PANTHEON

Community-Based Smart City Digital Twin Platform for Optimised DRM operations and Enhanced Community Disaster Resilience

D7.3

SECURE DATA REPOSITORY

The project has received funding from the European Union's Horizon Europe programme under Grant Agreement N°101074008.

DOCUMENT INFO

Deliverable Number	D7.3
Work Package Number and Title	7 Integration and testing of PANTHEON Platform
Lead Beneficiary	SIMAVI
Due date of deliverable	30/04/2025
Deliverable type[1]	DATA
Dissemination level[2]	PU
Author(s)	Andrei Alexandrescu(SIMAVI), Cristina Barrado (UPC), Iacob Crucianu (SIMAVI)
Internal reviewer(s)	Fanis Fakoukakis (FINT), Marc Bonazountas (EPSILON), Mike Karamousadakis (THL), Anna Tsabanakis (THL)
Version - Status	1.4

[1] **Please indicate the type of the deliverable using one of the following codes**:

R = Document, report

DEM = Demonstrator, pilot, prototype, plan designs

DEC = Websites, patents filing, press & media actions, videos

DATA = data sets, microdata

DMP = Data Management Plan

ETHICS: Deliverables related to ethics issues.

OTHER: Software, technical diagram, algorithms, models, etc.

[2] **Please indicate the dissemination level using one of the following codes**:

PU = Public

SEN = Sensitive

TASK ABSTRACT

The deliverable incorporates the result of Task 7.4 . This task is responsible for the management and storage of the data utilised by, and generated from components within the project. The model will be defined based upon the type, frequency, and content of each data asset utilised within the project. Based upon this data, model security plans will be defined to control data access within the PANTHEON project based upon user privileges. Encryption will be used to secure data access where required.

REVIEW HISTORY

Version	**Date**	**Modifications**	**Editor(s)**
1.0	29/04/2025	Tabel of Contents released	Andrei Alexandrescu (SIMAVI)
1.1	09/05/2025	Restructuring the report and adding content	Iacob Crucianu & Andrei Alexandrescu (SIMAVI), Cristina Barrado (UPC)
1.2	30/05/2025	Editing sections. Adding details about data	Iacob Crucianu & Andrei Alexandrescu (SIMAVI), Cristina Barrado (UPC)
1.3	06/06/2025	Version ready for internal review	Otilia Bularca (SIMAVI)
1.4	30/06/2025	Version ready for submission	Otilia Bularca (SIMAVI)

DISCLAIMER

Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

This deliverable (D7.3) details the design, implementation, and operationalization of the secure data repository developed for the PANTHEON Community-Based Smart City Digital Twin Platform. Created under Task 7.4 of Work Package 7, the repository is a central pillar of the PANTHEON ecosystem, supporting robust data-driven disaster risk management (DRM) and enhancing community resilience in urban environments exposed to multi-hazard risks.

The repository manages a diverse and high-volume collection of over 1,000 datasets across four core disaster scenarios—earthquakes, heatwaves, wildfires, and cyberattacks—focusing on two European pilot regions: Athens/Attica (Greece) and Vienna (Austria). It integrates data from authoritative sources such as UN/WorldPop, Copernicus, OpenStreetMap, ECMWF, and national emergency agencies, alongside synthetic datasets and real-time streams from UAVs, IoT sensors, and simulation outputs.

Technically, the data infrastructure is built on a federated, layered architecture incorporating PostgreSQL/PostGIS for spatial databases, MinIO for object storage, Neo4j for graph data, and Apache Kafka for real-time messaging. This hybrid setup ensures flexible and efficient data access, supporting both batch processing and real-time decision-making workflows. Standardized data models and metadata schemas promote cross-scenario interoperability and allow seamless integration with GIS tools, analytical platforms, and emergency response dashboards.

Security is a core feature of the repository, implemented through a comprehensive framework encompassing role-based access control via Keycloak, encrypted data transmission (TLS) and storage (AES-256), GDPR-aligned identity and data protection measures, and detailed audit trails. These mechanisms ensure that sensitive data—such as population vulnerability profiles and critical infrastructure maps—can be shared securely among authorized stakeholders.

The deliverable also defines a set of ingestion pipelines, data quality checks, and workflow orchestration strategies that support scenario-specific applications like evacuation planning, resource allocation, fire spread simulation, and cooling infrastructure optimization. Kafka-based event-driven dataflows and REST APIs provide flexible access to both historical and live data streams, making the repository suitable for integration with external emergency management platforms.

In summary, this secure data repository enables scalable, resilient, and interoperable data operations at the heart of the PANTHEON digital twin. It empowers planners, responders, and researchers with timely, accurate, and scenario-relevant datasets—facilitating informed decision-making across multiple hazard types and urban contexts.

1. INTRODUCTION

1.1 OVERVIEW

This deliverable constitutes the secure data repository component of the PANTHEON Community-Based Smart City Digital Twin Platform, specifically developed under Task 7.4 within Work Package 7. As a **DATA deliverable**, this document provides comprehensive documentation, specifications, and access mechanisms for the multi-hazard emergency response datasets that form the foundation of the PANTHEON platform's disaster risk management capabilities.

The PANTHEON platform addresses critical gaps in community-based disaster resilience by providing an integrated data ecosystem that supports emergency response planning, training, and real-time decision support across four major disaster scenarios: earthquakes, heatwaves, wildfires, and cyberattacks. The secure data repository serves as the central data infrastructure, managing heterogeneous datasets ranging from high-resolution population demographics and critical infrastructure mappings to real-time simulation outputs and UAV-collected field data.

Built on a federated storage architecture combining PostgreSQL/PostGIS, MinIO object storage, Neo4j graph databases, and Kafka streaming platforms, the repository ensures secure, scalable, and interoperable data access for emergency response stakeholders. The system implements role-based access control through Keycloak authentication, comprehensive encryption strategies, and ACID-compliant data management to protect sensitive emergency response information while enabling collaborative decision-making.

This data repository encompasses **two primary European urban regions**—Athens/Attica (Greece) and Vienna (Austria)—providing comprehensive coverage for diverse climatic, geographic, and urban planning contexts. The repository manages **datasets** spanning multiple scenarios, including real-time simulation outputs, authoritative demographic data from UN/WorldPop, infrastructure mappings from OpenStreetMap, environmental data from ERA5 and Copernicus, and specialized emergency response datasets developed specifically for the PANTHEON platform.

Key innovations include standardized data schemas enabling cross-scenario analysis, unified Kafka messaging architectures supporting real-time data streaming, comprehensive workflow documentation for dataset interdependencies, and robust security frameworks ensuring data confidentiality while supporting multi-agency emergency response coordination. The repository supports both batch and streaming data access patterns, enabling integration with external emergency management systems, research platforms, and operational dashboards.

As the foundational data infrastructure for the PANTHEON digital twin environment, this repository enables first responders, emergency planners, and community stakeholders to access reliable, current, and comprehensive datasets essential for effective disaster risk management, training simulation, and evidence-based emergency response planning across multiple hazard types and geographic scales.

1.2 STRUCTURE OF DELIVERABLE

This DATA deliverable is organized to provide comprehensive documentation and access specifications for the PANTHEON secure data repository, structured around data catalogue principles rather than traditional report organization. The document architecture reflects the multi-faceted nature of emergency response data management, ensuring that users can efficiently locate, understand, and access the datasets required for their specific emergency response or research applications.

Chapter 2: PANTHEON Dataset Framework establishes the foundational data architecture and organizational principles governing the entire repository. This chapter provides essential context for understanding how datasets are classified, organized, and integrated within the PANTHEON platform. Section 2.1 presents the layered system architecture, positioning the secure data repository within the broader PANTHEON ecosystem. Sections 2.2-2.4 define the data model, classification schemas, and source characterizations that ensure consistency across all emergency scenarios. Section 2.5 documents the comprehensive disaster scenario coverage, detailing geographic, temporal, and thematic boundaries for each of the four supported hazard types. Section 2.6 provides complete data source documentation and provenance tracking, enabling users to understand data authority, reliability, and licensing constraints. Section 2.7 presents detailed workflow documentation, illustrating how datasets flow through processing pipelines and interact across different emergency scenarios.

Chapter 3: Dataset Specifications by Scenario constitutes the core data catalogue, providing detailed specifications for every dataset within the repository. This chapter is organized by emergency scenario (earthquake, heatwave, wildfire, cyberattack) with comprehensive documentation for input datasets, processing requirements, output specifications, and shared cross-scenario resources. Each scenario section includes complete metadata specifications, sample data structures, file formats, coordinate systems, update frequencies, and integration requirements. Section 3.8 presents unified data schemas and metadata standards that enable seamless integration across all scenarios and processing components.

Chapter 4: Data Repository & Security Framework documents the technical infrastructure and security mechanisms that enable secure, scalable data access. This chapter provides essential information for system administrators, developers, and users requiring programmatic access to repository datasets. Sections 4.1-4.3 detail the infrastructure architecture, storage systems, and organizational structures. Sections 4.4-4.5 document data ingestion pipelines and query mechanisms, including comprehensive API specifications and sample usage patterns. Sections 4.6-4.9 present the complete security framework, including authentication protocols, encryption strategies, and system-specific security implementations.

Chapter 5: Technical Considerations addresses regulatory compliance, scalability, performance, and versioning requirements essential for operational deployment in emergency response contexts. This chapter ensures that technical stakeholders understand the system's capabilities, limitations, and compliance with relevant data protection regulations.

Chapter 6: Maintenance and Monitoring provides operational guidance for repository maintenance, including monitoring tools, logging procedures, backup strategies, and recovery protocols essential for ensuring continuous availability during emergency response operations.

The document includes comprehensive appendices with data format specifications, API examples, and detailed metadata schemas. All dataset references include direct access paths, enabling immediate data retrieval for authorized users. The structure prioritizes practical data access while maintaining complete documentation standards required for scientific reproducibility and operational reliability.

This organization ensures that the deliverable serves multiple user communities: emergency response planners requiring quick access to specific datasets, researchers needing comprehensive metadata and provenance information, system administrators managing repository infrastructure, and software developers integrating PANTHEON data into external applications. Each chapter provides both conceptual overview and practical implementation details, enabling effective utilization of the PANTHEON data repository across diverse emergency response applications.

1.3 DATASET DELIVERABLE SCOPE AND OBJECTIVES

1.3.1 PRIMARY SCOPE DEFINITION

This DATA deliverable encompasses the complete secure data repository developed for the PANTHEON Community-Based Smart City Digital Twin Platform, providing comprehensive dataset documentation, access mechanisms, and technical specifications for multi-hazard emergency response data management. The scope includes all datasets, schemas, security protocols, and access interfaces required to support disaster risk management operations across four primary emergency scenarios: **earthquake planning** (Attica, Greece), **heatwave planning** (Vienna, Austria), **wildfire training** (Attica, Greece), and **cyberattack training** (Vienna, Austria).

Geographic Coverage: The repository provides comprehensive data coverage for two primary European urban regions: the Athens/Attica metropolitan area in Greece (covering approximately 2,500 km² with coordinate bounds 23.3°E-24.0°E, 37.8°N-38.3°N) and the Vienna metropolitan area in Austria (covering approximately 415 km² with coordinate bounds 16.2°E-16.6°E, 48.1°N-48.3°N). This geographic scope was selected to represent diverse climatic conditions, urban planning approaches, and hazard exposure profiles characteristic of Mediterranean and Central European environments.

Temporal Coverage: Datasets span multiple temporal scales depending on data type and emergency scenario requirements. Base demographic and infrastructure datasets utilize 2020 reference periods with annual update cycles. Environmental datasets range from historical archives (ERA5 reanalysis from 1979-present) to real-time feeds (weather APIs, UAV data streams). Simulation outputs support time horizons from immediate response (minutes to hours) to strategic planning (72-hour post-event projections). All datasets include comprehensive temporal metadata enabling appropriate usage for both historical analysis and operational planning.

Data Volume and Complexity: The repository manages over 1,000 individual datasets totaling multiple terabytes of structured, semi-structured, and unstructured data. This includes high-resolution population grids (~100-meter resolution), detailed infrastructure mappings (building-level precision), environmental monitoring data (hourly to sub-hourly temporal resolution), and real-time simulation outputs (15-minute to real-time update frequencies). Dataset complexity ranges from simple tabular formats (CSV) to complex geospatial structures (GeoJSON, GeoTIFF) and streaming data formats (Kafka topics).

1.3.2 DATA CATEGORIES AND ASSET CLASSIFICATIONS

Foundational Geographic Datasets: Comprehensive base layers including OpenStreetMap infrastructure (roads, buildings, utilities), Copernicus Sentinel-2 satellite imagery (10-meter resolution), NASA SRTM elevation models (30-meter resolution), and administrative boundaries. These datasets provide consistent geographic foundations across all emergency scenarios and support spatial analysis, visualization, and routing applications.

Population and Demographic Datasets: High-resolution population density grids from UN/WorldPop collaboration, specialized vulnerable population distributions (elderly, children, institutionalized populations), and facility-specific demographic data (schools, hospitals, nursing homes). All demographic datasets utilize 2020 reference periods with approximately 100-meter spatial resolution, enabling detailed exposure assessment and evacuation planning.

Infrastructure and Point-of-Interest Datasets: Comprehensive mappings of critical facilities including healthcare infrastructure (hospitals, clinics), educational facilities (schools, universities), emergency services

(police, fire, medical), cooling infrastructure (parks, air-conditioned facilities), and transportation networks (roads, public transit). Infrastructure datasets include capacity estimates, operational parameters, and accessibility characteristics essential for emergency response planning.

Environmental and Hazard-Specific Datasets: Scenario-tailored environmental data including seismic hazard parameters (Peak Ground Acceleration, soil classifications), meteorological data (ERA5 reanalysis, real-time weather feeds), wildfire fuel mapping (ArcFuel project data), solar radiation data (Google Solar API), and specialized hazard modelling inputs. Environmental datasets support both historical analysis and real-time operational decision-making.

Simulation and Model Output Datasets: Dynamic datasets generated by PANTHEON processing components including fire propagation simulations, earthquake damage assessments, population allocation optimization results, blocked road networks, optimal routing solutions, and resource distribution plans. Simulation outputs utilize standardized formats enabling integration across multiple processing components and external systems.

Real-Time and Streaming Datasets: Dynamic data feeds including UAV imagery and telemetry, traffic monitoring data, weather observation feeds, and emergency response coordination data. Streaming datasets utilize Kafka messaging infrastructure with standardized topic naming conventions enabling real-time integration with operational systems.

1.3.3 TECHNICAL AND OPERATIONAL OBJECTIVES

Objective 1: Unified Data Access and Interoperability Establish comprehensive, standardized access mechanisms enabling seamless dataset integration across emergency response applications, research platforms, and operational systems. This includes RESTful API specifications with standardized parameter structures, OAuth2/OpenID Connect authentication protocols, and support for both JSON and CSV export formats. The repository provides consistent coordinate reference systems (WGS84), standardized metadata schemas, and format compatibility ensuring datasets can be efficiently integrated into existing GIS platforms, analytical tools, and emergency response systems.

Objective 2: Multi-Scenario Data Integration and Cross-Analysis Enable comprehensive analysis across different disaster types through shared foundational datasets, standardized schemas, and consistent quality frameworks. The repository supports comparative studies analysing different hazard impacts on common infrastructure, population exposure assessments across multiple threat types, and integrated vulnerability analysis combining different disaster scenarios. Standardized data models ensure that analysis methodologies developed for one scenario can be efficiently adapted to other hazard types.

Objective 3: Real-Time Operational Support and Scalability Provide high-performance data access supporting both strategic planning activities and real-time emergency response operations. The repository architecture scales from research applications requiring comprehensive historical datasets to operational environments demanding sub-second data access for critical decision support. Kafka-based streaming infrastructure supports real-time data integration, while federated storage systems ensure consistent performance across diverse access patterns and user loads.

Objective 4: Comprehensive Security and Compliance Framework Implement robust security mechanisms protecting sensitive emergency response data while enabling necessary multi-agency collaboration. Security objectives include role-based access control enabling appropriate data sharing across emergency response organizations, comprehensive encryption (TLS 1.2+ in transit, AES-256 at rest) protecting sensitive infrastructure and population data, audit logging providing complete traceability of data access and usage,

and GDPR compliance ensuring appropriate handling of potentially sensitive demographic and infrastructure information.

1.3.4 DELIVERABLE BOUNDARIES AND EXCLUSIONS

Included Components: Complete dataset catalogue with comprehensive metadata, API specifications and authentication protocols, security framework documentation, data schema definitions, ingestion and processing pipeline documentation, query and retrieval mechanism specifications, and system architecture documentation enabling repository deployment and operation.

