ISME-HYDRO DESP Service Proxy Specification

for

UrbanSquare Destination Earth Use Case

_____

Version 1.0

Prepared by Mariana Damova, Emil Stoyanov

August 2024

Contents

List of Figures

List of Tables

N/A

Revision History

Name	Date	Reason For Changes	Version
Mariana Damova	20.04.2024	Accepted template	1.0
Emil Stoyanov	20.06.2024	Objectives, Architecture	1.0
Mariana Damova	21.07.2024	Introduction, Use cases	1.0
Emil Stoyanov	25.07.2024	Communication Flow	1.0
Mariana Damova	30.07.2024	Conclusion, Overall review of document	1.0

Change Log

Date	Location in the document	Change	Version

1. Introduction

2. Purpose of ISME-HYDRO Service Proxy

The primary purpose of the ISME-HYDRO DESP Service Proxy implementation is to facilitate efficient and secure communication between user-facing components and data retrieval services involved in the DESP deployment environment of the application ISME-HYDRO. This service proxy aims to streamline data access and integration processes, ensuring that users can seamlessly interact with various data sources and services. The specific objectives include:

- Enhancing Data Accessibility
    Ensure that users can easily access and retrieve data from different sources in a consistent and reliable manner.
- Improving System Integration
    Enable seamless interaction and communication between different system components, including the GUI, mobile applications, and backend services.
- Ensuring Security
    Implement robust security measures to protect data during transmission and storage, including authentication and authorization mechanisms.
- Providing a Scalable Architecture
    Design the service proxy to be scalable, allowing it to handle increasing amounts of data and user requests without compromising performance.
- Facilitating Real-time Data Processing
    Support real-time data processing capabilities to provide users with up-to-date information and insights.

Figure 1 below illustrates the dependencies of the components involved in DESP use case scenario. The DESP hosts a number of registered services that facilitate the scenario implemented also via a GUI user-facing layer. The DESP hosts containerized instances of the necessary services. In the case of ISME-HYDRO the containerized services are the GUI Layer and the Service Proxy which connects the GUI Layer with the Intelligent Layer in the network domain of Mozaika.
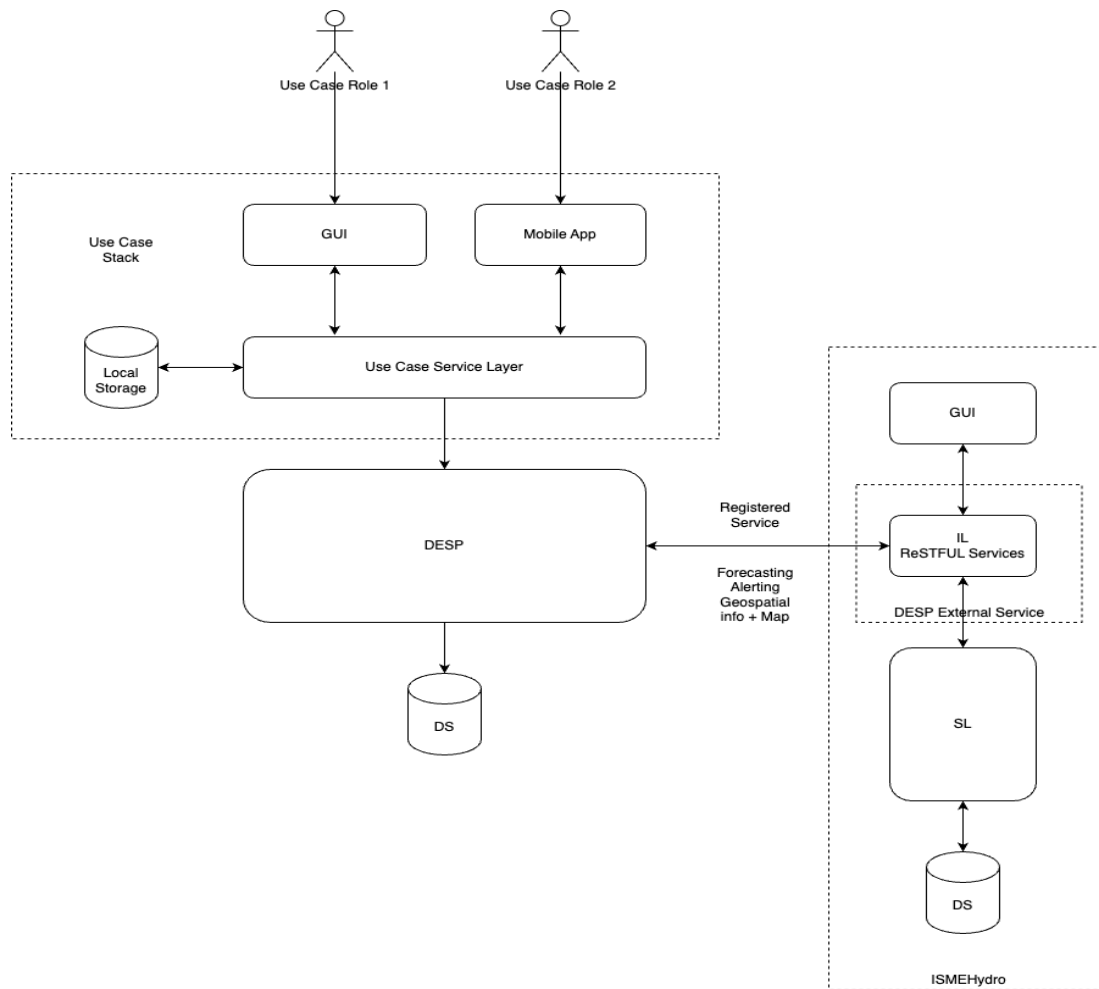
Figure 1 *DESP as an infrastructure provider for services used in different use case scenarios.*

