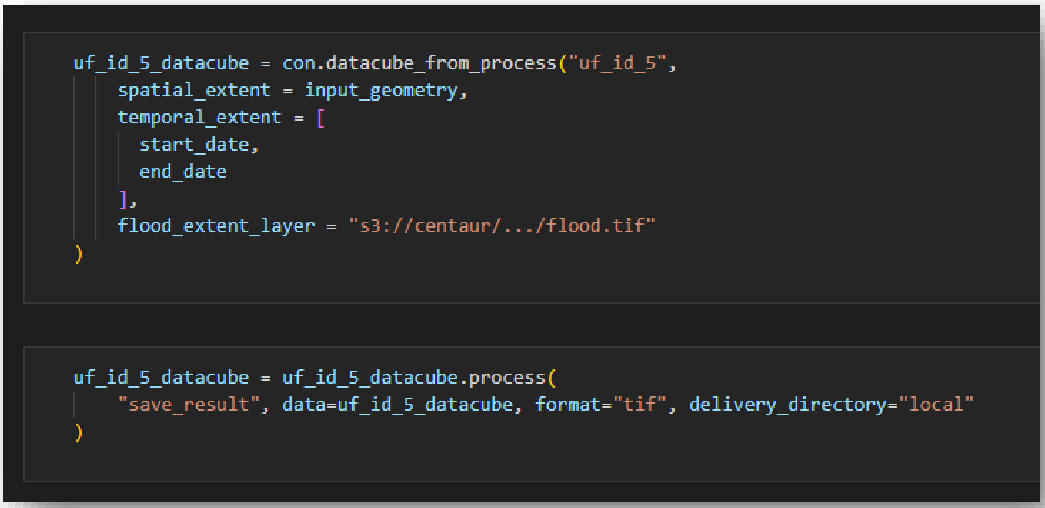
```
start_date = "2021-08-30T00:00:00"
end_date = "2021-08-30T23:59:59"
```

Figure 6-7: openEO – Parameters definition

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

- Identify the processor

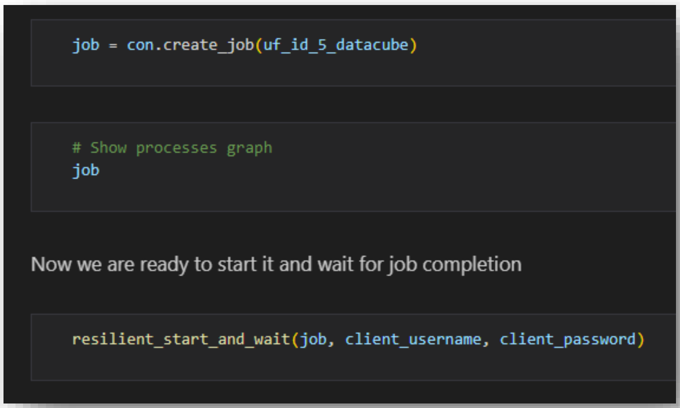


```
uf_id_5_datacube = con.datacube_from_process("uf_id_5",
    spatial_extent = input_geometry,
    temporal_extent = [
      start_date,
      end_date
    ],
    flood_extent_layer = "s3://centaur/.../flood.tif"
)
```

```
uf_id_5_datacube = uf_id_5_datacube.process(
    "save_result", data=uf_id_5_datacube, format="tif", delivery_directory="local"
)
```

Figure 6-8: openEO - Processor

- Submit the task to the local node



```
job = con.create_job(uf_id_5_datacube)
```

```
# Show processes graph
job
```

Now we are ready to start it and wait for job completion

```
resilient_start_and_wait(job, client_username, client_password)
```

6.3.2 Temporal repository

The local nodes generate the products, as it is an asynchronous process, the deliverable of it is based on the temporal repository, S3 bucket. This will allocate the products pending to be inserted into the platform, once they are ingested the products will be removed, in compliance with requirement DP-01, see D1.1 - Report on Urban Flood and Water & Food security indicators v1.0 ([RD03]). In parallel with the deliverable, local nodes will notify the central node of the action, event message will be sent, see 6.1.4.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

The components will deliver to the S3 bucket the generated products, as a prefix each product will include the next label.

- Regarding Urban flood domain:

  /*XXX*/UF/UF_ID_*ZZZZ*

- Regarding Water & Food security domain:
  /*XXX*/WFS/WFS_ID_*ZZZZ*

  Where *XXX* is the region where the product is generated, following ISO 3166 3-letter-country code[26]

Table 4: Region acronym

Region	XXX
France	FRA
Germany	DEU
Italy	ITA
Mali	MLI
Mozambique	MOZ
Somalia	SOM
Sudan	SDN

ZZZZ is a number following D2.1 – Catalogue of CENTAUR data and related specifications ([RD04]) deliverable.