Excluded Components: This deliverable does not include software implementation code for processing components, detailed deployment infrastructure specifications, specific hardware requirements or procurement guidance, user interface implementations, or training materials for end-user applications. The focus remains on data specifications, access mechanisms, and repository management rather than application development or system deployment.

Data Licensing and Usage Constraints: All PANTHEON-generated datasets are provided under open access principles, while third-party datasets (OpenStreetMap, HumData, Copernicus) retain their original licensing terms. Users must comply with applicable licensing requirements and acknowledge data sources appropriately. Sensitive infrastructure datasets may include access restrictions based on user authorization levels and intended usage purposes.

This comprehensive scope ensures that the PANTHEON secure data repository provides robust, scalable, and secure foundation for multi-hazard emergency response operations while maintaining compatibility with existing emergency management systems and research platforms across European and international contexts.

2. PANTHEON DATASET FRAMEWORK

2.1 PANTHEON REFERENCE ARCHITECTURE

The architecture of the Pantheon system was previously detailed in D3.7 (Pantheon Consortia, 2024).The security of Pantheon is described within the structural layered architecture to illustrate the placement in the implementation of the system.

This deliverable relates to the Secured Data Store component developed under SIMAVI's responsibility and is embedded in the Data/Persistence Layer (L1) of the PANTHEON system architecture. As shown in the PANTHEON architectural blueprint (D3.7 and figure below), this layer is responsible for the secure storage, indexing, and retrieval of heterogeneous data generated or consumed across the platform.

The repository is designed to store:

- simulation-generated data (from THL and INTEROPT),
- synthetic and conceptual models (e.g. INTEROPT, UPC, PRA),
- external data from UAVs, IoT sensors, and satellite observations (via the Data Connector),
- and other domain-specific datasets (e.g. infrastructure, traffic, and emergency data).

This storage layer interacts with:

- Model Management in L2 for input/output of AI and simulation models,
- Backend services in L3, which retrieve historical or real-time data for DSS modules,
- and the Data Connector, which ingests multiple data streams and routes them into appropriate repositories using standard protocols.

The Secured Data Store includes technologies such as:

- MinIO for object storage (e.g. satellite images, CSV, GeoJSON),
- PostgreSQL and Neo4j for structured and graph data,
- and is orchestrated in a containerized environment (Docker Swarm).

Authentication and access control are managed centrally through the Keycloak-based IAM system, which enforces secure, role-based access across all layers (L1–L5). The diagram below (Figure 1) illustrates the repository's integration within the full-stack architecture of the PANTHEON platform.

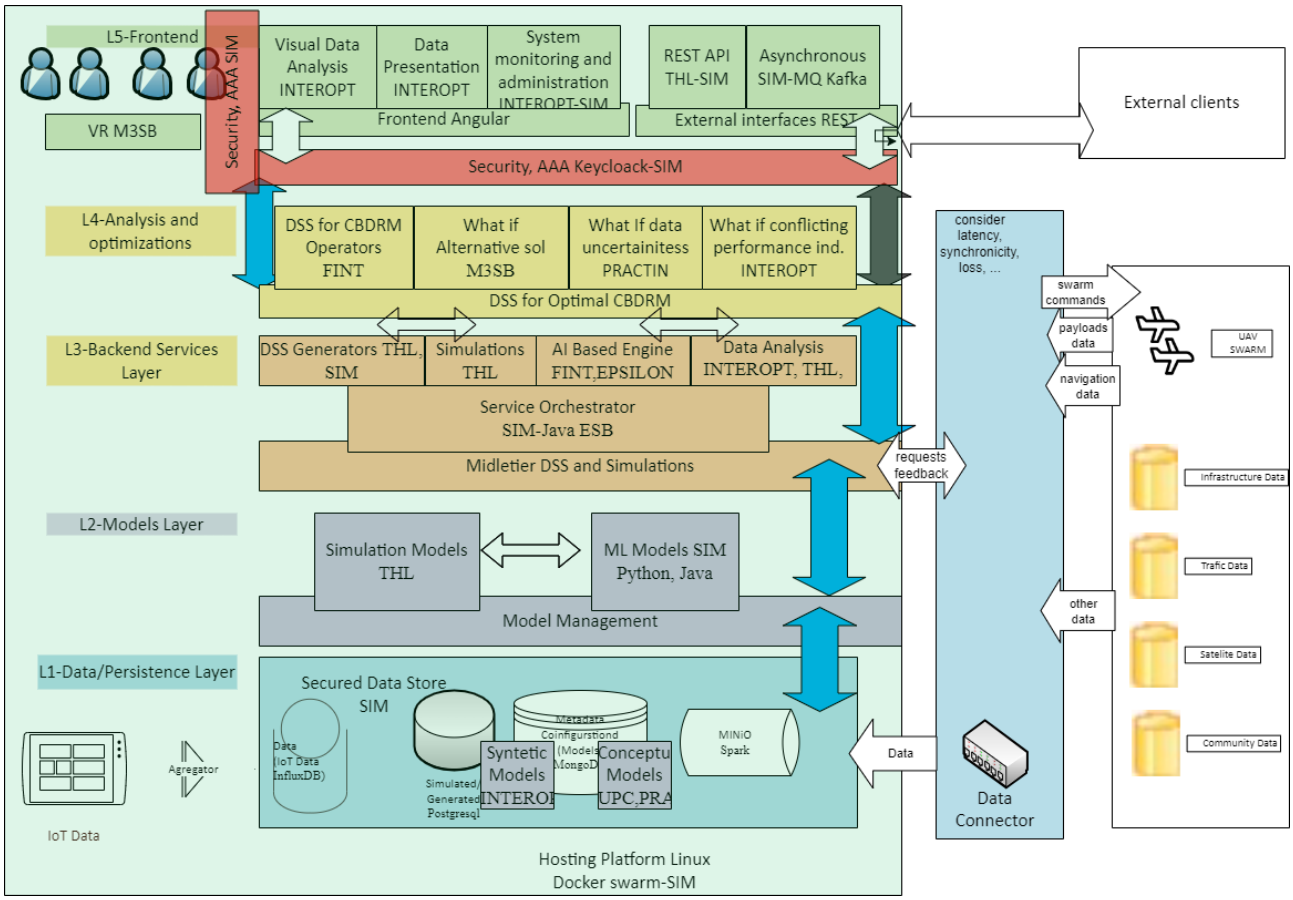

Figure 1. The Layered architecture of PANTHEON

The layers of the proposed system are:

- L1- data and persistence Layer. This layer is responsible for data storage and maintenance
- L2-Models Layer. This layer is responsible for model management. Simulation and ML models are part of this layer. Model storage is in L1, but their management is considered in this layer.
- L3-Backend Services Layer. It provides the homogeneous access of all services in the backend (L1 and L2) to the upper layers. Backup layer also organises the backend services by their semantics.
- L4-Analysis and Optimizations. This layer is placed on top of L3 and is also a backend layer but is specialized in analysis and optimization of data provided at the level of Layer 3.
- L5-Frontend. This layer is placed on top of the architecture and is responsible for presenting data to final users. The presentation can be in a user interface or exposed as APIs.

In the present document, the focus is primarily on L1, and Security Layer (user authentication) which concerns the security of the data, and security of the user.

2.2 DATA MODEL OVERVIEW (EPSILON)

Pantheon's secure data repository shall provide a structured and flexible foundation for handling many different datasets across the platform. It will support the organization, access and interconnection of data from simulations, UAVs, sensors and user interfaces. By leveraging technologies such as GraphDB, the model shall ensure interoperability between different types of data across the platform.

Built to accommodate both spatial and non-spatial data, the model represents entities such as incidents, assets, geofenced zones, sensor observations, and temporal events. Each entity is linked through unique identifiers and metadata describing source, time, spatial coverage, and confidentiality level. The model supports both static and dynamic data and aligns with the layered system architecture, particularly Layer 1 (persistence) and Layer 3 (backend services).

The implementation happens across multiple storage technologies. Neo4j and GraphDB handle semantic and relational graphs, while PostgreSQL is used for structured records MinIO manages unstructured files such images and Kafka handles real-time streams. Spatial data is integrated using standard formats such as GeoJSON and Shapefiles (SHP), stored via PostGIS enabled databases and object storage.

2.3 DATA TYPES CLASSIFICATION (EPSILON)

The repository should support a wide variety of data types, categorized by structure, origin, spatial format, temporal characteristics and security level. Structured data, such as relational tables and registries, is stored in PostgreSQL. Semi-structured formats like JSON and GeoJSON are used in API exchanges, while unstructured data—UAV imagery, videos, logs—is handled via MinIO object storage. Graph-structured relationships are stored using Neo4j and GraphDB.

Geospatial data such as vector-based datasets, such as administrative boundaries, hazard zones, road networks, and asset locations, are stored in formats compatible with GeoJSON, SHP, and PostGIS-enabled PostgreSQL tables. Raster datasets, including gridded risk heatmaps, satellite imagery, and terrain models, are stored in formats such as GeoTIFF and served through object storage or geospatial APIs. This supports compatibility with GIS platforms and supports spatial visualization.

Temporal classification includes static (e.g., base maps), dynamic (e.g., live sensor data), categories. Each type informs appropriate storage strategies and retrieval mechanisms. Data is further tagged by origin (system-, user-, or externally sourced) and sensitivity level (public, internal, or restricted), informing access control and encryption strategies managed via Keycloak and described in Section 4.

2.4 SOURCES, FORMATS AND FREQUENCIES OF DATA

The PANTHEON platform ingests and manages a wide variety of datasets, originating from both internal platform components and external data providers. These datasets can be classified as follows:

- **Sensor and IoT Data:** Captured in real time or near real-time from environmental sensors (e.g. temperature, gas, humidity) or infrastructure monitors deployed in pilot areas.
- **UAV and Satellite Data:** Collected from drone missions or remote sensing platforms (e.g. Copernicus, Google Solar API), often in raster or GeoTIFF format.
- **Simulation Outputs:** Generated by modules responsible for modeling earthquake impacts, wildfire spread, heatwave allocation, and other critical events.
- **Open and Institutional Data Sources:** Includes statistics from HumData, OpenStreetMap (OSM), Google Maps API, or local authorities.
- **Synthetic and Conceptual Models:** Manually constructed or machine-generated datasets used for training or testing (e.g., synthetic population distributions, model templates).

The data managed within the repository comes in diverse formats, such as:

- **GeoJSON / Shapefiles (.shp):** For spatial features (roads, buildings, infrastructure, affected zones).

- **CSV / XLSX:** For tabular data, population statistics, precomputed distances.
- **GeoTIFF / NetCDF:** For raster data (e.g. solar radiation, elevation, meteorological maps).
- **JSON / XML:** For structured metadata or simulation configurations.

Frequencies range from static datasets (e.g. urban infrastructure maps) to periodically updated files (e.g. weather data, social vulnerability metrics).

2.5 DISASTER SCENARIO COVERAGE

The PANTHEON platform provides comprehensive disaster scenario coverage across four major hazard types, encompassing both planning and training applications in two distinct European urban environments. This section outlines the geographic, temporal, and thematic coverage of each disaster scenario, establishing the scope and boundaries of the secure data repository.

GEOGRAPHIC COVERAGE OVERVIEW

The PANTHEON platform focuses on two primary European urban regions, selected to represent diverse climatic, geographic, and urban planning characteristics:

Athens/Attica Region, Greece

- **Coordinate Bounds**: 23.3°E - 24.0°E, 37.8°N - 38.3°N
- **Coverage Area**: ~2,500 km²
- **Urban Population**: ~3.8 million (Greater Athens Area)
- **Scenarios Covered**: Earthquake (Planning), Wildfire (Training)
- **Key Geographic Features**:
    - Mediterranean coastal environment
    - Mountainous terrain with fire-prone vegetation
    - Dense urban core with sprawling suburbs
    - Seismic activity zone

Vienna, Austria

- **Coordinate Bounds**: 16.2°E - 16.6°E, 48.1°N - 48.3°N
- **Coverage Area**: ~415 km²
- **Urban Population**: ~1.9 million (Greater Vienna Area)
- **Scenarios Covered**: Heatwave (Planning), Cyberattack (Training)
- **Key Geographic Features**:
    - Continental European climate
    - Danube River valley environment
    - Well-planned urban infrastructure
    - High population density districts

SCENARIO-SPECIFIC COVERAGE ANALYSIS

Earthquake Scenario Coverage (Attica, Greece)

Geographic Scope:

- **Primary Focus Area**: Fyli region, West Attica
- **Detailed Coverage**: 23.66°E - 23.72°E, 38.10°N - 38.14°N
- **Coverage Resolution**: Building-block level (centroids)
- **Infrastructure Elements**:
    - Road networks via OpenStreetMap
    - Building vulnerability assessments
    - Emergency assembly points
    - Critical infrastructure locations

Temporal Coverage:

- **Base Data Period**: 2020-2025
- **Simulation Timeframe**: Real-time to 72-hour post-event
- **Update Frequency**: Static infrastructure (annual), dynamic routing (real-time)
- **Historical Context**: Integration with historical seismic records

Hazard Parameters Covered:

- **Seismic Intensity**: Modified Mercalli Scale VII-VIII
- **Peak Ground Acceleration**: 200-300 cm/s²
- **Soil Types**: Rock, Stiff, Soft classifications
- **Vulnerability Classes**: Low to Moderate building vulnerability
- **Population Exposure**: High-resolution demographic grids

Heatwave Scenario Coverage (Vienna, Austria)

Geographic Scope:

- **Coverage Area**: All 23 Vienna administrative districts
- **Infrastructure Detail**:
    - 289 parks with capacity data
    - 139 nursing homes with resident counts
    - 29 hospitals and 85 schools
    - Cooling infrastructure network

Temporal Coverage:

- **Baseline Data**: 2020 demographics and infrastructure
- **Seasonal Focus**: Summer months (June-September)
- **Event Duration**: 3-14 day heatwave periods

- **Climate Projections**: Integration with future climate scenarios

Environmental Parameters Covered:

- **Temperature Thresholds**: >35°C daily maximum, >20°C nightly minimum
- **Solar Radiation**: Monthly flux data from Google Solar API
- **Urban Heat Island**: District-level temperature variations
- **Population Vulnerability**: Age-stratified risk assessment
- **Cooling Capacity**: Infrastructure-based mitigation resources

Coverage Completeness:

- **Complete Coverage**: Public cooling facilities, administrative districts
- **High Coverage**: Elderly population distribution, park capacity
- **Moderate Coverage**: Private cooling resources, building-level thermal data
- **Limited Coverage**: Individual health status, behavioral adaptation patterns

Wildfire Scenario Coverage (Attica, Greece)

Geographic Scope:

- **Fire-Prone Areas**: West Attica wildland-urban interface
- **Vegetation Mapping**: ArcFuel project coverage for fuel load assessment
- **Elevation Coverage**: 30-meter resolution digital elevation model
- **Weather Monitoring**: ERA5 reanalysis data for historical and current conditions

Temporal Coverage:

- **Fire Season Focus**: May-October high-risk periods
- **Simulation Resolution**: 15-minute fire progression intervals
- **Weather Integration**: Hourly meteorological data
- **Historical Context**: Past fire perimeter data for model validation

Fire Behavior Parameters:

- **Fuel Types**: Mediterranean shrubland, forest, grassland classifications
- **Topographic Influence**: Slope and aspect analysis for fire spread
- **Weather Dependencies**: Wind speed/direction, humidity, temperature
- **Suppression Resources**: Fire station locations and response capabilities

Coverage Limitations:

- **Real-time Fuel Moisture**: Limited ground-truth measurements
- **Suppression Effectiveness**: Simplified resource modeling
- **Smoke Dispersion**: Basic atmospheric modeling
- **Evacuation Behavior**: Simplified population response models

Cyberattack Scenario Coverage (Vienna, Austria)

Geographic Scope:

- **Critical Infrastructure**: Power, telecommunications, transportation networks
- **Urban Density Areas**: High-impact zones for service disruption
- **Emergency Response Network**: Hospitals, police, fire stations
- **Explosion Impact Zone**: 0.5-4 km radius depending on scenario parameters

Temporal Coverage:

- **Attack Scenarios**: Immediate impact to 48-hour recovery period
- **Cascade Effects**: Infrastructure interdependency failure progression
- **Response Timeline**: Emergency service activation and coordination

Threat Parameters Covered:

- **Cyber Infrastructure**: SCADA systems, communication networks
- **Physical Impact**: Explosion damage assessment
- **Smoke/Chemical Dispersion**: Gaussian plume modeling
- **Population Exposure**: Building-level impact assessment
- **Service Disruption**: Multi-sector infrastructure failure analysis

Coverage Constraints:

- **Classified Infrastructure**: Limited detail on sensitive systems
- **Attack Vectors**: Simplified cyber intrusion modeling
- **Interdependency Complexity**: Reduced-order infrastructure models
- **Recovery Processes**: Simplified restoration timelines