2. Use cases

ISME-HYDRO Service Proxy has to cover the following two use cases:

1) Data access layer of ISME-HYDRO pulls data from DESP via Service Proxy and pushes them to the Forecast modeler and to the Data normalizer to be inserted in the Semantic Layer of ISME-HYDRO
2) GUI of DESP asks for data and retrieves them via Service Proxy from ISME-HYDRO

3. Architecture of ISME-HYDRO Service Proxy

The Service Proxy is a component that has one operational feature - to forward authenticated requests from the GUI Layer from the remote domain of DESP to the Intelligent Layer in the local domain of Mozaika.

This is achieved by providing the same interface with the same fingerprint as the Intelligent layer, but in a different domain. For this reason, an authentication mechanism has to be present to avoid undesired access or excess usage of network resources.

The Service proxy can be configured to accept requests only from desired clients, including the GUI layer, or other DESP services, in case the ISME-HYDRO data is interesting for other DESTINE collaborators.

Figure 2 illustrates how the data flows from the GUI Layer, through the Service Proxy to the Intelligent Layer. It is visible how the data crosses domains of operations, which means certain access control has to be implemented to avoid undesired data access.

2.1. Authentication

2.1.1. Service Proxy Level Authentication

The authentication is achieved by adding the Authentication header which contains a JWT token. This token must match a token in the configuration of the service proxy before the request is to be dispatched to the configured remote endpoint.

2.1.2. Remote Service Level Authentication (IL)

When the request is dispatched to the remote service, the service proxy must provide its own JWT token that has been issued on the side of IL. This gives the possibility for multiple service proxy instances each of which is able to serve their own set of clients or UI implementations.

In case of token mismatch or absence of a token, the service proxy must return an error code 403 FORBIDDEN.

The remote endpoint is also subject to configuration and should not be hardcoded.