The conventional names to use to deliver a product in the S3 bucket must follow the next rule:

MUST		MUST		MUST		OPTIONAL		OPTIONAL		OPTIONAL		OPTIONAL
Region	_	**Domain**	_	**Indicator**	_	**SubCategory**	_	**StartDate**	_	**EndDate**	_	**Suffix**
FRA		UF		0000		*max 8 chars		YYYYmmdd[27]		YYYYmmdd[28]		*max 8 chars
DEU		WFS										
ITA		AUX										
MLI												
MOZ												
SOM												
SDN												

Figure 6-9: Conventional file names

[26] https://www.iso.org/iso-3166-country-codes.html
[27] https://www.iso.org/iso-8601-date-and-time-format.html
[28] https://www.iso.org/iso-8601-date-and-time-format.html

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Table 5: Examples of product names

Allowed combinations, examples	
Complete	FRA_UF_0001_EVENT_20200101_20200101_yearly.json
Optional until EndDate	FRA_UF_0001_EVENT_20200101_20200101.json
Optional until SartDate	FRA_UF_0001_EVENT_20200101.json
Optional until SubCategory	FRA_UF_0001_EVENT.json

Every product or dataset submitted from local nodes to the central node has associated STAC metadata available. This metadata should conform to the STAC specification and accurately describe the dataset, including essential information such as title, description, and spatial and temporal extents.

6.4 Infrastructure Layer

To facilitate the integration during the different development cycles, the CENTAUR platform will be based on a Kubernetes infrastructural layer. The layer will provide the following base services in order to provide a suitable environment for the Central Node:

- S3 like interface as a storage with implementation of ACL rules to guarantee separation of concerns
- Kubernetes cluster where the Central Node first version can be deployed using Helm Chart
- Security services and other infrastructural services

The on-boarding of the local nodes described above will be defined in the second version of the project.

7 USER STORIES

This section includes the list of the CENTAUR platform user stories based on the requirements defined in D1.1 ([RD03]). As the current document will have two major releases, the list will be enriched progressively.

The high-level functions of the CENTAUR platform identified so far are the following: data discovery, situational awareness, early warning notifications, user management, area-of-interest management, and request on-demand data. Each of the high-level functions is decomposed into user stories in the sections below.

7.1 Discover Products

The following user stories are part of the Discover Products functional block.

Table 6: Discover data user stories

Title:	*Discover products*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-001	End-user	Display relevant information from the catalogue data, including agreements or licenses	Whether or not it is possible to re-use the data; to obtain further details about the data	Viewer Service catalogue	GR-07 DP-05 PR-05

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Title:	*Discover products*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-002	End-user	Search for urban flood datasets	To study the urban flood situation	Viewer Service catalogue	OR-03 DP-05
US-003	End-user	Access to datasets that enrich the climate security perspective. Somalia, Mozambique, Mali	To study the water and food insecurity situation	Viewer Service catalogue	OR-03 DP-05 OR-09 DP-05
US-004	End-user	Access to different types of data formats including EO data, Non-EO data, or from Copernicus services	To support analysis for decision-making, information can be represented in a variety of formats, from map visualizations to documents	Viewer Service catalogue	DP-03 DP-05
US-005	End-user	Explore the dataset/indicators/indexes available in the system	To display data/indicators/indexes available in the system to improve situational awareness	Viewer Service catalogue	DP-05 PR-03
US-006	End-user	Filter the datasets in the catalogue with very specific searches	To better target the search for information on a particular domain	Viewer Service catalogue	DP-05 PR-13

7.2 SITUATIONAL AWARENESS

The following user stories are part of the Situational Awareness functional block, which includes how the system displays situational awareness with the information generated by the system and how it can be configured.

Table 7: Situational awareness user stories

Title:	Situational Awareness				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-101	End-user	Display an accurate mapping of urban flooding with colors that make it easy to understand the view	React promptly to a potential risk to the population.	Viewer, Received data manager, Urban flood processor	OR-02