CROSS-SCENARIO COVERAGE INTEGRATION

All scenarios utilize common foundational datasets that ensure consistency across different hazard types:

Population Demographics:

- **Austria**: High-resolution WorldPop 2020 estimates
- **Greece**: UN/WorldPop demographic grids
- **Resolution**: ~100-meter grid cells
- **Coverage Quality**: Complete urban area coverage

Infrastructure Base Layers:

- **OpenStreetMap**: Road networks, buildings, points of interest
- **Satellite Imagery**: Copernicus Sentinel-2 L2A true color
- **Administrative Boundaries**: Official district/municipality boundaries

Multi-Hazard Interactions

The platform supports analysis of compound and cascading disaster scenarios:

- COVERAGE QUALITY ASSESSMENT

Table 1. Data Quality Metrics

Scenario Component	Spatial Resolution	Temporal Resolution	Data Source	Update Frequency
Population Data	100m grid	Annual	HumData/WorldPop	Annual
Infrastructure	Building-level	Static/Real-time	OpenStreetMap	Monthly/Real-time
Environmental	30m-1km	Hourly-Daily	ERA5/Google Solar API	Hourly/Monthly
Hazard Models	Variable	Minutes-Hours	Simulation outputs	On-demand
Response Resources	Facility-level	Static	OSM/Google Places API	Monthly

Coverage Gaps and Limitations

Technical Limitations:

- Real-time structural health monitoring data availability
- Individual behavioral response modeling complexity
- Private infrastructure detailed vulnerability information
- Cross-border data sharing and coordination protocols

Data Availability Constraints:

- Subsurface infrastructure comprehensive mapping
- Detailed building material composition databases
- Real-time traffic and crowd density information
- Economic impact assessment detailed methodologies

Mitigation Strategies:

- Synthetic data generation for missing elements
- Statistical modeling for behavioral parameters
- Simplified assumptions for complex systems
- Regular data update and validation cycles

COVERAGE VALIDATION AND VERIFICATION

Validation Methods

- **Ground Truth Comparison**: Validation against historical events and field data
- **Expert Review**: Subject matter expert assessment of scenario realism
- **Cross-Reference Analysis**: Comparison with authoritative datasets

- **Stakeholder Feedback**: End-user validation of scenario relevance

Quality Assurance Procedures

- **Automated Quality Checks**: Spatial bounds, temporal consistency, format compliance
- **Manual Review Processes**: Critical dataset validation and metadata verification
- **Version Control**: Comprehensive tracking of data updates and modifications
- **Documentation Standards**: Complete provenance and quality documentation

This comprehensive disaster scenario coverage framework ensures that the PANTHEON platform provides robust, reliable, and scalable emergency response capabilities across diverse hazard types and urban environments, while maintaining high standards for data quality and system interoperability.

2.6 DATA SOURCES AND PROVENANCE

The PANTHEON platform integrates datasets from multiple authoritative sources, ensuring comprehensive coverage of emergency response requirements while maintaining data quality, reliability, and legal compliance. This section provides complete documentation of data sources, acquisition methods, licensing terms, and provenance tracking mechanisms essential for understanding dataset authority and appropriate usage constraints.

PRIMARY DATA SOURCE CATEGORIES

International and Supranational Organizations

United Nations / WorldPop Collaboration (HumData)

- **Organization**: Humanitarian Data Exchange / WorldPop Research Group
- **Website**: https://data.humdata.org/
- **Datasets**: High-resolution population density grids
- **Coverage**: Austria (aut_general_2020), Greece (grc_general_2020)
- **Resolution**: ~100-meter grid cells
- **Temporal Coverage**: 2020 demographic estimates
- **Licensing**: Creative Commons Attribution International
- **Update Frequency**: Annual
- **Acquisition Method**: Direct download from HumData platform
- **Data Authority**: UN agencies in collaboration with academic institutions
- **Quality Assurance**: Validated against national census data and survey records

European Space Agency / Copernicus Programme

- **Organization**: European Space Agency (ESA)
- **Website**: https://dataspace.copernicus.eu/
- **Primary Datasets**:
    - Sentinel-2 L2A satellite imagery (true color, 10-meter resolution)

    - Copernicus Digital Elevation Model (30-meter resolution)
- **Coverage**: Both Vienna and Athens/Attica regions
- **Licensing**: Full, open and free access under Copernicus data policy
- **Update Frequency**: Satellite imagery (5-day revisit), DEM (static)
- **Acquisition Method**: Copernicus Data Space Ecosystem API
- **Quality Standards**: ESA processing algorithms with geometric and radiometric corrections

European Meteorological and Environmental Agencies

European Centre for Medium-Range Weather Forecasts (ECMWF)

- **Organization**: ECMWF (intergovernmental organization)
- **Website**: https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5
- **Dataset**: ERA5 Reanalysis Data
- **Temporal Coverage**: 1979-present
- **Spatial Resolution**: 0.25° × 0.25° (approximately 31 km)
- **Temporal Resolution**: Hourly
- **Parameters**: Wind speed/direction, temperature, humidity, precipitation, pressure
- **Usage**: Wildfire scenario meteorological modeling
- **Licensing**: Copernicus Climate Change Service license
- **Access Method**: Climate Data Store API

ArcFuel Project (Greece)

- **Organization**: Laboratory of Forest Management and Remote Sensing, Aristotle University
- **Website**: https://fmrs.web.auth.gr/project/arcfuel/
- **Dataset**: Mediterranean Vegetation Fuel Maps
- **Coverage**: West Attica region
- **Content**: Fuel type classification, fuel load density, moisture content indicators
- **Format**: GeoTIFF
- **Usage**: Wildfire spread rate calculations and burn intensity predictions
- **Licensing**: Academic research license
- **Quality**: Field-validated fuel models specific to Mediterranean ecosystems

National Emergency Management Agencies

KEMEA (Greek Emergency Management Agency)

- **Organization**: Κέντρο Μελετών Ασφάλειας (KEMEA)
- **Country**: Greece
- **Datasets**:
    - Building vulnerability assessments for Fyli region

  - Seismic hazard parameters (Peak Ground Acceleration, soil classifications)
  - Earthquake magnitude and intensity data
- **Format**: Shapefiles (.shp), CSV
- **Coverage**: Fyli region, Attica
- **Authority**: Official Greek government emergency management data
- **Usage**: Earthquake scenario infrastructure vulnerability modeling
- **Access**: Institutional partnership agreement
- **Quality**: Based on official seismic hazard assessments and building surveys

Commercial and Technology Providers

Google Earth Engine / Google Solar API

- **Organization**: Google LLC
- **Service**: Google Solar API (data-layers API)
- **Dataset**: Solar radiation maps
- **Coverage**: Vienna region (specific locations)
- **Format**: Monthly Flux GeoTIFF (processed to CSV)
- **Temporal Resolution**: Monthly data
- **Units**: Solar radiation in kWh/m²/day
- **Usage**: Heatwave scenario heat exposure assessment
- **Licensing**: Google API Terms of Service
- **Access Method**: RESTful API with authentication tokens

OpenWeatherMap

- **Organization**: OpenWeather Ltd.
- **Website**: https://openweathermap.org
- **Dataset**: Real-time weather data
- **Coverage**: Vienna region
- **Parameters**: Current weather conditions, wind speed/direction, temperature, humidity, precipitation, visibility
- **Update Frequency**: Real-time/hourly
- **Usage**: Cyberattack scenario dispersion modeling, general weather impact assessment
- **Licensing**: Commercial API license
- **Access Method**: REST API with subscription-based access

Open Source and Community-Driven Sources

OpenStreetMap Foundation

- **Organization**: OpenStreetMap Foundation (OSMF)
- **Website**: https://www.openstreetmap.org
- **Datasets**:
  - Road networks and transportation infrastructure
  - Building footprints and geometries
  - Points of interest (hospitals, schools, emergency services)
  - Utility infrastructure (power, telecommunications)
- **Coverage**: Global (Vienna and Athens/Attica regions)
- **Licensing**: Open Database License (ODbL)
- **Access Method**: OSMNX Python package, Overpass API
- **Quality**: Community-validated with regular updates
- **Authority**: Crowd-sourced with institutional contributions

OpenInfraMap

- **Organization**: OpenInfraMap Project
- **Website**: https://openinframap.org/
- **Dataset**: Infrastructure network data derived from OpenStreetMap
- **Content**: Power infrastructure, telecommunications, transportation networks, utility networks
- **Usage**: Cross-scenario infrastructure interdependency analysis
- **Licensing**: Open Database License (ODbL)
- **Quality**: Enhanced OSM data with infrastructure-specific validation

DATA ACQUISITION AND INGESTION METHODS

Automated API Integration

Real-time Sources:

- OpenWeatherMap: Hourly automated polling via REST API
- Google Solar API: Monthly data retrieval with automatic processing
- Copernicus services: Scheduled downloads via official APIs

Batch Processing:

- ERA5 data: Weekly batch downloads from Climate Data Store
- Satellite imagery: Event-triggered downloads based on coverage requirements
- Population data: Annual updates from HumData platform

Institutional Partnerships

KEMEA Collaboration:

- Direct data transfer via secure protocols
- Institutional agreements for sensitive infrastructure data
- Quality verification through joint validation procedures

Academic Collaborations:

- ArcFuel data through research partnerships
- Validation datasets from partner institutions
- Shared processing algorithms and methodologies

Manual Curation and Processing

OpenStreetMap Processing:

- OSMNX-based extraction for specific geographic regions
- Custom filtering and validation for emergency response applications
- Regular updates synchronized with OSM change feeds

Scenario-Specific Processing:

- Custom processing of elevation data for UAV operations
- Spatial joins of demographic data with infrastructure layers
- Temporal aggregation of meteorological data for scenario requirements

DATA QUALITY AND VALIDATION PROCEDURES

Source Authority Verification

Government and International Organizations: Direct validation against official sources, institutional partnerships ensuring data authenticity, regular updates from authoritative providers.

Scientific and Academic Sources: Peer-review validation of methodologies, cross-validation with multiple academic sources, citation tracking for scientific reproducibility.

Commercial Sources: Service level agreements defining data quality standards, validation against independent data sources, monitoring of data provider reliability.

Technical Quality Assurance

Spatial Validation:

- Coordinate system verification (WGS84 compliance)
- Spatial bounds checking against known geographic extents
- Geometric validity testing for vector datasets

- Resolution and accuracy assessment for raster datasets

Temporal Validation:

- Timestamp consistency checking across time series data
- Temporal coverage verification against specified requirements
- Update frequency monitoring for dynamic data sources
- Version control for dataset revisions

Content Validation:

- Range checking for numerical data (population densities, elevation values)
- Completeness assessment for required fields and attributes
- Cross-reference validation between related datasets
- Schema compliance verification for structured data

LICENSING AND USAGE RIGHTS

Open Access Datasets

Public Domain: certain government datasets **Creative Commons:** HumData/WorldPop population data (CC BY) **Open Database License:** OpenStreetMap and derived datasets (ODbL) **Copernicus License:** ESA satellite imagery and environmental data

Restricted and Commercial Licenses

Google Services: Google Solar API (Terms of Service compliance required) **OpenWeatherMap:** Commercial API license with usage limitations **KEMEA Data:** Institutional agreement with usage restrictions for sensitive infrastructure data **ArcFuel:** Academic research license with citation requirements

PANTHEON-Generated Data

Simulation Outputs: Open access under PANTHEON project licensing **Processed Datasets:** Derivative works licensed according to source data constraints **Synthetic Data:** Project-generated synthetic datasets available under open licenses

DATA UPDATE AND MAINTENANCE STRATEGIES

Table 2. Update Frequency Classifications

Source Category	Update Frequency	Method	Examples
Static Reference	Once/Rarely	Manual download	COPERNICUS, administrative boundaries
Annual Updates	Yearly	Scheduled batch	Population statistics, infrastructure surveys
Periodic Updates	Monthly/ Quarterly	Automated API	Satellite imagery, weather normals
Dynamic Data	Daily/Hourly	Real-time API	Weather data, traffic information

Event-Triggered	As needed	Manual/Automated	Emergency response data, simulation results

Data Freshness Monitoring

Quality Degradation Detection: Algorithms detect when data sources become stale or unreliable.

Alternative Source Activation: Backup data sources activated when primary sources become unavailable.

This comprehensive data sources and provenance framework ensures that all PANTHEON datasets maintain complete traceability, appropriate attribution, and compliance with licensing requirements while providing users with the information necessary to evaluate dataset appropriateness for their specific emergency response applications.

2.7 DATASET INTERDEPENDENCIES AND WORKFLOWS

The PANTHEON platform implements a sophisticated data workflow architecture that ensures seamless integration and processing of heterogeneous datasets across all emergency scenarios. This section documents the dataset interdependencies, processing pipelines, and data flow patterns that enable real-time decision support and training capabilities.

COMMON WORKFLOW ARCHITECTURE

All PANTHEON scenarios follow a standardized five-layer workflow architecture that ensures consistency, scalability, and interoperability:

CORE PROCESSING PIPELINE

The fundamental data flow pattern across all scenarios follows this sequence:

User Interface → Simulation/Processing Components → Data Storage → Kafka Messaging → Decision Support Systems

SHARED COMPONENTS

User Interface: Single entry point for scenario initiation and parameter configuration

GeoServer: Central geospatial data hub providing standardized access to geographic datasets

Database Storage: Multiple storage systems (PostgreSQL/PostGIS, MinIO, Neo4j) with scenario-specific organization

Kafka Messaging: Real-time data streaming with standardized topic naming conventions

UAV Swarms: Cross-scenario component for aerial data collection and validation

Self-adaptation/Traffic: Dynamic optimization component for route and resource allocation

DATA FLOW PATTERNS

Input Data Acquisition: Population density, weather data, infrastructure datasets retrieved from GeoServer

Simulation Processing: Scenario-specific impact assessment and vulnerability analysis

Intermediate Storage: Results stored in databases with appropriate formatting

Kafka Notification: Processing completion triggers topic-specific messages

Decision Support: Downstream components consume data for optimization and resource allocation

Output Generation: Final results stored with metadata and made available via APIs

SCENARIO-SPECIFIC WORKFLOWS

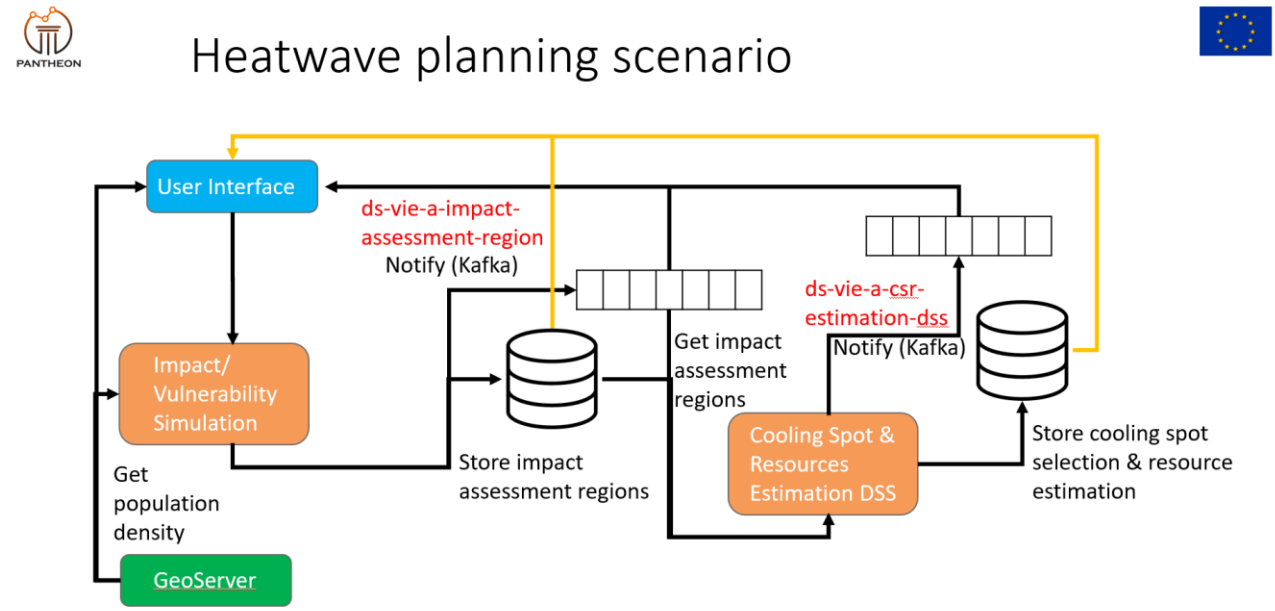

Figure 2. Heatwave Planning Scenario (Vienna)

Primary Data Dependencies:

- **Input**: Population density data from GeoServer
- **Processing**: Impact/Vulnerability Simulation assesses heat exposure risks
- **Intermediate**: Impact assessment regions stored in database
- **Decision Support**: Cooling Spot & Resources Estimation DSS processes allocation optimization
- **Output**: Cooling spot selection and resource estimation stored with format specifications

Key Interdependencies:

- Population density data feeds directly into vulnerability simulation
- Impact assessment results trigger Kafka notifications for downstream processing
- Cooling spot allocation depends on both population data and infrastructure capacity
- Resource estimation requires cross-referencing with facility databases

Kafka Topics:

- **ds-vie-a-impact-assessment-region**: Impact assessment results (2-3 min processing time)
- **ds-vie-a-csr-estimation-dss**: Cooling spot and resource allocation results (O(1min) processing)

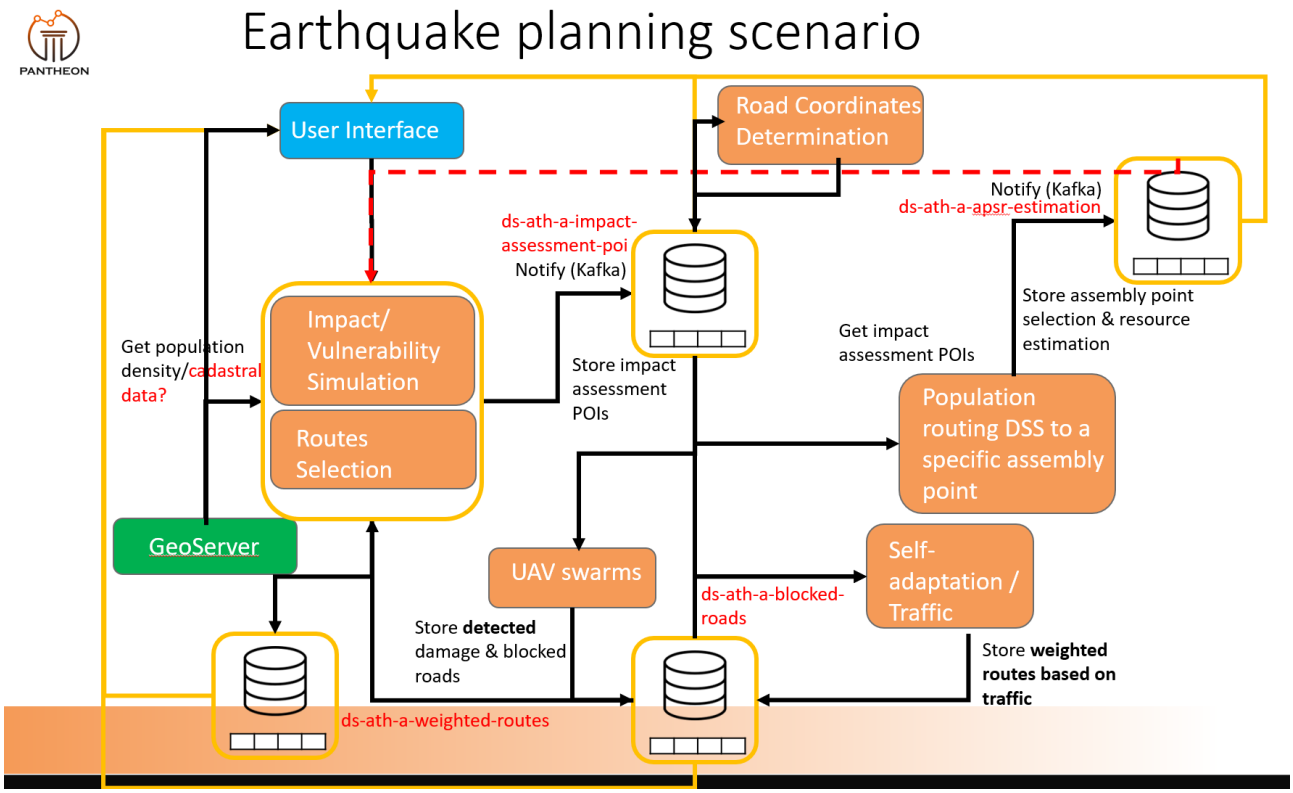

Figure 3. Earthquake Planning Scenario (Athens)

Primary Data Dependencies:

- **Input**: Population density and cadastral data from GeoServer
- **Processing**: Dual-component simulation (Impact/Vulnerability + Routes Selection)
- **Intermediate**: Impact assessment POIs and route optimization results
- **UAV Integration**: Damage detection and blocked road identification
- **Decision Support**: Population routing to assembly points and traffic optimization
- **Output**: Assembly point selection and weighted route storage

Key Interdependencies:

- Population and cadastral data feed into both impact simulation and route selection
- Impact assessment POIs inform assembly point selection algorithms
- UAV-detected damage updates route selection parameters dynamically
- Self-adaptation component optimizes traffic flow based on blocked road data
- Assembly point selection requires integration of population density, infrastructure damage, and accessibility analysis

Kafka Topics:

- **ds-ath-a-impact-assessment-poi**: Impact assessment points of interest (<1s processing)
- **ds-ath-a-blocked-roads**: Blocked road notifications (O(<10s) processing)
- **ds-ath-a-weighted-routes**: Optimized routing solutions

- **ds-ath-a-apsr-estimation**: Assembly point and resource estimation

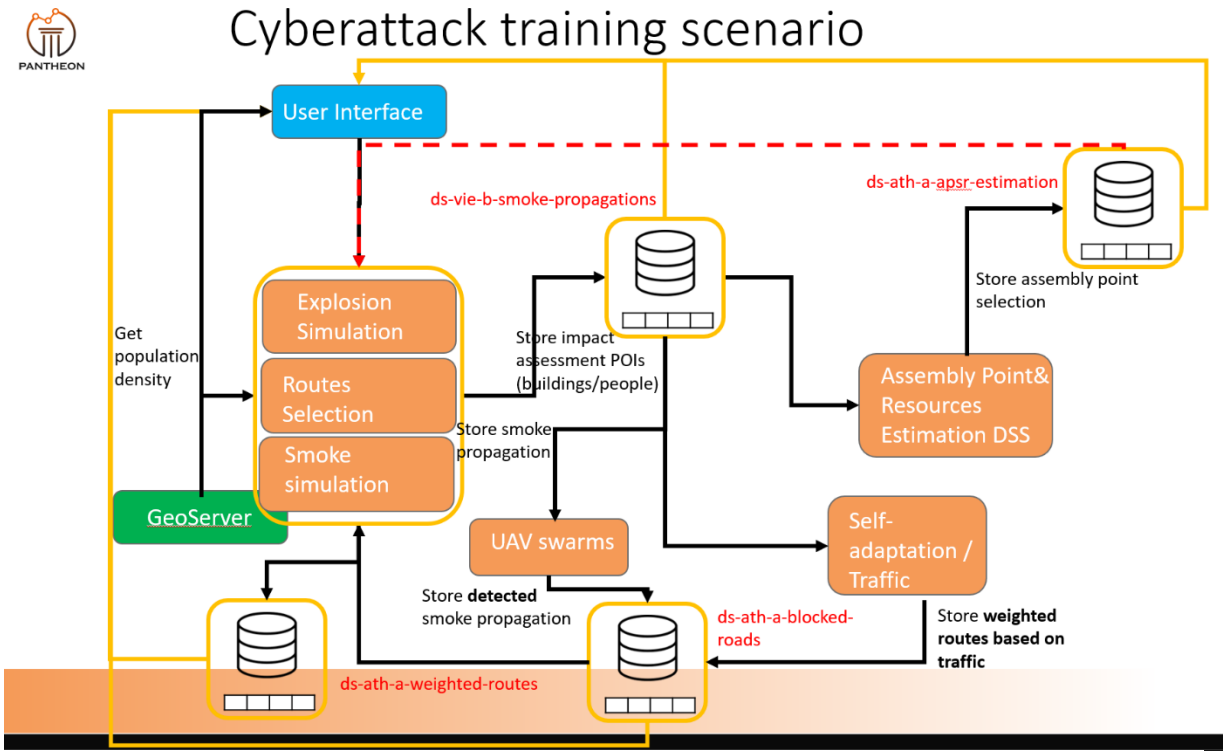

Figure 4. Cyberattack Training Scenario (Vienna)

Primary Data Dependencies:

- **Input**: Population density and weather data from GeoServer
- **Processing**: Triple-component simulation (Explosion + Routes Selection + Smoke Simulation)
- **Intermediate**: Impact assessment POIs and smoke propagation data
- **UAV Integration**: Damage assessment and smoke dispersion monitoring
- **Decision Support**: Assembly Point & Resources Estimation DSS
- **Output**: Assembly point selection and resource estimation (marked with restrictions)

Key Interdependencies:

- Weather data critical for smoke simulation accuracy
- Explosion simulation results inform both route selection and assembly point placement
- Smoke propagation affects UAV operational parameters and evacuation routing
- Assembly point selection must account for both explosion damage and smoke dispersion patterns
- Resource estimation considers contamination levels and accessibility constraints

Kafka Topics:

- **ds-vie-b-impact-assessment-poi**: Explosion impact assessment (O(30s) processing)
- **ds-vie-b-smoke-propagations**: Smoke dispersion modeling (O(1s) processing)

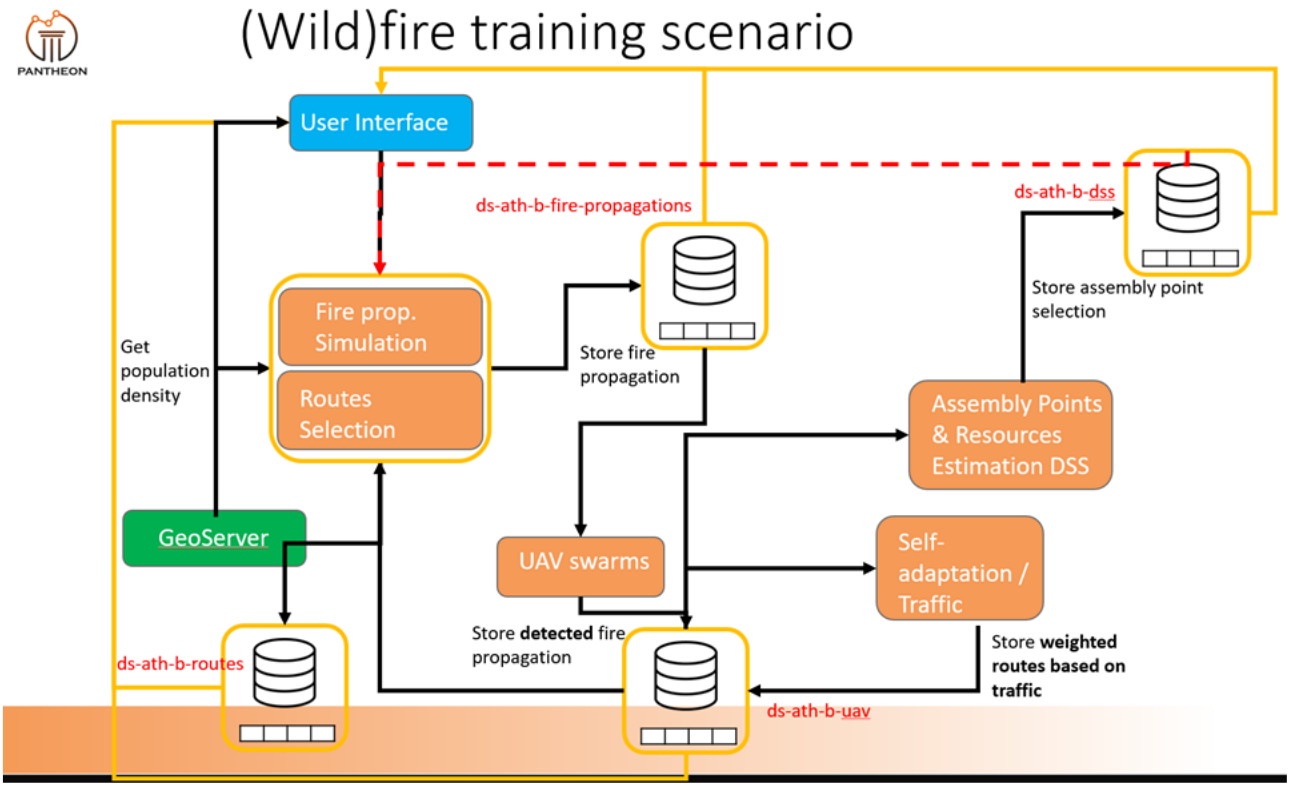

Figure 5. Wildfire Training Scenario (Attica)

Primary Data Dependencies:

- **Input**: Population density and weather data from GeoServer
- **Processing**: Fire propagation simulation with integrated route selection
- **Intermediate**: Fire propagation data storage
- **UAV Integration**: Fire detection and spread monitoring
- **Decision Support**: Assembly Points & Resources Estimation DSS
- **Output**: Assembly point selection and resource estimation (with uncertainties marked)

Key Interdependencies:

- Weather data essential for accurate fire propagation modeling
- Fire propagation results directly impact route selection algorithms
- UAV-detected fire progression updates simulation parameters in real-time
- Assembly point selection must account for dynamic fire boundaries and wind patterns
- Resource estimation considers fire spread predictions and evacuation time constraints

Kafka Topics:

- **ds-ath-b-fire-propagations**: Fire spread simulation results (<10s processing)
- **ds-ath-b-routes**: Fire-aware routing optimization
- **ds-ath-b-uav**: UAV fire monitoring data (<20s processing)

- **ds-ath-b-traffic**: Traffic optimization during evacuation
- **ds-ath-b-dss**: Decision support system results

CROSS-SCENARIO DATA DEPENDENCIES

Shared Input Dependencies

All scenarios rely on common foundational datasets that create cross-scenario interdependencies:

Population Demographics (HumData/WorldPop)

- Austria population grid: aut_general_2020 (Vienna scenarios)
- Greece population grid: grc_general_2020 (Attica scenarios)
- Provides baseline exposure assessment for all hazard types
- Updates propagate across all scenario impact calculations

Geographic Infrastructure (OpenStreetMap)

- Road networks for routing algorithms (all scenarios)
- Building footprints for impact assessment (earthquake, cyberattack)
- Points of interest for resource allocation (all scenarios)
- Changes in infrastructure data affect multiple scenario calculations simultaneously

Environmental Base Data

- Elevation models (Copernicus DEM) for UAV operations (all scenarios)
- Weather data sources for scenario-specific modeling
- Satellite imagery for visual context and change detection

Processing Component Dependencies

UAV Swarms Integration

- Earthquake: Damage assessment and blocked road detection
- Wildfire: Fire perimeter monitoring and spread validation
- Cyberattack: Damage assessment and smoke/contaminant tracking
- Heatwave: Infrastructure monitoring and crowd density assessment
- Dependency: UAV data updates affect route optimization and resource allocation across all scenarios

Self-adaptation/Traffic Optimization

- Receives inputs from all scenario-specific simulations
- Provides traffic-weighted route optimization for emergency response
- Dependencies: Real-time traffic data, blocked road information, and dynamic hazard boundaries

KAFKA TOPIC INTEGRATION AND MESSAGING PATTERNS

Unified Topic Naming Convention

All Kafka topics follow the standardized pattern:

ds-{city}-{scenario_letter}-{component_type}

City Codes:

- ath: Athens/Attica, Greece
- vie: Vienna, Austria

Scenario Letters:

- a: Planning scenarios (Earthquake-Attica, Heatwave-Vienna)
- b: Training scenarios (Wildfire-Attica, Cyberattack-Vienna)

Message Flow Patterns

Trigger-Based Processing

1. User Interface initiates scenario via user input
2. Primary simulation components process input data
3. Kafka notifications trigger downstream processing
4. Secondary components consume messages and begin processing
5. Final results published to output topics

Real-Time Updates

- UAV components publish continuous updates to monitoring topics
- Fire propagation publishes periodic updates (15-minute intervals)
- Traffic optimization publishes dynamic route updates
- Impact assessments publish threshold-based alerts

Cross-Component Communication

- Impact assessment results inform route selection algorithms
- UAV detection data updates simulation parameters
- Route optimization results inform resource allocation decisions
- Assembly point selection integrates multiple data streams

Processing Time Dependencies

Different components have varying processing requirements that affect workflow timing:

Fast Processing (<10s):

- Fire propagation simulation
- Blocked road detection
- Smoke propagation modeling
- Impact assessment (earthquake)

Medium Processing (1-3 min):

- Heatwave impact assessment
- Cooling spot allocation

- Population routing optimization

Complex Processing (>30s):

- Explosion simulation
- UAV data processing
- Assembly point optimization
- Resource estimation algorithms

STORAGE AND FORMAT CONSIDERATIONS

Format Specifications

The workflow diagrams highlight critical format decisions that affect data interoperability:

GeoJSON Formats

- Blocked roads: MultiLineString geometries with confidence attributes
- Fire propagation: MultiPolygon geometries with temporal metadata
- Smoke dispersion: Point geometries with concentration values
- Assembly points: Point geometries with capacity and accessibility attributes

Database Storage Formats

- Impact assessment regions: PostGIS-enabled spatial tables
- Population allocation: Relational tables with foreign key constraints
- Route optimization: Graph structures in Neo4j with weighted edges
- Resource estimation: JSON documents with nested allocation structures

MinIO Object Storage

- UAV imagery: GeoTIFF with embedded spatial reference
- Simulation outputs: Timestamped GeoJSON with metadata
- Historical data: Compressed archives with version control
- Large datasets: Chunked storage with parallel access patterns

Data Synchronization Requirements

Real-Time Synchronization

- UAV data streams require immediate propagation to route optimization
- Fire propagation updates must trigger evacuation route recalculation
- Traffic optimization requires continuous integration of blocked road data

Batch Synchronization

- Population demographic updates processed during low-activity periods
- Infrastructure updates synchronized across all scenario databases
- Historical data archival with consistency verification

Event-Driven Synchronization

- Simulation completion triggers downstream processing initiation
- Threshold breaches (fire spread, smoke concentration) trigger alert workflows
- Resource capacity changes trigger reallocation algorithms

This comprehensive workflow architecture ensures that the PANTHEON platform can handle complex, multi-scenario emergency response operations while maintaining data consistency, processing efficiency, and real-time responsiveness across all supported disaster types and geographic regions.