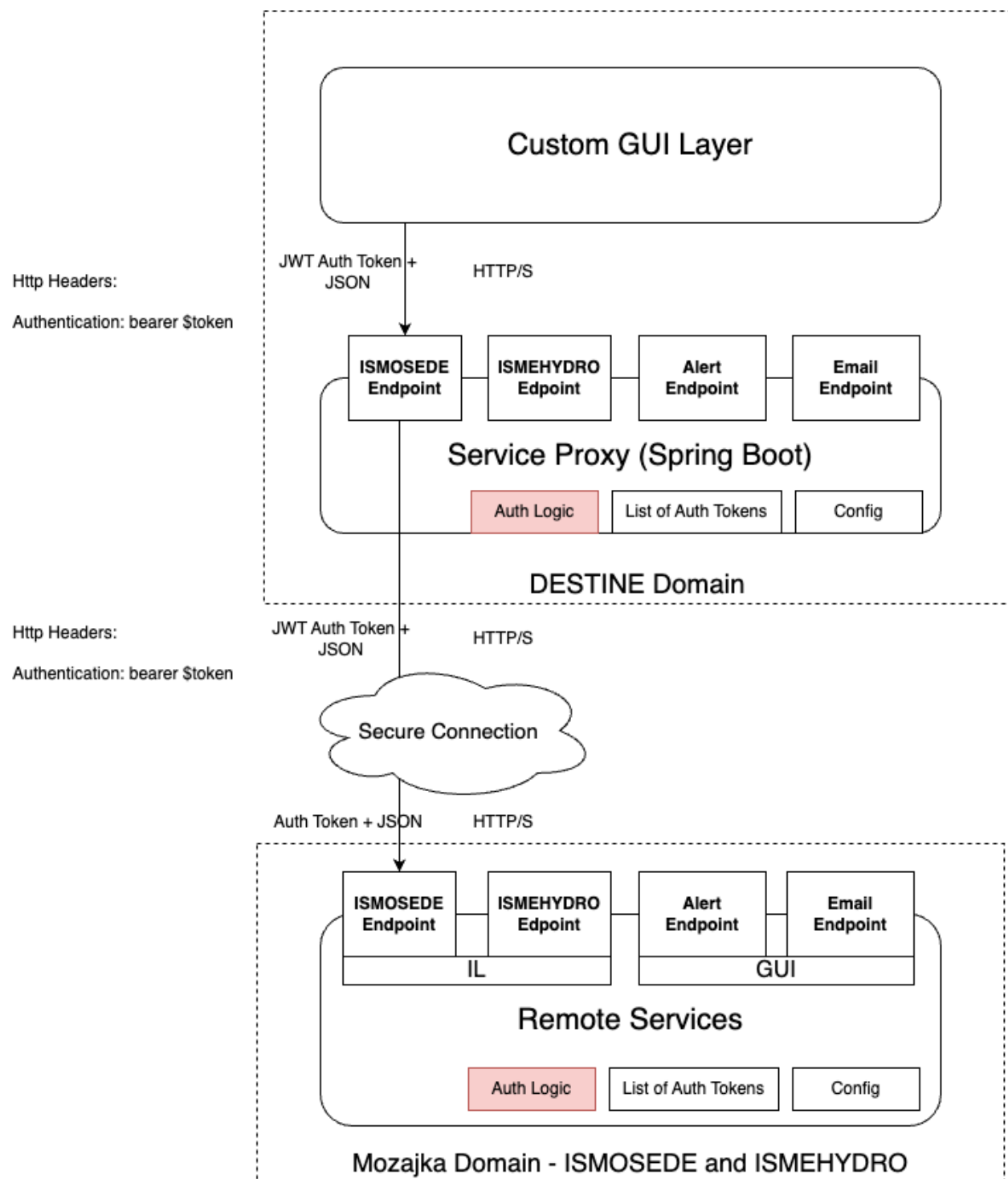Figure 2 below shows the architecture of the described solution.