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Title:	Situational Awareness				
Identifier	**Who**	**What**	**Why**	**Components:**	**Applicable Requirements:**
US-102	End-user	Display future trends in climate security using time-series charts	While studying indicators, the need for predictive analytics to improve situational awareness	Viewer, Urban flood processor, Water & food security processor	OR-04
US-103	End-user	Display Climate Security Risk indicators and indexes with colors that make it easy to understand the view	To support the decision-making process	Viewer, Received data manager, Urban flood processor, Water & Food security processor	OR-05 PR-03
US-104	End-user	Display Urban Flood Indicators on a map	To assess urban flood impact on population providing early information to support the decision-making process. To improve CEMS pre-tasking success	Viewer, Received data manager, Urban flood processor	OR-07 PR-03
US-105	End-user	Display real-time data flood for a predefined set of areas of interest on a map: - Mozambique (Cabo Delgado, Maputo and Manica) - Ebro Basin (Navarra) - German Floods (Danube River) - Piemonte, Italy	Assess flood events and climatic aspects	Viewer, Received data manager, Urban flood processor	OR-08
US-106	End-user	Display foreseen risk assessment for water and food insecurity in a predefined set of areas of interest on a map: - Somalia	Assess the water and food insecurity with predictions related to the area of interest	Viewer, Received data manager, Water &	OR-09

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Title:	Situational Awareness				
Identifier	**Who**	**What**	**Why**	**Components:**	**Applicable Requirements:**
		- Mozambique - Mali	under observation	Food security processor	
US-107	End-user	Display early-warning alerts for water and food insecurity in a predefined set of areas of interest on a map: - Somalia - Mozambique - Mali	Improve situational awareness related to the area of interest under observation	Viewer, Received data manager, Water & food security processor	OR-09
US-108	End-user	Display the data available in the system through different types of visualizations: -a map to present the information in a customized view (regarding the georeferenced data available) -charts, plots	To support the decision-making process in the urban flood and water & food security domains	Viewer	PR-03
US-109	End-user	Display a landing page with the capabilities of the system including the services provided by the system	For an overview of all services provided on the platform	Viewer	PR-08
US-110	End-user	Generate a view of the alerts generated by the system by setting thresholds.	When studying the information displayed on a map, an alert may appear as a result of a combination of several inputs. The alert warns to improve situational awareness and preparedness	Viewer Received data manager Urban flood processor Water & Food security processor	OR-01
US-111	End-user	Customize an alert by managing thresholds for different metrics, indicators, or indexes	To refine the data displayed in the area of interest, it can define the thresholds of	Viewer Received data manager Urban flood processor	PR-06

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Title:	Situational Awareness				
Identifier	Who	What	Why	Components:	Applicable Requirements:
			the metrics, indicators, or indexes	Water & Food security processor	

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

7.3 Early-warning notification

The following user stories are part of the Early-Warning notification functional block, which includes the subscription and un-subscription to events.

Table 8: Early-warning notification user stories

Title:	*Early-warning notification*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-201	End-user	Subscribe to receive an email notification when a relevant event occurs (for example to the generation of an urban flood index)	To get an early assessment of risks of concern to the end-user	Received data manager	GR-05 IR-01
US-202	End-user	Unsubscribe from a notification	To stop following a system notification	Received data manager	GR-05 IR-01

7.4 User management

The following user stories are part of the User Management functional block, which includes the creation and end-user profile configuration.

Table 9: User management user stories

Title:	*User management*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-301	End-user	Login to the system	To explore the capabilities that the system provides	Access manager	AR-01
US-302	Administrator	Create users	Create end-users to be able to use the system and allow them to access a specific type of products	Access manager	AR-03
US-303	End-user	Update end-user profile	Allow the user to update their personal data stored on the platform	Viewer Access manager	PR-16

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

7.5 AREA OF INTEREST MANAGEMENT

The following user stories are part of the Area of Interest Management functional block, which includes the creation/deletion of the area of interest needed to request information from the system.

Table 10: Area of interest management

Title:	*Area Of Interest management*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-401	End-user	List the area of interest available in the system	To select an area of interest to perform a request to the system	Viewer	PR-03
US-402	Admin	Create/delete areas of interest	To assist in requesting retrieval of information from a particular area of interest, set out the most common areas of interest	Viewer	PR-15

7.6 REQUEST ON-DEMAND PRODUCTS

The following user stories are part of the Request On-demand Products functional block, which includes the generation of a request to a service that the system provides.

Table 11: Request on-demand data

Title:	*Request on-demand products*				
Identifier	Who	What	Why	Components:	Applicable Requirements:
US-501	End-user	Request products on-demand	Support decision-making by obtaining on-demand data.	Viewer, Data requester, Received data manager, Catalogue	DP-06

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

8 TRACEABILITY MATRIX

This section contains a direct traceability matrix from the user requirements as defined in D1.1 ([RD03]) to the Architectural Design and development document. The traceability matrix between the user stories and the components is detailed in Section 6.2.5.