3. DATASET SPECIFICATIONS BY SCENARIO

3.1 EARTHQUAKE SCENARIO DATASETS

The earthquake scenario utilizes a comprehensive set of input datasets covering geographic infrastructure, seismic hazard parameters, elevation data, and population demographics for the Fyli region in Attica, Greece.

INPUT DATASETS (BLOCKEDROADS, SIMULATION OUTPUTS, ALLOCATION POINTS)

Seismic Hazard and Infrastructure Data (KEMEA)

Building Infrastructure

- **Format**: Shapefiles (.shp)
- **Content**: Centroids of building blocks in Fyli
- **Source**: KEMEA (Greek Emergency Management Agency)
- **Usage**: Structural vulnerability assessment and damage estimation

Seismic Parameters

- **Format**: CSV
- **Source**: KEMEA
- **Content**: Peak ground acceleration and earthquake parameters for building blocks
- **Key Fields**:
    - OBJECTID: Unique building block identifier
    - vuln_block: Vulnerability score (0.49-0.58 range)
    - vuln_asses: Vulnerability assessment (low, moderate)
    - CentroidX, CentroidY: Building block coordinates
    - PGA (cm/s2): Peak Ground Acceleration values
    - Soil: Soil type classification (Rock=0, Stiff=1, Soft=2)
    - Mw: Earthquake magnitude
    - SAKKAS Intensity: Seismic intensity scale

Sample Data Extract:

OBJECTID	Shape	JOIN_CODE	vuln_block	vuln_asses	CentroidX	CentroidY	X_EGSA	Y_EGSA
R (km)	Soil	S (Rock=0, Stiff=1, Soft=2)	Mw	F(Fault type)	PGA (cm/s2)	SAKKAS Intensity		
(DANCIU & TSELENTIS)		Round Intensity						
1	Point Z	10	0.58	moderate	470905.8	4217068	23.66986	38.10355	6.244	Soft
	2	6.3	0	266.5650	7.6971	8				
2	Point Z	10001	0.49	low	470843.4	4217441	23.66913	38.10691	6.278	Soft
	6.3	0	266.1079	7.6945	8					

266.1079 7.6945 8

Geographic and Transportation Data

Road Networks

- **Format**: Vector data (line strings)

- **Source**: OpenStreetMap accessed via OSMNX Python package
- **Coverage**: Fyli region street networks
- **Usage**: Generate blocked roads by debris, routing calculations
- **Processing**: Coordinate extraction for road segments affected by seismic damage

General Geographic Data

- **Format**: Various OSM formats
- **Source**: OpenStreetMap
- **Coverage**: Fyli region
- **Content**: Buildings, infrastructure, points of interest
- **Usage**: Spatial context for disaster simulation

Elevation and Terrain Data

Digital Elevation Model

- **Format**: SRTM data
- **Source**: NASA SRTM altitude database
- **URL**: https://step.esa.int/auxdata/dem/SRTMGL1/
- **Resolution**: 1 arc-second (approximately 30 meters)
- **Usage**: Calculate drone altitude above mean sea level for UAV operations
- **Coverage**: Attica region including Fyli

Population and Demographics

Vulnerable Population Statistics

- **Format**: CSV
- **Source**: HumData dataset (UN/WorldPop)
- **Coverage**: Attica region
- **Content**: High-resolution population density estimates
- **Temporal Coverage**: 2020 demographic estimates

Sample Data Structure:

```
longitude,latitude,grc_general_2020
19.3913888890492387,41.9999999999763958,1.18973100185394287
19.3916666668270139,41.9999999999763958,1.0109020471572876
19.3919444446047926,41.9999999999763958,1.0109020471572876
19.3927777779381287,41.9999999999763958,1.0109020471572876
19.3930555557159039,41.9999999999763958,1.0109020471572876
```

Key Characteristics:

- **Coordinate System**: WGS84 decimal degrees
- **Resolution**: High-resolution grid (approximately 100m)

- **Population Density**: Values represent estimated population per grid cell
- **Usage**: Population exposure assessment, evacuation planning

Simulation and Emergency Response Data

Road Network Disruption

- **File**: BlockedRoads.geojson
- **Format**: GeoJSON (LineString geometries)
- **Content**: Road segments no longer accessible due to physical blockages
- **Usage**: Define transportation constraints for emergency response routing
- **Generation**: Derived from seismic impact analysis and infrastructure damage assessment

Earthquake Simulation Results

- **File**: earthquake_sim_0.geojson
- **Format**: GeoJSON (LineString/Polygon geometries)
- **Content**: Road segments and areas impacted by earthquake scenario
- **Usage**: Model inaccessible routes and damaged infrastructure zones
- **Integration**: Combines with BlockedRoads.geojson for comprehensive routing constraints

Resource Allocation Points

- **File**: allocation_point.geojson
- **Format**: GeoJSON (Point geometries)
- **Content**: Strategic locations for resource distribution (aid stations, supply centers)
- **Usage**: Destination points for emergency logistics and evacuation routing
- **Selection Criteria**: Based on accessibility, capacity, and geographic coverage

Emergency Origin Points

- **Files**: start_point_*.geojson (multiple numbered files)
- **Format**: GeoJSON (Point geometries)
- **Content**: Origin locations for routing calculations
- **Types**:
    - Affected individual locations requiring evacuation
    - Emergency vehicle dispatch points
    - Supply depot starting locations
- **Usage**: Calculate optimal routing from multiple origins to allocation points

Data Integration Workflow

These input datasets are processed through the following pipeline:

1. **Seismic hazard analysis** using KEMEA earthquake parameters and PGA values
2. **Infrastructure impact assessment** combining building vulnerability with seismic intensity

3. **Transportation network disruption** generating BlockedRoads.geojson from debris modeling
4. **Simulation execution** producing earthquake_sim_0.geojson impact zones
5. **Strategic point placement** defining allocation_point.geojson for resource distribution
6. **Origin point identification** creating start_point_*.geojson for routing scenarios
7. **Population exposure calculation** overlaying demographics with hazard zones
8. **Elevation-adjusted UAV planning** using SRTM data for drone operations

The integrated analysis generates comprehensive routing constraints and emergency response scenarios that feed into the shortest path optimization algorithms.

OUTPUT DATASETS (SHORTEST PATHS, ROUTING RESULTS)

The earthquake scenario generates optimized routing solutions that account for infrastructure damage, road blockages, and emergency response requirements. These outputs demonstrate the practical application of the secure data repository in disaster response planning.

Emergency Routing Solutions

Optimized Path Networks

- **Files**: shortest_path_*.geojson (multiple numbered files)
- **Format**: GeoJSON (LineString geometries)
- **Content**: Shortest navigable paths from each origin to allocation points
- **Generation Method**: Graph-based routing algorithms considering:
    - Blocked roads from BlockedRoads.geojson
    - Damaged infrastructure from earthquake_sim_0.geojson
    - Real-time accessibility constraints
    - Vehicle type and capacity restrictions

Key Output Characteristics:

- **Multi-origin routing**: Each file represents optimal path from corresponding start_point_*.geojson
- **Constraint integration**: Routes avoid blocked and damaged road segments
- **Distance optimization**: Minimizes travel time while ensuring route viability
- **Emergency prioritization**: Considers urgency and resource allocation efficiency

3.2 HEATWAVE SCENARIO DATASETS

The heatwave scenario focuses on Vienna, Austria, utilizing comprehensive datasets for urban heat management, vulnerable population protection, and cooling resource allocation during extreme temperature events.

INFRASTRUCTURE DATASETS (PARKS, COOLING SPOTS, FACILITIES)

Urban Green Spaces

- **File**: vienna_largest_parks.csv

- **Format**: CSV
- **Content**: Park locations, areas, and administrative district identifiers
- **Key Fields**: Park name, coordinates, area (hectares), district, capacity calculations
- **Usage**: Calculate park capacities and assign people to nearest green spaces during heat events

Cooling Infrastructure

- **File**: cooling_spots.csv
- **Format**: CSV
- **Content**: Designated cooling infrastructure locations (shaded areas, air-conditioned spaces)
- **Usage**: Primary destinations for priority populations (elderly, hospitalized individuals)
- **Selection Criteria**: Accessibility, capacity, cooling effectiveness

Educational Facilities

- **File**: schools_vienna.csv
- **Format**: CSV
- **Content**: School locations including name, coordinates, and district assignments
- **Key Fields**: School name, latitude, longitude, district, capacity
- **Usage**: Allocate student populations to the nearest cooling spots or parks

Healthcare Facilities

- **File**: hospitals_vienna.csv
- **Format**: CSV
- **Content**: Hospital locations and identifiers
- **Key Fields**: Hospital name, coordinates, patient capacity, district
- **Usage**: Allocate patient/staff populations to cooling spots or nearby parks

Care Facilities

- **File**: nursing_home_vienna.csv
- **Format**: CSV
- **Content**: Geospatial and attribute data for nursing homes
- **Key Fields**: Facility name, coordinates, resident count, vulnerability index
- **Usage**: Allocate vulnerable elderly populations to cooling spots or nearby parks

POPULATION DATASETS (DEMOGRAPHICS, VULNERABLE GROUPS)

Elderly Population Density

- **File**: elderly_dens.csv
- **Format**: CSV
- **Content**: Spatial distribution of elderly population density
- **Usage**: Simulate optimal cooling spot locations based on elderly population concentration

- **Risk Factor**: Higher vulnerability to heat-related health impacts

General Population Statistics

- **Source**: HumData dataset (UN/WorldPop)
- **Format**: CSV
- **Coverage**: Vienna region (Austria)
- **Sample Data Structure**:

```
longitude,latitude,aut_general_2020
9.000000000015199042,48.289999999981454,2.59261798858642578
9.00027777792976735,48.289999999981454,2.59261798858642578
9.00055555570754606,48.289999999981454,2.59261798858642578
9.00083333348532477,48.289999999981454,2.59261798858642578
9.00333333348532605,48.289999999981454,2.59261798858642578
9.00388888904088169,48.289999999981454,5.18523597717285156
```

- **Key Characteristics**:
  - High-resolution population density grid
  - 2020 demographic estimates
  - WGS84 coordinate system

Population Distribution by Institution

- **Format**: Python dictionaries/JSON
- **Content**:
  - people_per_district: Total general population by administrative district
  - people_per_school: Student populations assigned to educational institutions
  - people_per_hospital: Patient and staff populations in healthcare facilities
  - people_per_nursing_home: Resident populations in care facilities
- **Usage**: Population allocation calculations for heat emergency response

Solar Radiation Data

- **Source**: Google Solar API (data-layers API)
- **Format**: Monthly Flux GeoTIFF (processed to CSV)
- **Content**: Solar radiation values for specific Vienna locations
- **Sample Data Structure**:

```
Latitude,Longitude,SolarRadiation
 48.271829, 16.405452, 35.000000
 48.271829, 16.405458, 37.000000
 48.271829, 16.405465, 37.000000
 48.271828, 16.405472, 36.000000
```

```
48.271828, 16.405479, 33.000000
```

- **Units**: Solar radiation in kWh/m²/day
- **Temporal Resolution**: Monthly data
- **Usage**: Heat exposure assessment and cooling demand modeling

ALLOCATION AND RESULTS DATASETS

Spatial Analysis Support

- **File**: precomputed_distances.csv
- **Format**: CSV
- **Content**: Pairwise distance matrix between Vienna districts and parks
- **Usage**: Support reallocation when local park capacity is insufficient
- **Optimization**: Pre-calculated for computational efficiency during emergency response

District Geographic Centers

- **File**: district_centers.csv
- **Format**: CSV
- **Content**: Latitude and longitude of each district's geographic center
- **Usage**: Represent centroid for general district-level population in nearest park calculations
- **Coordinate System**: WGS84 decimal degrees

Final Allocation Results

- **File**: combined_df (allocation results table)
- **Type**: Output
- **Format**: DataFrame/CSV
- **Content**: Comprehensive population allocation summary
- **Key Fields**:
    - Allocated people from schools, hospitals, nursing homes
    - General population assignments
    - Distances traveled to cooling locations
    - Remaining park capacities after allocation
    - Intra-district vs. inter-district movement indicators
    - Heat vulnerability risk scores

Geographic Base Data

- **Source**: OpenStreetMap Vienna
- **Format**: Various OSM formats
- **Content**: Urban infrastructure, buildings, transportation networks

- **Usage**: Spatial context for heat island analysis and accessibility modeling

Integration Workflow

The heatwave scenario processes these datasets through:

1. **Heat exposure analysis** using solar radiation and urban heat island data
2. **Vulnerable population identification** combining elderly density with institutional populations
3. **Cooling resource mapping** from parks and designated cooling spots
4. **Capacity-based allocation** using precomputed distances and facility capacities
5. **Optimization algorithms** generating final allocation results
6. **Emergency response planning** with intra/inter-district movement coordination

3.3 WILDFIRE SCENARIO DATASETS

The wildfire scenario targets the Attica region of Greece, particularly around Fyli, incorporating comprehensive environmental, meteorological, and topographic datasets essential for wildfire spread modelling and emergency response planning.

