Design of the Multi-Hazard Early Warning System

Álvaro Rodríguez[1], Jordi Roca [1], Paolo Campanella[2], Rafael Cuestas[1], Xavier Llort[1], Nicola Rebora[2], Rafael Sánchez-Diezma[1]
[1]Hydrometeorological Innovate Solutions S.L. (Barcelona, Spain)
[2]Centro Internazionale in Monitoraggio Ambientale (Savona, Italy)

The main outcome of the ANYWHERE WP3 is the Multi-Hazard Early Warning System (MH-EWS), a Software as a Service (SaaS) cloud-based platform that essentially integrates the forecast and impact models developed within WP2 and the existing Pan-European platforms, computes the impact products and serves the results and other generated information to the users. The MH-EWS is internally divided into several interconnected modules, where each of them has a specific role.

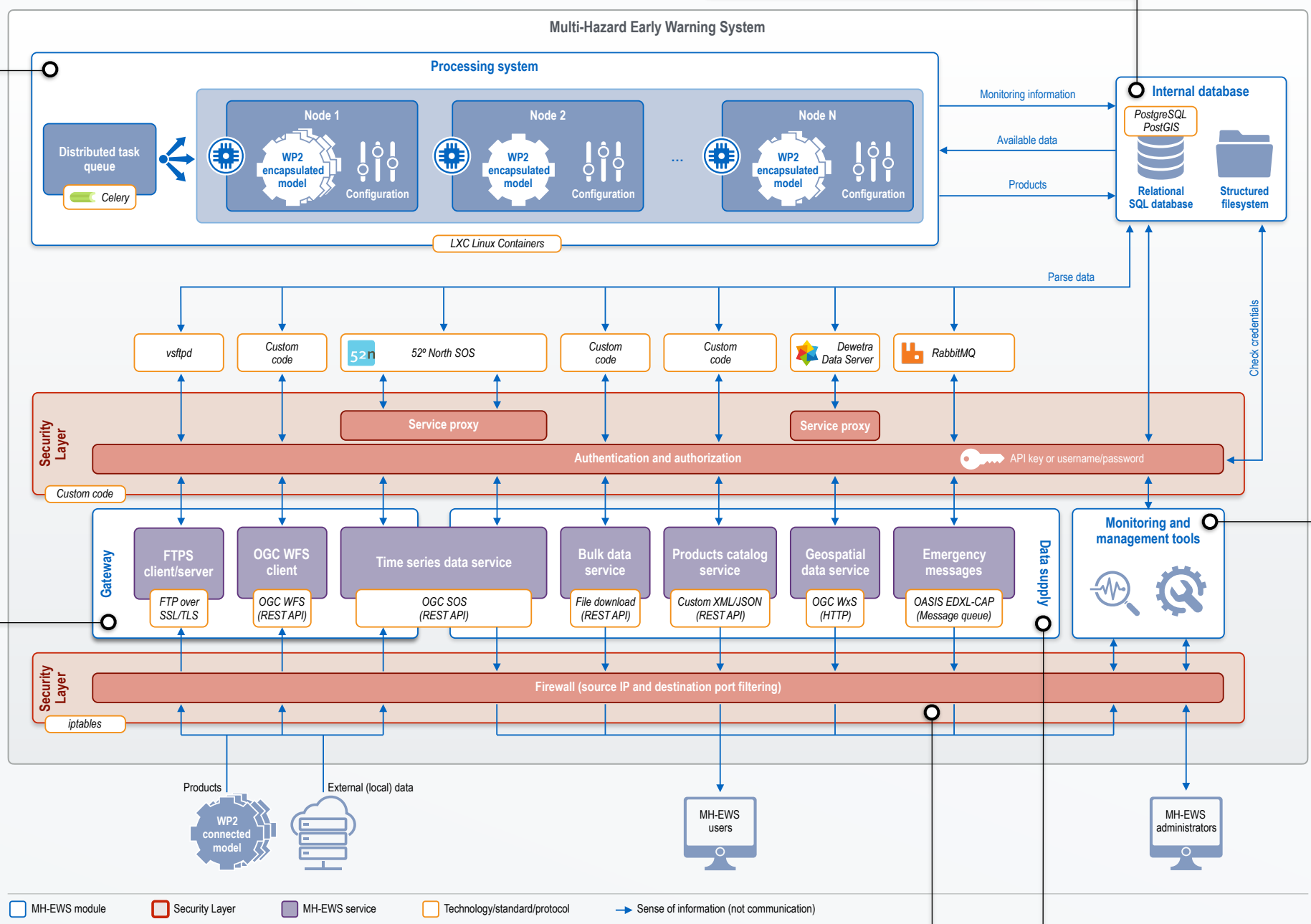
The **Processing system** is in charge of executing the **encapsulated models**. Physical servers' resources (CPU, RAM, disk, etc.) are distributed among several `LXC containers`. Each container implements a *node* inside of which the models are executed using specific configurations. *Nodes* are orchestrated by a special container running a `Celery` distributed task queue that assigns processes to nodes depending on their requirements. Then, there are different types of *nodes*:

- Big nodes with a high amount of computing resources assigned to be able to execute the most demanding models.
- Small nodes with less computing resources assigned to execute lightweight models.
- License nodes that include commercial software (like IDL) required to execute certain models.

The **Internal database** stores all the information used by the other MH-EWS modules. Therefore, it stores external data integrated through the **Gateway**, output products generated by the impact models in the **Processing system**, **monitoring and management** information and users-related information (accounts, permissions, etc.). The module is composed by a `PostgreSQL` with `PostGIS` relational database and a structured filesystem to store regular files.

The **Monitoring and management tools** are a set of web-based utilities focused on the management of the MH-EWS users and impact products, and the monitoring of the different processes taking place in the **Processing system**. These tools are intended for two different roles:

- The users can view/update their personal information, view the current and historical contracted products and check their availability (e.g. which is the most recent data, status of the data sources, etc.). Users also receive automatic reports about the status of the system and products, and a notification when a product subscription is about to expire.
- In addition to what users can do, administrators can manage (add/remove/modify) users and products, assign/remove products to a user, and access to detailed monitoring information regarding products' generation and also system-level processes in the servers.



The **Gateway** module is in charge of acquiring the external information, converting it to the appropriate formats and inserting it into the **Internal database**.

The **Data supply** module serves the information available in the MH-EWS **Internal database**.

The access to any resource within the MH-EWS is controlled and protected by the **Security layer** at different levels:

- Network security, where a firewall (implemented with `iptables`) filters all incoming connections looking at the source IP address and keeping closed all unused ports to protect those modules directly exposed to the Internet (**Gateway**, **Data supply** and **Monitoring and management**).
- Authentication and authorization of the users (e.g. clients, administrators and data providers) through API keys for the REST API and WxS requests, and with classic username/password authentication for the **SFTP** and **emergency messages**' queue services. Essentially, the API key is a token (string) that uniquely identifies the user and that must be provided in each query to the MH-EWS. Using this token, the system will check the user's identity and available features (e.g. account validity, products/areas/requests availability, etc. according to the business model) and consequently allow or deny the access to the services. For those services implemented using existing solutions (like the **Time series data** service that uses `52º North SOS` and the **Geospatial data** service that uses `Dewetra Data Server`), the **Security Layer** implements a *Service proxy* that removes the API key -once validated- and transfers the request to the underlying service.

Acknowledgments:
This work is being carried out in the framework of the EU H2020-Project ANYWHERE (grant: 700099). This document reflects only the authors' views and the European Union is not liable for any use that may be made of the information contained herein.