8.1 GENERAL

Code	Requirement Name	Priority	Functional/Non-Functional	Component
GR-01	Security	MUST HAVE	Non-Functional	Section 3.2.7
GR-02	Reliability	MUST HAVE	Non-Functional	Section 3.2.3 Section 3.2.6
GR-03	Ethics	MUST HAVE	Non-Functional	Section 3.2.8
GR-05	Notifications	MUST HAVE	Functional	Received data manager
GR-07	Licensing	MUST HAVE	Functional	Viewer

8.2 ACCESSIBILITY

Code	Requirement Name	Priority	Functional/Non-functional	Component
AR-01	Access regulation	MUST HAVE	Functional	Access manager
AR-03	Data Access Restrictions	MUST HAVE	Functional	Access manager

8.3 OPERATIONAL

Code	Requirement Name	Priority	Functional/Non-functional	Component
OR-01	Risk Assessment and Early-warning System	MUST HAVE	Functional	Viewer, Received data manager, Urban flood processor, Water & Food security processor
OR-02	Urban Flood Mapping	MUST HAVE	Functional	Viewer, Received data manager, Urban flood processor
OR-03	Catalogue of Datasets	MUST HAVE	Functional	Viewer, Service catalogue
OR-04	Simulation and Predictive Analysis	MUST HAVE	Functional	Viewer, Urban flood processor, Water & food security processor
OR-05	Improved Situational Awareness	MUST HAVE	Functional	Viewer, Received data manager, Urban flood

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Code	Requirement Name	Priority	Functional/Non-functional	Component
				processor, Water & food security processor
OR-07	Improved CEMS pre-tasking	SHOULD HAVE	Functional	Viewer, Received data manager, Urban flood processor
OR-08	CEMS Products and Services	MUST HAVE	Functional	Viewer, Received data manager, Urban flood processor
OR-09	CSS-SEA Products and Services	MUST HAVE	Functional	Viewer, Received data manager, Water & food security processor
OR-10	Maturity level	MUST HAVE	Non-functional	N/A

8.4 Data Indicator

Code	Requirement Name	Priority	Functional/Non-functional	Component
DP-01	Big Data	MUST HAVE	Non-functional	Section 6.1.5 Section 6.3.2
DP-02	Satellite Imagery Access	MUST HAVE	Non-functional	Section 6.2.1
DP-03	Multisource data	MUST HAVE	Functional	Viewer, Service catalogue
DP-05	Catalogue	MUST HAVE	Functional	Viewer, Service catalogue
DP-06	Asynchronous jobs	MUST HAVE	Functional	Viewer, Data requester, Received data manager, Service catalogue
DP-07	Tools and AI algorithms	MUST HAVE	Functional	Section 6.2.1
DP-08	Model Development and Simulation	MUST HAVE	Functional	Section 6.2.1
DP-09	Orchestration of processing chains	MUST HAVE	Functional	Section 6.2.1

8.5 Platform

Code	Requirement Name	Priority	Functional/Non-functional	Component
PR-01	Graphic User Interfaces	MUST HAVE	Non-functional	Section 6.1.2
PR-02	Cloud architecture	MUST HAVE	Non-functional	Section 3.2
PR-03	Geoviewer component	MUST HAVE	Functional	Viewer
PR-04	Graphs and statistical visualization	MUST HAVE	Functional	Viewer

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Code	Requirement Name	Priority	Functional/Non-functional	Component
PR-05	Datasets visualization	MUST HAVE	Functional	Viewer
PR-06	Alerts and thresholds configuration and management	MUST HAVE	Functional	Viewer
PR-08	Single access point	MUST HAVE	Functional	Viewer
PR-11	Language	MUST HAVE	Non-Functional	Section 6.1.2
PR-15	Configurable AOIs	MUST HAVE	Functional	Viewer
PR-16	Personal Area Component	MUST HAVE	Functional	Viewer, Access manager

8.6 INTEROPERABILITY

Code	Requirement Name	Priority	Functional/Non-functional	Component
IR-01	Data delivery: online platform or web service.	MUST HAVE	Functional	Received data manager

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

9 CONCLUSIONS

The platform design is based on the user requirements defined in D1.1 ([RD03]). They were crucial in establishing the design of the platform. A traceability matrix that traces user requirements to design is used to check compliance. The system is explained from the end-user perspective through user stories, an informal way of describing high-level functions. The purpose is to describe how a user can interact with the system.

As mentioned in the introduction, the platform design is presented, addressing the following topics:

- the general principles of the architecture, the scalability, and the agnostic design to the IT resource tier.
- the high-level concept, with the description of different nodes allocated in the distributed architecture based on microservices.
- standard interfaces to connect all components.
- type of users to access the platform, taken from the user requirements document D1.1 ([RD03]).
- a list of the technologies and pre-existing components, relying on open-source software.
- the high-level functions of the CENTAUR platform identified to cover the functional user requirements.

This document will be the input for the integration of components in the platform, which will be done progressively, that's why the use of microservices and a scalable platform will make easier this task.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101082720 - CENTAUR