ENVIRONMENTAL DATASETS (WEATHER, ELEVATION, FUEL MAPS)

Historical Weather Data

- **Source**: ERA5 (ECMWF Reanalysis v5)
- **Format**: NetCDF
- **URL**: https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5
- **Coverage**: Attica region, Greece
- **Content**: Historical meteorological data including:
  - Wind direction and speed
  - Temperature profiles
  - Humidity levels
  - Precipitation data
  - Atmospheric pressure
- **Temporal Resolution**: Hourly data
- **Usage**: Wildfire spread prediction, wind-driven fire behavior modeling

Digital Elevation Model

- **Source**: Copernicus DEM (Digital Elevation Model)
- **Format**: TIFF (GeoTIFF)
- **URL**: https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM
- **Coverage**: Attica region
- **Resolution**: 30-meter spatial resolution

- **Content**: Terrain elevation data
- **Usage**:
    - Slope analysis for fire spread modeling
    - Terrain-influenced wind pattern analysis
    - Evacuation route planning considering topography
    - UAV flight path optimization

Fuel Load Mapping

- **Source**: ArcFuel Project
- **Format**: TIFF (GeoTIFF)
- **URL**: https://fmrs.web.auth.gr/project/arcfuel/
- **Coverage**: West Attica region
- **Content**: Vegetation fuel maps including:
    - Fuel type classification
    - Fuel load density
    - Vegetation moisture content indicators
    - Fire behavior fuel models
- **Usage**: Fire spread rate calculations, burn intensity predictions

NASA Elevation Data

- **Source**: NASA SRTM (Shuttle Radar Topography Mission)
- **Format**: Digital elevation data
- **URL**: https://step.esa.int/auxdata/dem/SRTMGL1/
- **Resolution**: 1 arc-second (approximately 30 meters)
- **Usage**: Calculate drone altitude above mean sea level for UAV fire monitoring operations

GEOGRAPHIC DATASETS (OSM, BOUNDARIES)

OpenStreetMap Infrastructure

- **Source**: OpenStreetMap
- **Coverage**: Fyli region, Attica
- **Format**: Various OSM formats (XML, PBF)
- **Content**:
    - Road networks and transportation infrastructure
    - Buildings and structures
    - Land use classifications
    - Water bodies and natural features
    -

- **Usage**:
    - Evacuation route planning
    - Infrastructure vulnerability assessment
    - Fire barrier identification

Road Networks for Emergency Response

- **Source**: OpenStreetMap via OSMNX Python package
- **Format**: Vector data (line strings)
- **Content**: Detailed street network coordinates
- **Usage**:
    - Generate blocked roads due to fire damage
    - Emergency vehicle routing
    - Evacuation path optimization
    - Accessibility analysis during fire events

3.4 POPULATION AND VULNERABILITY DATA

Population Demographics

- **Source**: HumData dataset (UN/WorldPop)
- **Format**: CSV
- **Coverage**: Attica region, Greece
- **Temporal Coverage**: 2020 estimates

Sample Data Structure:

```
longitude,latitude,aut_general_2020
9.000000000015199042,48.289999999981454,2.59261798858642578
9.00027777792976735,48.289999999981454,2.59261798858642578
9.00055555570754606,48.289999999981454,2.59261798858642578
9.00083333348532477,48.289999999981454,2.59261798858642578
9.00333333348532605,48.289999999981454,2.59261798858642578
9.00388888904088169,48.289999999981454,5.18523597717285156
```

Key Characteristics:

- **Coordinate System**: WGS84 decimal degrees
- **Resolution**: High-resolution population density grid (~100m)
- **Population Density**: Values represent estimated population per grid cell
- **Usage**:
    - Population at risk assessment
    - Evacuation demand modeling

- Resource allocation for firefighting operations
- Vulnerable population identification in fire-prone areas

Wildfire Modeling Integration

Fire Spread Simulation Workflow:

1. **Meteorological Analysis**: Process ERA5 weather data for wind conditions
2. **Terrain Analysis**: Analyze elevation and slope from DEM data
3. **Fuel Assessment**: Integrate ArcFuel vegetation maps for combustibility
4. **Population Exposure**: Overlay demographic data with fire risk zones
5. **Infrastructure Impact**: Assess road networks and building vulnerability
6. **Emergency Response**: Generate evacuation routes and resource deployment plans

Multi-Layer Risk Assessment:

- **Weather-driven risk**: High wind speeds, low humidity conditions
- **Topographic risk**: Steep slopes, fire-prone elevation zones
- **Fuel-based risk**: High-density vegetation areas, dry fuel conditions
- **Population risk**: Residential areas within fire simulation perimeters
- **Infrastructure risk**: Critical facilities and transportation networks

The wildfire datasets enable comprehensive fire behavior modeling, real-time risk assessment, and evidence-based emergency response planning for the Attica region's fire-prone landscape.

3.5 CYBERATTACK / EXPLOSION SCENARIO DATASETS

The cyberattack and explosion scenarios focus on Vienna, Austria, incorporating urban infrastructure data, weather conditions, and population demographics to model infrastructure disruption, physical damage, and emergency response in urban environments.

INFRASTRUCTURE DATASETS (BUILDINGS, NETWORKS)

Urban Building Infrastructure

- **Source**: OSM Buildings Data
- **Format**: Vector data (GeoJSON/OSM)
- **URL**: https://osmbuildings.org
- **Coverage**: Vienna metropolitan area
- **Content**:
  - Building footprints and geometries
  - Building heights and structure types
  - Residential, commercial, and industrial classifications
  - Critical infrastructure identification

- **Usage**:
    - Blast damage assessment modeling
    - Infrastructure vulnerability analysis
    - Debris field calculations
    - Evacuation shelter identification

Transportation Networks

- **Source**: OpenStreetMap via OSMNX Python package
- **Format**: Vector data (line strings)
- **Coverage**: Vienna road networks
- **Content**: Street network coordinates and road classifications
- **Usage**:
    - Generate blocked roads from explosion debris
    - Emergency vehicle routing during infrastructure failure
    - Evacuation route optimization
    - Network resilience analysis during cyberattacks

General Vienna Infrastructure

- **Source**: OpenStreetMap
- **Format**: Various OSM formats
- **Coverage**: Vienna urban area
- **Content**:
    - Transportation infrastructure
    - Utilities and services
    - Public facilities
    - Communication networks
- **Usage**: Infrastructure interdependency analysis during cyber incidents

Elevation Data for Drone Operations

- **Source**: NASA SRTM altitude database
- **Format**: Digital elevation data
- **URL**: https://step.esa.int/auxdata/dem/SRTMGL1/
- **Resolution**: 1 arc-second spatial resolution
- **Usage**: Calculate drone altitude above mean sea level for:
    - Damage assessment flights
    - Emergency surveillance operations
    - Communication relay deployment

WEATHER AND ENVIRONMENTAL DATA

Real-Time Weather Data

- **Source**: OpenWeatherMap
- **Format**: JSON/API responses
- **URL**: https://openweathermap.org
- **Coverage**: Vienna region
- **Content**:
  - Current weather conditions
  - Wind speed and direction
  - Temperature and humidity
  - Precipitation data
  - Visibility conditions
- **Update Frequency**: Real-time/hourly
- **Usage**:
  - Dispersion modeling for chemical/radiological releases
  - Weather impact on emergency operations
  - Drone flight condition assessment

Specialized Weather Data (FINT)

- **Source**: FINT Weather Data (to be confirmed)
- **Format**: TBC (To Be Confirmed)
- **Coverage**: Vienna area
- **Content**: Specialized meteorological parameters
- **Usage**: Enhanced weather modeling for emergency response scenarios

3.6 POPULATION AND DEMOGRAPHIC DATA

Vienna Population Distribution

- **Source**: HumData dataset (UN/WorldPop)
- **Format**: CSV
- **Coverage**: Vienna region, Austria
- **Temporal Coverage**: 2020 demographic estimates

Sample Data Structure:

```
longitude,latitude,aut_general_2020
9.00000000015199042,48.289999999981454,2.59261798858642578
9.00027777792976735,48.289999999981454,2.59261798858642578
```

```
9.00055555570754606,48.289999999981454,2.59261798858642578
9.00083333348532477,48.289999999981454,2.59261798858642578
9.00333333348532605,48.289999999981454,2.59261798858642578
9.00388888904088169,48.289999999981454,5.18523597717285156
```

Key Characteristics:

- **Coordinate System**: WGS84 decimal degrees
- **Resolution**: High-resolution population density grid
- **Population Density**: Values represent estimated population per grid cell
- **Usage**:
    - Casualty estimation for explosion scenarios
    - Evacuation demand modeling
    - Emergency service resource allocation
    - Vulnerable population identification in affected areas

Scenario-Specific Applications

Cyberattack Scenario:

- **Infrastructure Dependency Mapping**: Identify critical systems vulnerable to cyber disruption
- **Cascading Failure Analysis**: Model how cyberattacks on key infrastructure affect urban systems
- **Emergency Communication**: Alternative communication networks during system failures
- **Population Impact Assessment**: Areas affected by infrastructure service disruption

Explosion Scenario:

- **Blast Radius Modeling**: Calculate damage zones based on building data and population density
- **Debris Field Analysis**: Predict road blockages and infrastructure damage
- **Casualty Estimation**: Population-based injury and fatality projections
- **Emergency Response Routing**: Optimal paths for first responders considering damaged infrastructure

Integration with Emergency Response

Multi-Hazard Analysis Workflow:

1. **Infrastructure Vulnerability Assessment**: Analyze building and network data for critical points
2. **Weather Impact Evaluation**: Incorporate real-time weather for dispersion/visibility modeling
3. **Population Exposure Calculation**: Overlay demographic data with incident impact zones
4. **Emergency Response Optimization**: Generate response plans considering infrastructure damage
5. **Real-Time Monitoring**: Continuous data updates for dynamic situation assessment

Cross-Scenario Compatibility:

- **Shared Infrastructure Data**: Building and network data applicable to both cyber and explosion scenarios

- **Common Population Base**: Vienna demographic data used across scenario types
- **Weather Integration**: Meteorological data supports both dispersion modeling and operational planning
- **Transportation Networks**: Road data essential for both evacuation and emergency response routing

The cyberattack and explosion datasets provide comprehensive urban infrastructure modeling capabilities, enabling realistic simulation of modern threat scenarios in dense urban environments like Vienna.

3.7 SHARED / CROSS-SCENARIO DATASETS

The PANTHEON platform utilizes several foundational datasets that provide common infrastructure, geographic, and operational data across multiple disaster scenarios. These shared datasets ensure consistency, reduce data redundancy, and enable cross-scenario analysis and planning.

BASE GEOGRAPHIC DATA (OSM, SATELLITE IMAGERY)

OpenStreetMap Infrastructure Data

- **Source**: OpenStreetMap / OpenInfraMap
- **Format**: GeoJSON
- **URL**: https://openinframap.org/
- **Coverage**: Vienna (Austria) and Athens/Attica region (Greece)
- **Content**:
  - Power infrastructure (transmission lines, substations, power plants)
  - Telecommunications infrastructure (cell towers, fiber networks)
  - Transportation networks (roads, railways, public transit)
  - Utility networks (water, gas pipelines)
- **Usage Across Scenarios**:
  - **Earthquake**: Infrastructure vulnerability and damage assessment
  - **Heatwave**: Power grid stress analysis during high cooling demand
  - **Wildfire**: Critical infrastructure protection prioritization
  - **Cyberattack**: Cyber-physical infrastructure interdependency mapping

OpenStreetMap Road Networks

- **Source**: OpenStreetMap via OSMNX Python package
- **Format**: Vector data (line strings)
- **Coverage**: Both Vienna and Athens/Fyli regions
- **Content**: Detailed street network geometries and road classifications
- **Cross-Scenario Applications**:
  - **Earthquake & Cyberattack**: Generate blocked roads from debris/infrastructure failure
  - **All Scenarios**: Emergency vehicle routing and evacuation path planning

- **Wildfire**: Fire service access routes and evacuation corridors
- **Heatwave**: Access routes to cooling centers and parks

Satellite Imagery Base Maps

- **Source**: Copernicus Sentinel-2 L2A
- **Format**: Raster (True color map tiles)
- **URL**: https://sentiwiki.copernicus.eu/web/s2-products#S2Products-Level-2AProductsS2-Products-L2Atrue
- **Coverage**: Both study regions (Vienna and Attica)
- **Resolution**: 10-meter multispectral imagery
- **Content**: High-resolution true color satellite imagery
- **Usage**:
  - Base layer for all scenario visualizations
  - Change detection and damage assessment
  - Situational awareness and geographic context
  - Training simulation backgrounds

Elevation Data (NASA SRTM)

- **Source**: NASA SRTM altitude database
- **Format**: Digital elevation model
- **URL**: https://step.esa.int/auxdata/dem/SRTMGL1/
- **Coverage**: Global (including both Vienna and Attica regions)
- **Resolution**: 1 arc-second (approximately 30 meters)
- **Cross-Scenario Applications**:
  - **Earthquake & Cyberattack**: Drone altitude calculations for damage assessment
  - **Wildfire**: Terrain analysis for fire spread modeling
  - **All Scenarios**: UAV flight planning and 3D visualization

INFRASTRUCTURE AND POI DATASETS

Points of Interest (Critical Facilities)

- **Sources**: OSM and Google Maps Places API
- **Format**: GeoJSON
- **Coverage**: Vienna and Athens regions
- **Content**:
  - **Schools**: Educational facilities with capacity and location data
  - **Police Stations**: Law enforcement facilities and jurisdiction areas
  - **Fire Stations**: Fire service locations and response coverage areas

    - **Hospitals**: Healthcare facilities with capacity and specialization data
- **Metadata Fields**: Name, address, coordinates, capacity, operating hours, contact information
- **Cross-Scenario Relevance**:
    - **Emergency Response**: First responder dispatch and resource allocation
    - **Evacuation Planning**: Shelter locations and assembly points
    - **Capacity Analysis**: Available resources during different disaster types
    - **Vulnerability Assessment**: Critical infrastructure protection priorities

Transportation Infrastructure

- **Source**: Google Roads API
- **Format**: GeoJSON
- **URL**: https://developers.google.com/maps/documentation/roads
- **Coverage**: Both Vienna and Athens road networks
- **Content**: Snap-to-nearest roads functionality data
- **Features**:
    - Precise road geometry alignment
    - Speed limit and traffic data integration
    - Route optimization parameters
- **Usage**:
    - Enhanced routing accuracy for emergency vehicles
    - Traffic flow modeling during evacuations
    - Real-time navigation support for field operations

POPULATION AND DEMOGRAPHIC DATASETS

HumData Population Statistics

- **Source**: HumData (UN/WorldPop collaboration)
- **Format**: CSV (high-resolution grids)
- **Coverage**:
    - Austria (Vienna region): aut_general_2020
    - Greece (Attica region): grc_general_2020
- **Temporal Coverage**: 2020 demographic estimates
- **Resolution**: Approximately 100-meter grid cells

Austria Population Sample:

```
longitude,latitude,aut_general_2020
9.00000000015199042,48.289999999981454,2.59261798858642578
9.00027777792976735,48.289999999981454,2.59261798858642578
```

```
9.00055555570754606,48.289999999981454,2.59261798858642578
9.00083333348532477,48.289999999981454,2.59261798858642578
9.00333333348532605,48.289999999981454,2.59261798858642578
9.00388888904088169,48.289999999981454,5.18523597717285156
```

Greece Population Sample:

```
longitude,latitude,grc_general_2020
19.3913888890492387,41.9999999999763958,1.18973100185394287
19.3916666668270139,41.9999999999763958,1.0109020471572876
19.3919444446047926,41.9999999999763958,1.0109020471572876
19.3927777779381287,41.9999999999763958,1.0109020471572876
19.3930555557159039,41.9999999999763958,1.0109020471572876
```

Cross-Scenario Applications:

- **Population Exposure Analysis**: Consistent demographic baseline across all disaster types
- **Evacuation Demand Modeling**: Population-based resource requirement calculations
- **Vulnerability Assessment**: Identify high-density areas requiring priority attention
- **Impact Assessment**: Standardized population impact metrics across scenarios

Data Integration and Standardization

Coordinate Reference Systems:

- **Standard**: WGS84 (EPSG:4326) for all geographic datasets
- **Consistency**: Ensures seamless integration across data sources and scenarios
- **Compatibility**: Supports web mapping and mobile applications

Data Quality Standards:

- **Temporal Consistency**: Regular updates from authoritative sources
- **Spatial Accuracy**: Validated against ground truth and official sources
- **Metadata Completeness**: Comprehensive documentation for all shared datasets
- **Version Control**: Synchronized updates across all scenario implementations

Cross-Scenario Benefits:

- **Reduced Storage**: Single instances of common datasets across multiple scenarios
- **Consistent Analysis**: Standardized geographic and demographic foundations
- **Comparative Studies**: Enable analysis across different disaster types using common baselines
- **Operational Efficiency**: Shared data preparation and validation processes
- **Platform Integration**: Common data formats support unified PANTHEON platform architecture

Integration Architecture:

- **MinIO Object Storage**: Centralized repository for large raster datasets (satellite imagery, elevation models)

- **PostgreSQL/PostGIS**: Structured storage for vector data and attribute tables
- **Neo4j**: Graph relationships between infrastructure elements across scenarios
- **Kafka**: Real-time data streaming for dynamic updates to shared datasets

The shared dataset collection provides a robust, standardized foundation that enables the PANTHEON platform to deliver consistent, high-quality disaster simulation and emergency response capabilities across diverse threat scenarios and geographic regions.

3.8 PANTHEON DATA SCHEMAS AND METADATA – ALL SCENARIOS

The PANTHEON platform implements standardized data schemas across all four emergency scenarios (earthquake, heatwave, wildfire, cyberattack), enabling seamless integration through unified Kafka messaging, MinIO storage, and PostgreSQL databases while supporting scenario-specific data requirements.