Figure 2 *Architecture of the Solution.*

3. Communication Flow

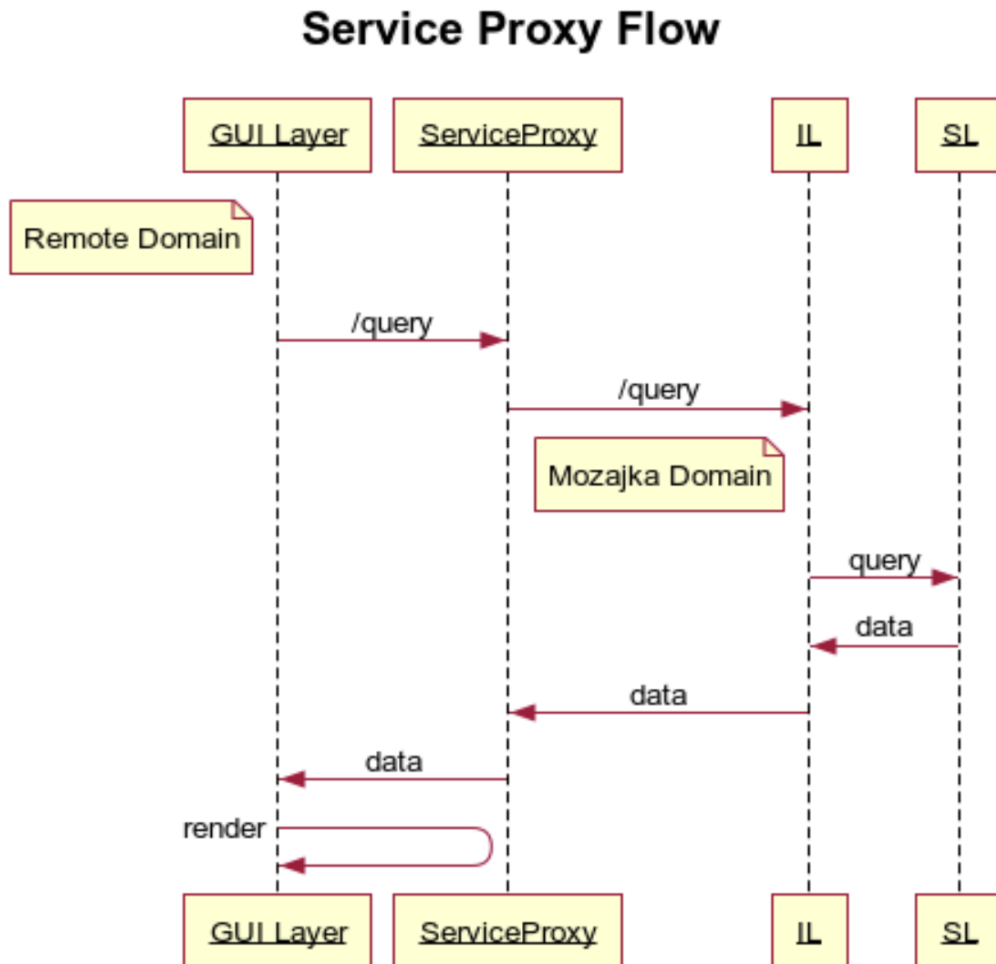Figure 3 below shows the process flow of ISME-HYDRO Service Proxy.

Figure 3 *Service Proxy Flow.*

The communication flow outlines the interaction between different components within the ISME-HYDRO DESP Service Proxy. The flow is designed to ensure efficient data exchange and processing.

1. **User Interaction** - Users interact with the system through the GUI or mobile application.
2. **Service Proxy Requests** - The GUI or mobile application sends a request to the Service Proxy.
3. **Data Query** - The Service Proxy forwards the request to the appropriate backend service or data source.
4. **Data Processing** - The backend service or data source processes the query and returns the requested data.
5. **Data Rendering** - The Service Proxy receives the data and renders it for the user interface.

6. **Display to User** - The processed data is displayed to the user through the GUI or mobile application.

This communication flow ensures that users can seamlessly retrieve and view data from various sources in real-time.

The same communication flow applies for all of the available endpoints in the service proxy.

4. Conclusion

This document presents the functional features and the architecture of ISME-HYDRO Service Proxy, destined to connect the application ISME-HYDRO to the DESP platform and to demonstate federated integration and use of DestinE platform as part of UrbanSquare use case covering Flood and Resources demonstrators.