System Requirements Document (SRD)
Project Name: Wildfire Visualization Platform
Prepared by: CRIM

1. Introduction

This document outlines the system requirements for developing a Wildfire Visualization Platform in response to the 2023 CSSP challenge. The platform aims to provide an open architecture system for reviewing historical wildfire events and generating synthetic wildfire simulations. This platform will utilize open-source technologies, OGC standards, and various external data sources to dsupport wildfire management and firefighting operations.

2. System Overview

The platform will be a web-based desktop application designed for local use. It will offer two modes:

1. **Past Wildfire Event Analysis**: Integrating and visualizing historical wildfire data.
2. **Synthetic Wildfire Simulation**: Enabling users to create wildfire scenarios with user-defined parameters.

The platform will feature 2D map visualizations, time-lapse animations, and a cache system for offline data reuse. The back-end will be containerized using Docker and designed with microservice architecture for flexibility and scalability.

3. System Requirements

3.1. Functional Requirements

3.1.1. Data Sources and Integration

All datasets, including **STAC items** or **WFS datasets**, will be provided to the project team. The platform will not search for datasets autonomously.
The platform will be able to **list all the provided datasets**, including metadata such as:

- Dataset name and description.
- Data format (e.g., GeoTIFF, SHP, GeoJSON).

- The processes required to process the data (e.g., fire boundary extraction, satellite imagery processing).
- The source of the dataset (STAC or WFS).

3.1.1.1. User Selection

When a user selects a dataset, the platform will:

- Download the selected dataset (whether from a **STAC** catalog or a **WFS** service).
- Download associated **satellite imagery** for the selected geographic zone.
- Retrieve **historical meteorological data** for the region.
- Access relevant **topological data** to supplement wildfire analysis.

3.1.1.2. Data Rendering

The platform will process all downloaded data and render it using **GeoServer**. This includes:

- Displaying fire perimeters, meteorological overlays, satellite imagery, and topographical maps as layers in the user interface.
- Ensuring seamless integration of the datasets, providing users with a cohesive view of all geospatial data for the selected area.

3.1.2. Visualization

The platform will feature:

- **2D map visualizations** with multiple layers (e.g., satellite imagery, meteorological data, topographical information).
- **Time-lapse animations** to visualize the progression of wildfires over time.
- Compatibility with **GeoServer** for map generation.
- The platform will incorporate the **PMS 936 Symbology Polygon**, which is part of the **NWCG (National Wildfire Coordinating Group) PMS 936 Symbology** standard. This standard defines how wildfires and related elements (e.g., fire perimeters, firelines) are represented on maps using polygons. The system will display wildfire perimeters and other wildfire-related features using these standardized polygons, ensuring consistency with wildfire management practices.

3.1.3. Synthetic Wildfire Simulation (To be refined)

Users will be able to:

- Select a geographic area to simulate wildfires.
- Define parameters such as wind direction, humidity, temperature, and vegetation.
- Run simulations and visualize results on the platform.
- The system will allow integration of third-party libraries or free simulation software for wildfire behavior modeling.

3.1.4. Cache and Offline Mode

- The platform will locally cache downloaded data to avoid repeated requests.
- When reopened, the system should load cached data without requiring re-download.

3.1.5. No User Management

- The platform will not require user roles, authentication, or access management.
- All functions will be available to all users, with no need for logging.

3.2. Non-Functional Requirements

3.2.1. Architecture

- The platform will be developed with a **microservice architecture**, using Docker to containerize subsystems.
- The architecture will ensure that computationally intensive processes (e.g., wildfire simulations) can be offloaded to external infrastructure.

3.2.2. Performance and Scalability

- The platform will handle medium to large-scale geospatial datasets.
- It must be scalable to accommodate future integration of real-time data (although not a current requirement).
- The system must be capable of running on local desktop environments without cloud dependencies, though it may be adapted for cloud use if needed.

3.2.3. Security

As no personal or sensitive data will be processed, the system has no specific security or data privacy requirements.

3.2.4. Standards Compliance

The system will comply with **OGC standards** for geospatial data processing and visualization.

- **STAC** will be used to catalog and retrieve raw geospatial data.
- **OGC WFS** will provide ready-to-use, processed geospatial data.
- **Weaver (OGC API for Processes)** will be used to orchestrate and automate data processing workflows, transforming raw data into usable formats for visualization or analysis.

3.2.5. Visualization Technology

- The system must support web technologies (HTML5, CSS3, JavaScript), and be accessible via modern browsers.

- Visualization frameworks such as **Leaflet**, **OpenLayers**, or **D3.js** may be used depending on project needs.

3.3. User Interface (UI) Requirements

3.3.1. Initial Screen on Application Load

When the application is loaded for the first time, the main user interface will consist of:

- A **map display area** covering the center of the screen, initially showing a base layer (e.g., global or national map) with no active data layers.
- A **sidebar or toolbar** providing options for users to choose which data layers to display (e.g., satellite imagery, fire perimeters, weather conditions).
- A **search bar or map navigation tools** (zoom, pan) to allow users to move to specific regions of interest.

3.3.2. Resetting the Application

- The user will be able to **reset the application** to its initial state by:
- Clicking a **reset button** located in the toolbar or settings menu.
- The reset function will clear all currently active data layers, remove any simulation parameters, and return the map to its default view (e.g., zoomed-out base layer).
- The reset function will also clear any cached local data for a fresh session.

3.3.3. Data and Dataset Management

The user will be able to select and manage which datasets to use by:

- A **data source selection panel** in the sidebar or a dedicated data management menu.
- This panel will provide options to:
    - Choose specific datasets to load (e.g., historical wildfire data, satellite imagery, meteorological data).
    - Indicate the time period and geographical area for which data is required.
    - Download data from external sources (e.g., Planetary Computer, Open Meteo) via integrated API calls.
    - A **progress indicator** will show the download status for large datasets.
    - After the dataset is loaded, users will be able to toggle its visibility on the map and interact with its layers.

3.3.4. Simulation Controls (To be refined)

In the simulation mode, the UI will include:

- A **parameter input panel** where users can define simulation variables such as wind direction, temperature, and humidity.
- A **simulation start/stop button** to run or halt the simulation.

- A **timeline control** for time-lapse animations of the wildfire simulation, allowing users to fast-forward, pause, or replay the simulation.

3.4. System Constraints

- The platform must operate on desktop environments (Windows, macOS, Linux).
- It should not depend on real-time data feeds but should allow for future adaptability.
- The system should not have any external dependencies requiring constant internet access, aside from the initial data retrieval.