Multi-Scenario Overview

PANTHEON processes emergency data across four major hazard types with real datasets and operational components:

Table 3. Unified Kafka Topic Architecture

Scenario	Location	Real Datasets	Components	Processing Pipeline
Earthquake	Attica, Greece	GeoJSON blocked roads	6 components	UI → Impact → UAV → Routes → SelfAdapt → Population
Heatwave	Vienna, Austria	1,023 records (10 CSV files)	3 components	UI → Impact → Cooling DSS
Wildfire	Athens, Greece	Fire propagation GeoJSON	6 components	UI → Fire Sim → Routes → UAV → SelfAdapt → Assembly
Cyberattack	Vienna, Austria	Explosion + GeoJSON outputs	6 components	UI → Explosion → Routes → Gaussian Plume → SelfAdapt → UAV

Cross-Scenario Topic Naming Convention

Pattern: ds-{city}-{scenario_letter}-{component_type}

Examples:

- ds-ath-a-impact-assessment-poi (Athens - scenario A - earthquake)
- ds-vie-a-impact-assessment-region (Vienna - scenario A - heatwave)
- ds-ath-b-fire-propagations (Athens - scenario B - wildfire)
- ds-vie-b-impact-assessment-poi (Vienna - scenario B - cyberattack)

Scenario-Specific Schemas and Real Data

1. Earthquake Scenario (Attica, Greece)

Real Data Evidence: GeoJSON files with MultiLineString geometries

- earthquake_BlockedRoads_2025-05-21_0.geojson
- Geographic bounds: 23.6°-23.7°E, 38.1°-38.2°N
- 62 coordinate segments representing blocked road networks

Key Topics:

- ds-ath-a-impact-assessment-poi (< 1s processing)
- ds-ath-a-blocked-roads (O(<10s) processing)
- ds-ath-a-weighted-routes
- ds-ath-a-apsr-estimation

Blocked Roads Schema (Real Data Structure):


```
{
 "type": "FeatureCollection",
 "features": [{
  "type": "Feature",
  "geometry": {
   "type": "MultiLineString",
   "coordinates": [
    [[23.669002300016587, 38.102929299983025], [23.669038415692544, 38.102930761882874]],
    [[23.669002004790524, 38.1029082077779], [23.669002300016587, 38.102929299983025]]
   ]
  },
  "properties": {
   "type": "BlockedRoads",
   "detection_method": "uav_visual|simulation",
   "confidence": "number (0-1)",
   "severity": "partial|complete"
  }
 }]
}
```


2. Heatwave Scenario (Vienna, Austria)

Real Data Evidence: 1,023 records across 10 CSV datasets

- 139 nursing homes with coordinates

- 289 parks with tree coverage data
- 230 high-resolution elderly population points
- 29 hospitals + 85 schools + 23 district centers

Key Topics:

- ds-vie-a-impact-assessment-region (2-3 min processing)
- ds-vie-a-csr-estimation-dss (O(1min) processing)

Impact Assessment Schema (Based on Real Vienna Data):

```
{
  "sim_id": "string",
  "assessment_results": [{
    "id": "integer",
    "type": "node|district|facility",
    "coordinates": "(longitude,latitude)",
    "likelihood": "number (0-5)",
    "impact": "number (0-10)",
    "risk": "number (0-1)",
    "facility_type": "nursing_home|hospital|school|park|district_center",
    "vulnerability_factors": {
      "elderly_population": "integer",
      "children_population": "integer",
      "tree_coverage": "integer",
      "cooling_capacity": "boolean"
    }
  }],
  "district_summaries": [{
    "district_id": "integer (1-23)",
    "total_population": "integer",
    "elderly_population": "number",
    "heat_risk_score": "number (0-1)"
  }]
}
```

3. Wildfire Scenario (Athens, Greece)

Real Data Evidence: Fire propagation simulation files

- MinIO bucket: ds-ath-b-fire-propagations
- GeoJSON temperature and fuel boundary data
- 15-minute interval fire progression files

Key Topics:

- ds-ath-b-fire-propagations (< 10s processing)
- ds-ath-b/routes
- ds-ath-b/uav (< 20s processing)
- ds-ath-b/traffic
- ds-ath-b-dss

Fire Propagation Schema:


```
{
 "type": "FeatureCollection",
 "name": "temperature",
 "crs": {"type": "name", "properties": {"name": "urn:ogc:def:crs:OGC:1.3:CRS84"}},
 "features": [{
  "type": "Feature",
  "properties": {},
  "geometry": {
   "type": "MultiPolygon",
   "coordinates": [
    [[[23.699387434465272, 38.112030721668653],
      [23.699415952036244, 38.112030794624054]]]
   ]
  }
 }],
 "fire_statistics": {
  "burned_area_hectares": "number",
  "fire_perimeter_km": "number",
  "progression_rate_mh": "number",
  "fuel_type": "string",
  "weather_conditions": "object"
 }
```

}

4. Cyberattack Scenario (Vienna, Austria)

Real Data Evidence: Explosion simulation + blocked roads

- cyberattack_BlockedRoads_2025-05-21_1.geojson - Infrastructure damage mapping
- Gaussian Plume GeoJSON - Real concentration dispersion modeling
- Geographic bounds: 16.37°-16.41°E, 48.227°-48.232°N
- 45+ coordinate segments for blocked roads + high-resolution plume grid**Key Topics**:

Key Topics:

- ds-vie-b-impact-assessment-poi (O(30s) processing)
- ds-vie-b-smoke-propagations (O(1s) processing)

Gaussian Plume Schema (Real Vienna Data):{

{
 "type": "FeatureCollection",
 "features": [{
  "type": "Feature",
  "geometry": {
   "type": "Point",
   "coordinates": [16.370959285651086, 48.22734783579403]
  },
  "properties": {
   "concentration": 1.047807885200388
  }
 }],
 "plume_metadata": {
  "source_location": {
   "latitude": "number",
   "longitude": "number",
   "height": "number (meters)"
  },
  "grid_resolution": "~4.5m spacing",
  "area_coverage": "4.1km x 0.5km",
  "concentration_units": "scientific_units",
  "wind_conditions": {
   "direction": "number (degrees)",
   "speed": "number (m/s)"

```
  },
  "calculation_time": "O(1s)",
  "total_points": "1000+ grid points"
 }
}
```


Unified MinIO Storage Architecture

Cross-Scenario Bucket Organization

PANTHEON MinIO Buckets:

```
├── ds-ath-a-impact-assessment-poi/     # Earthquake (Athens)
├── ds-ath-a-blocked-roads/
├── ds-ath-a-weighted-routes/
├── ds-ath-a-apsr-estimation/
├── ds-vie-a-impact-assessment-region/  # Heatwave (Vienna)
├── ds-vie-a-csr-estimation-dss/
├── ds-ath-b-fire-propagations/         # Wildfire (Athens)
├── ds-ath-b-routes/
├── ds-ath-b-uav/
├── ds-ath-b-traffic/
├── ds-ath-b-dss/
├── ds-vie-b-impact-assessment-poi/     # Cyberattack (Vienna)
└── ds-vie-b-smoke-propagations/
```

Universal File Naming Convention

Pattern: {scenario}_{datatype}_{YYYY-MM-DD}_{HH-MM-SS.microseconds}_{sequence}.{ext}

Real Examples:

- earthquake_BlockedRoads_2025-05-21_0.geojson
- cyberattack_BlockedRoads_2025-05-21_1.geojson
- cyberattack_Outputs_2025-05-07_12-39-05.021392_1.geojson

Common Object Metadata Schema


```
{
 "simulation-id": "string",
 "scenario-type": "earthquake|heatwave|wildfire|cyberattack",
 "city": "athens|vienna",
```

"component": "string",
"geographic-bounds": {
"north": "number", "south": "number",
"east": "number", "west": "number"
},
"processing-time-seconds": "number",
"confidence-level": "high|medium|low",
"timestamp": "ISO 8601 datetime",
"file-size-bytes": "number",
"coordinate-count": "number"
}

Table 4. Processing Performance (Real Measurements)

Component	Response Time	Scenario
Impact Simulation	< 1s	Earthquake
UAV Swarms	O(<10s)	Earthquake/Wildfire
Impact Assessment	2-3 min	Heatwave
Cooling DSS	O(1min)	Heatwave
Fire Propagation	< 10s	Wildfire
Explosion Simulation	O(30s)	Cyberattack
Gaussian Plume	O(1s)	Cyberattack

This unified schema framework demonstrates PANTHEON's comprehensive emergency response data infrastructure across all four hazard scenarios, integrating real datasets from Athens and Vienna with standardized processing pipelines, storage systems, and API interfaces for complete multi-hazard emergency management capability.

4. DATA REPOSITORY & SECURITY FRAMEWORK

4.1 INFRASTRUCTURE OVERVIEW

The PANTHEON platform is architected as a **layered, modular, and secure digital infrastructure** designed to support disaster risk management (DRM) through real-time data integration, simulation, AI-assisted decision support, and federated access. The system accommodates multiple stakeholders, scalable data sources, and AI/ML model orchestration while ensuring strict security and interoperability.

LAYERED ARCHITECTURE

The platform is organized into **five logical layers**, each serving a distinct operational role:

L1 – Data/Persistence Layer

- This foundational layer provides robust, secure storage for raw, simulated, and metadata-rich datasets.
- **Secure Data Store (T7.3-D7.3)**: Encrypted storage for sensitive IoT data, UAV recordings, and simulated outputs.
- **Metadata Repository**: Descriptive information about models and datasets to support discoverability and validation.
- **Index Service**: References to external and internal data sources, ensuring semantic linkage and federated queries.
- **Data Connector (T7.2-D7.1)**: Interfaces with external systems (e.g., IoT data providers, third-party DRM operators), translating formats and routing inputs via APIs or secure gateways.

L2 – Models Layer

This layer manages the lifecycle of computational models essential for DRM simulations and predictions.

- **Model Management (T4.2, T4.3, T4.4)**: Tools for deploying, updating, and version-controlling AI, ML, and simulation models.
    - **Model Categories**:
    - **ML Models (T4.3)**
    - **Simulation Models (T4.4)**
    - **AI and Synthetic Models**
    - **Conceptual Models**
- **Kafka Integration**: Enables real-time model input/output streaming to downstream services.

L3 – Backend Services Layer

Acts as the central processing layer coordinating decision support and analytics workflows.

- **Service Orchestrator (T7.1)**: Coordinates backend logic, model execution, and pipeline triggering.
- **DSS Engine (T5.2)**: Provides reasoning and decision-support logic for DRM operators.
- **AI-Based Engine**: Integrates cognitive and learning models for adaptive recommendations.
- **Simulations Connectors (T4.3)**: Interfaces connecting models to real-time or scenario data.
- **Data Analysis (T4.2)**: Performs statistical, spatial, and risk assessments on input/output data.
- **Kafka Bus**: High-throughput event/message exchange between services and layers.

L4 – Analysis and Optimization Layer

This layer encapsulates **high-level reasoning and decision tools** tailored for crisis management actors.

- **DSS for CBDRM Operators (T5.5)**: End-user dashboard for community-based disaster risk management, enriched with visual analytics and optimization modules.
- **Kafka**: Real-time channel for delivering simulation insights and model outputs to operational interfaces.

L5 – Frontend Layer

Provides the user-facing tools for decision-making, training, and system management.

- **Virtual Representation & Visual Data Analysis (T4.5)**: Interactive dashboards, 2D/3D maps, scenario comparisons.
- **System Monitoring and Admin Tools**: Interfaces for platform status, data health, and user management.
- **REST API & External Interfaces (T4.5)**: Provides programmatic access to data, simulations, and results.
- **Training Tool**: Educational interface for simulating responses and running scenario drills.
- **Security Integration (T7.4-D7.3)**: Access control, authentication, and session management enforced across layers via Keycloak.

Input Data Sources (right-hand side):

- **IoT Data Streams**: Satellite, infrastructure, traffic, UAV, and community data are ingested via the Data Connector.
- **Third-Party DRM Operators**: Federated access to external systems and datasets via API integration or secure exchange protocols.

Data Routing and Enrichment:

- Data flows upward from ingestion (L1) → modeling (L2) → backend reasoning (L3) → operator-facing DSS (L4) → visual frontends (L5).
- Bi-directional flow supports real-time feedback loops and adaptive decision-making.

SECURITY, IDENTITY, AND ACCESS CONTROL

Security Layer (T7.4–D7.3) cuts vertically across all components:

- **AAA**: Authentication (Keycloak), Authorization (role- and token-based), and Accounting (logging and auditing).
- **Encryption**: TLS/SSL in transit, AES-based encryption at rest.
- **Frontend + API Protection**: Token-based OAuth2/OIDC protocols.
- **Kafka & Data Store Protection**: Role-based access and secure channel policies.

HOSTING AND DEPLOYMENT

The architecture supports **cloud-native and on-premise deployment**, with containerization (e.g., Docker), orchestration (e.g., Kubernetes), and service meshes.

Hosting Platform (T7.3-D7.2) ensures system scalability, performance isolation, and modular deployment.

SCALABILITY AND INTEROPERABILITY

- **Kafka-based streaming** enables horizontal scaling for data processing and analytics.
- **REST APIs and Open Standards** (OGC, JSON, GeoJSON) ensure interoperability with national/international DRM tools.
- **Model Plug-and-Play**: The architecture supports model substitution and update without disrupting other services.

4.2 STORAGE SYSTEM

The PANTHEON platform utilizes a hybrid storage strategy that combines file-based storage, relational databases, object storage, and graph databases. This architecture supports scalability, performance optimization, data typing flexibility, and federated data access for diverse disaster response scenarios.

FILE SYSTEM-BASED STORAGE

Purpose

- Low-level storage for local logs, configuration files, simulation binaries, static datasets, and temporary cache data.
- Often used during development, simulation execution, or when data cannot yet be processed for higher-tier systems.

Structure

- Directory tree organized by scenario, data type, and timestamp.
- Example:

```
/data/
├── earthquake/
│   ├── raw/
│   └── outputs/
├── wildfire/
└── temp_cache/
```

Security & Redundancy

- Filesystem-level encryption (e.g., **LUKS**, **eCryptfs**).
- Backed up periodically and optionally mounted on encrypted volumes.

Usage

- Staging area before ingestion into MinIO or PostgreSQL.
- Suitable for lightweight jobs and single-node data processing.

POSTGRESQL/POSTGIS (RELATIONAL & SPATIAL DATABASE)

Purpose

Stores **structured data**, including:

- Tabular datasets (e.g., demographics, infrastructure assets)
- Geospatial data (points, lines, polygons) using **PostGIS**
- Metadata and simulation parameters
- Application configurations and logs

Data Types

- SQL tables with typed schemas
- PostGIS geometries (e.g., `GEOMETRY(Point, 4326)`)
- JSONB columns for hybrid structured/semi-structured data

Security

- TLS/SSL for connections
- Role-based access (`GRANT`, `REVOKE`, and `RLS`)
- Sensitive fields encrypted using **pgcrypto** (AES-256)

Spatial Capabilities

- Fast spatial indexing using R-trees (GIST)
- Supports spatial joins, buffering, distance queries

Use Cases

- Routing calculations, infrastructure queries, risk zoning
- Storing simulation metadata and preprocessed population layers

MINIO (OBJECT STORAGE SYSTEM)

Purpose

Stores **large unstructured and semi-structured files**:

- Satellite imagery (GeoTIFF, JPEG2000)
- UAV footage (MP4, TIFF)
- Simulation results (GeoJSON, NetCDF, CSV)
- Fire propagation maps and elevation models

Storage Model

- S3-compatible object buckets
- Object metadata stored alongside files (e.g., timestamp, coordinates, simulation ID)

Security

- AES-256 **server-side encryption** (SSE-S3 or SSE-KMS)
- TLS for all connections
- Bucket policies + JWT-based access control via **Keycloak**

Advantages

- Scalable horizontally (multi-node clusters)
- High throughput for large file read/write
- Versioning and replication support

Use Cases

- Storing fire simulation GeoJSONs and video feeds
- Making large datasets available to APIs and AI models on-demand
- Archiving historical simulation runs

NEO4J (GRAPH DATABASE)

Purpose

Stores **interconnected datasets**:

- Infrastructure interdependencies (power ↔ telecom ↔ water)
- Network topologies (transport, sensor graphs)
- Knowledge graphs for scenario logic or AI inference

Data Model

- Nodes and relationships:
  - Nodes: Hospitals, Roads, Schools, etc.
  - Relationships: CONNECTED_TO, DEPENDS_ON, LOCATED_IN
- Supports complex traversals and impact propagation queries

Security

- TLS-encrypted bolt connections
- Fine-grained access to graph elements via custom roles

Use Cases

- Tracing cascading failures (e.g., power outage → hospital shutdown)
- Visualizing dependencies between emergency resources
- Enhancing AI models with graph features

COORDINATED STORAGE ARCHITECTURE

The hybrid approach allows for optimal data access across use cases:

Table 5. Hybrid approach to store and access data

Component	Best For	Accessed By	Indexed?	Scalable?
File System	Configs, small files	Sim tools, batch jobs	No	Limited
PostgreSQL	Structured, spatial data	APIs, dashboards	Yes (B-tree, GIST)	Moderate
MinIO	Large files (media, GeoTIFF)	Web apps, ML	Metadata only	Yes (horizontal)
Neo4j	Relationships, topologies	AI/Graph tools	Yes (graph)	Moderate

DATA SYNCHRONIZATION & INTEGRITY

- **Metadata Linking**: PostgreSQL tables maintain references to MinIO object URIs and Neo4j node IDs.
- **Transactional Coordination**: APIs and pipelines ensure ACID compliance by staging data in PostgreSQL before object uploads or graph creation.
- **Validation Routines**: Ingestion scripts verify that files and relational records are synchronized and referenced correctly.

4.3 DATASET ORGANIZATION AND FILE STRUCTURE

The PANTHEON platform manages an extensive and diverse set of datasets across four primary emergency scenarios: earthquake, heatwave, wildfire, and cyberattack/explosion. The dataset organization is carefully structured to ensure **scenario-specific clarity**, **cross-scenario integration**, and **high performance** for emergency response simulations. The datasets are fully described in previous chapter.

HIGH-LEVEL FOLDER STRUCTURE

Datasets are grouped by scenario and stored using a standardized directory and naming convention:

```
/datasets/
├── earthquake/
│   ├── input/
│   └── output/
├── heatwave/
│   ├── input/
│   └── output/
├── wildfire/
│   ├── input/
│   └── output/
├── cyberattack/
│   ├── input/
│   └── output/
└── shared/
    └── base_data/
```

Each scenario includes:

- **`input/`**: Raw and preprocessed datasets (e.g., GeoJSONs, CSVs, NetCDFs).
- **`output/`**: Simulation results, optimized routing files, and final allocations.

FILE NAMING CONVENTIONS

All files use a uniform naming scheme:

```
{scenario}_{datatype}_{YYYY-MM-DD}_{sequence}.{ext}
```

Examples:

- `earthquake_BlockedRoads_2025-05-21_0.geojson`
- `heatwave_AllocationResults_2025-05-30.csv`

This format supports versioning, timestamp tracking, and automated ingestion into analytics pipelines.

STORAGE SYSTEMS

The datasets are integrated into a federated architecture using:

- **MinIO**: Object storage for large files (satellite imagery, elevation, fire simulations).
- **PostgreSQL/PostGIS**: Vector and tabular data with spatial indexing.
- **Kafka**: Real-time data streaming using unified topic structures per scenario.
- **Neo4j**: Graph-based infrastructure modeling (e.g., interdependent utilities).

DATASET CATEGORIES

Each scenario includes common thematic categories:

Table 6. Dataset Categories

Category	Example Data Types	Formats
Geographic Infrastructure	Road networks, buildings, POIs	GeoJSON, Shapefile
Environmental Conditions	Seismic, weather, solar radiation	CSV, NetCDF, GeoTIFF
Population Demographics	Elderly density, general population grids	CSV
Emergency Resources	Hospitals, schools, shelters	CSV, GeoJSON
Simulation Outputs	Routes, blocked roads, fire zones	GeoJSON

SHARED CROSS-SCENARIO DATASETS

A `/shared/base_data/` folder contains foundational datasets reused across all scenarios:

- **OpenStreetMap**: Road, building, and utility infrastructure.
- **HumData Population Grids**: High-resolution population estimates.
- **Satellite Imagery (Copernicus Sentinel-2)**: Used for visual basemaps and analysis.
- **Elevation Data (NASA SRTM/Copernicus DEM)**: Used for slope analysis and UAV planning.

METADATA AND STANDARDS

Each dataset is accompanied by metadata:

- **Coordinate System**: WGS84 (EPSG:4326)
- **Metadata Fields**: Geographic bounds, confidence levels, data source, file size, timestamp
- **Schema Definitions**: Common JSON schemas for impact, routing, and risk data

SCENARIO-SPECIFIC STORAGE BUCKETS (MINIO EXAMPLE)

```
MinIO Buckets:
├── ds-ath-a-blocked-roads/
├── ds-vie-a-impact-assessment-region/
├── ds-ath-b-fire-propagations/
└── ds-vie-b-smoke-propagations/
```

Each bucket follows Kafka topic alignment and supports scalable, streamable access.

4.4 DATA INGESTION AND PROCESSING PIPELINES

The PANTHEON platform implements robust, modular data ingestion and processing pipelines to manage complex, heterogeneous datasets across multiple emergency scenarios. These pipelines ensure that data from diverse sources—ranging from raw sensor outputs to authoritative public datasets—is standardized, validated, transformed, and integrated into the platform's real-time decision support workflows.

INGESTION ARCHITECTURE OVERVIEW

The data ingestion framework is structured around a **multi-layer pipeline** integrating:

- **Batch Ingestion**: For static or periodic datasets (e.g., demographic CSVs, shapefiles, GeoTIFFs).
- **Streaming Ingestion**: For dynamic sources (e.g., real-time weather APIs, drone feeds).
- **ETL Jobs**: Standardized Extract, Transform, Load routines for pre-processing.
- **Unified Messaging**: Kafka-based topic structure to ensure traceable, scenario-specific routing of ingested data.

Core Technologies:

- **Apache Kafka** for real-time streaming
- **MinIO** for object storage (e.g., satellite imagery, fire propagation models)
- **PostgreSQL/PostGIS** for geospatial database ingestion
- **Python** for data transformation

STANDARDIZED DATA INGESTION PIPELINE

Each dataset follows a common ingestion workflow:

`[Source] → [Extractor] → [Transformer] → [Validator] → [Serializer] → [Storage & Messaging]`

Pipeline Stages:

- **Source Acquisition**
  - Pull from authoritative APIs (e.g., OpenWeatherMap, Google Solar, ERA5)
  - Download or receive via SFTP (e.g., HumData, Copernicus DEM)
  - Sensor push or manual upload (e.g., UAV data, local simulations)
- **Extraction**
  - Parse formats: CSV, GeoJSON, NetCDF, Shapefile, TIFF
  - Georeferencing and decoding of embedded metadata
  - Extract geospatial and temporal metadata
- **Transformation**
  - Coordinate normalization (WGS84 standard)
  - Data cleaning (null handling, type coercion, reclassification)
  - Spatial operations (buffering, overlay, clipping)
  - Scenario tagging (`scenario_type`, `city`, `component`)
- **Validation**
  - Schema compliance (based on JSON/CSV/GeoJSON schemas)
  - Range checks (e.g., population values, coordinates, impact scores)
  - Cross-validation with historical or authoritative reference layers
- **Serialization & Publishing**
  - Storage in appropriate backends (MinIO, PostgreSQL/PostGIS)
  - Kafka topic publishing with enriched metadata
  - Topic naming: `ds-{city}-{scenario}-{component}`
- **Logging and Monitoring**
  - Logs for ingestion time, size, errors, and Kafka publication
  - Real-time dashboards for data health and ingestion metrics

SCENARIO-SPECIFIC PROCESSING WORKFLOWS

Each scenario incorporates custom logic on top of the base ingestion pipeline:

Earthquake (Attica)

- **Seismic Analysis**: Apply PGA and intensity scales to buildings via spatial joins.
- **Blocked Road Detection**: Compute overlaps of seismic zones with road networks.
- **Routing Graph Generation**: Weight road segments with damage levels and accessibility.

Heatwave (Vienna)

- **Cooling Demand Modelling**: Merge elderly population data with solar radiation maps.
- **Facility Capacity Matching**: Assign individuals to parks or cooling spots using precomputed distance matrices.
- **Optimization**: Allocate vulnerable populations based on risk and proximity.

Wildfire (Attica)

- **Fire Spread Simulation**: Time-step ingestion of fire perimeters from propagation GeoJSONs.
- **Terrain and Wind Integration**: Elevation and wind profiles fused into fire model inputs.
- **Exposure Estimation**: Overlay population grids with dynamic burn zones.
- **Blocked Road Detection**: Compute overlaps of seismic zones with road networks.
- **Routing Graph Generation**: Weight road segments with damage levels and accessibility.

Cyberattack/Explosion (Vienna)

- **Plume Dispersion Ingestion**: Gaussian plume grid values ingested from simulation outputs.
- **Infrastructure Impact Graph**: Critical infrastructure dependency graph updated in Neo4j.
- **Blocked Road Detection**: Compute overlaps of seismic zones with road networks.
- **Routing Graph Generation**: Weight road segments with damage levels and accessibility.

DATA SYNCHRONIZATION AND UPDATE STRATEGIES

Table 7. Fdata Update SDtrategy

Source Type	Update Frequency	Method
Static Datasets	Monthly or ad hoc	Manual trigger or cron ETL
Public APIs	Hourly	Automated API polling scripts
UAV/Drone Feeds	Near-real-time	Kafka-connected ingestion
Simulation Outputs	On-demand	Pipeline trigger post-simulation

All datasets are versioned, timestamped, and assigned a unique simulation or processing ID to enable rollback, auditing, and comparative analysis.

DATA QUALITY AND ERROR HANDLING

The ingestion pipeline includes:

- **Automated anomaly detection** (e.g., missing coordinates, empty polygons)
- **Schema mismatch logging**

- **Quarantine zone** for failed ingestions (with retry or manual review workflows)
- **Metadata completeness checks** for coordinate systems, time coverage, source attribution

INTEGRATION WITH PROCESSING COMPONENTS

Post-ingestion, data is automatically routed to the appropriate PANTHEON processing component:

- **Impact Assessment**
- **Routing and Optimization**
- **UAV Coordination**
- **Cooling DSS**
- **Population Allocation**
- **Self-Adaptive Feedback Systems**

Each component consumes Kafka topics with predictable structure and standardized message schemas.

4.5 QUERY AND RETRIEVAL MECHANISMS

The PANTHEON platform provides powerful, flexible data access through a standardized set of query and retrieval mechanisms. These APIs and interfaces allow authorized users and systems to extract both raw and processed data from the federated data storage, supporting scenario analysis, simulation monitoring, visualization, and decision-making workflows.

The platform supports both **synchronous queries** via RESTful APIs and **streaming subscriptions** via Kafka for real-time updates. Retrieved data is available in standard formats, including **CSV** and **JSON**, to ensure compatibility with analytical tools, dashboards, and external systems.

API SPECIFICATIONS

All APIs conform to RESTful principles and follow a consistent URI structure:

Base Endpoint Structure:

```
GET /api/v1/{scenario}/{component}/{dataset}
```

Common Parameters:

Table 8. API specifications-general

Parameter	Type	Description
scenario	String	One of: `earthquake`, `heatwave`, `wildfire`, `cyberattack`
component	String	Submodule or processing unit, e.g., `routing`, `impact`, `allocation`
dataset	String	Specific data type, e.g., `blocked_roads`, `shortest_paths`, `cooling_spots`
format	String	Optional: `json` (default), `csv`
date	Date	Optional date filter (e.g., `2025-05-21`)
bbox	String	Optional bounding box filter (`minX,minY,maxX,maxY`)
limit	Integer	Max number of records to return (pagination)

Authentication:

- Token-based authentication via **Keycloak**
- Role-based access control per user group (e.g., researcher, planner, responder)

Sample Endpoints:

- **Retrieve blocked roads (GeoJSON):**

  `GET /api/v1/earthquake/routing/blocked_roads?date=2025-05-21&format=json`

- **Get heatwave allocation summary (CSV):**

  `GET /api/v1/heatwave/allocation/final_results?format=csv`

- **Retrieve population grid within bounding box:**

  `GET /api/v1/shared/demographics/population?bbox=16.36,48.22,16.42,48.24`

SAMPLE QUERIES AND USE CASES

Use Case 1: Emergency Planner Requests Road Constraints

`GET /api/v1/earthquake/routing/blocked_roads?format=json`

Returns a GeoJSON of currently inaccessible roads due to seismic damage. Used for dynamic routing engine updates.

Use Case 2: Researcher Downloads Heat Vulnerability Allocations

`GET /api/v1/heatwave/allocation/final_results?format=csv`

Downloads population allocation to cooling spots and parks, including travel distances and risk scores for further statistical analysis.

Use Case 3: UAV Team Requests Elevation Data

`GET /api/v1/shared/terrain/elevation?bbox=23.67,38.10,23.72,38.14`

Returns sliced elevation data (GeoTIFF converted to JSON or CSV) for UAV flight planning in wildfire zones.

Use Case 4: Real-Time Subscriber for Fire Spread Updates

Subscribed to Kafka topic:

`ds-ath-b-fire-propagations`

Receives periodic GeoJSON updates of wildfire perimeter growth every 15 minutes.

Use Case 5: Exporting All Hospitals in Vienna

`GET /api/v1/shared/infrastructure/hospitals?city=vienna&format=csv`

Used by logistics teams to identify nearby healthcare capacity during cyberattack or explosion scenarios.

OPEN API SPECIFICATION

```
openapi: 3.0.3
info:
  title: PANTHEON Data Retrieval API
  version: 1.0.0
  description: >
    This API provides access to processed and raw datasets for multi-hazard
emergency scenarios (earthquake, heatwave, wildfire, cyberattack).
    Supports JSON and CSV exports, authenticated access, and spatial queries.
```

```
servers:
  - url: https://api.pantheon-platform.eu/api/v1

paths:
  /{scenario}/{component}/{dataset}:
    get:
      summary: Retrieve data for a specific scenario/component/dataset
      tags: [Query]
      parameters:
        - in: path
          name: scenario
          required: true
          schema:
            type: string
            enum: [earthquake, heatwave, wildfire, cyberattack]
          description: Scenario type
        - in: path
          name: component
          required: true
          schema:
            type: string
          description: Component name (e.g., routing, impact, allocation)
        - in: path
          name: dataset
          required: true
          schema:
            type: string
          description: Dataset identifier (e.g., blocked_roads, final_results)
        - in: query
          name: format
          required: false
          schema:
            type: string
            enum: [json, csv]
          description: Output format (default: json)
        - in: query
          name: date
          schema:
            type: string
            format: date
          description: Filter by specific date (YYYY-MM-DD)
        - in: query
          name: bbox
          schema:
            type: string
            example: "23.67,38.10,23.72,38.14"
          description: Bounding box for spatial queries
(minLon,minLat,maxLon,maxLat)
        - in: query
          name: limit
          schema:
            type: integer
            default: 1000
          description: Limit number of returned records
      responses:
        '200':
          description: Successful data retrieval
          content:
```

```
            application/json:
              schema:
                type: object
                example:
                  type: FeatureCollection
                  features:
                    - type: Feature
                      geometry:
                        type: LineString
                        coordinates: [[23.669, 38.102], [23.670, 38.103]]
                      properties:
                        type: BlockedRoad
                        severity: complete
                        confidence: 0.9
            text/csv:
              schema:
                type: string
                example: |
                  district_id,park_name,allocated_population,remaining_capacity
                  2,Augarten,1425,315
                  4,Stadtpark,998,0
                  5,Donaupark,1840,120
        '400':
          description: Invalid request parameters
        '401':
          description: Unauthorized access
        '404':
          description: Dataset not found
        '500':
          description: Internal server error
      security:
        - bearerAuth: []

components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
      bearerFormat: JWT

tags:
  - name: Query
    description: Scenario-based data query endpoints
```

DATA EXPORT FORMATS

Data retrieved through the APIs can be exported in **CSV** or **JSON**, depending on user preference or system compatibility.

JSON Format

Default format for GeoJSON, object arrays, or Kafka messages

Well-suited for geospatial visualization, map rendering, and graph-based APIs

Example:

```
{
  "type": "FeatureCollection",
```

```
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [[23.669, 38.102], [23.670, 38.103]]
      },
      "properties": {
        "type": "BlockedRoad",
        "severity": "complete",
        "confidence": 0.9
      }
    }
  ]
}
```


CSV Format

- Ideal for tabular datasets (allocations, population grids, infrastructure lists)
- Readable by Excel, R, Python pandas, etc.

Example 1:

```
osmid,y,x
150561883,38.0990521,23.6685049
330844702,38.1072985,23.6670819
330844706,38.1084351,23.6717049
330844708,38.1074979,23.6744051
```

Example 2:

```
u,v,name
150561883,936539731,"{'osmid': 111436653, 'highway': 'residential', 'maxspeed': '50', 'name': 'Ιάσμου', 'oneway': False, 'reversed': True, 'length': np.float64(6.810426653090085)}"
330844702,2527701883,"{'osmid': 112985440, 'highway': 'tertiary', 'lanes': '2', 'maxspeed': '50', 'name': 'Φυλής - Πύλης', 'oneway': False, 'reversed': True, 'length': np.float64(22.882095805404266)}"
```

INTEGRATION WITH OTHER COMPONENTS

- **Dashboard Tools**: Web clients query APIs directly for visual overlays (maps, heatmaps).
- **Python Clients**: Built-in SDK functions to retrieve and convert responses to DataFrames.
- **External Systems**: Third-party emergency platforms can ingest data via secure API gateways.

4.6 SECURITY PRINCIPLES AND REQUIREMENTS

The PANTHEON platform is designed with robust security principles to ensure data confidentiality, integrity, availability, and accountability across all disaster scenarios and system components. Given the sensitive nature of emergency datasets, including population exposure, infrastructure vulnerabilities, and real-time simulation outputs, a comprehensive security framework is applied at all architectural levels.

CORE SECURITY PRINCIPLES

The system adheres to the following foundational principles:

- **Confidentiality**: Sensitive data (e.g., vulnerable population locations, critical infrastructure) is encrypted in transit and protected via authenticated access.
- **Integrity**: All data operations follow ACID-compliant transactions to prevent corruption and ensure consistency across the system.
- **Availability**: Infrastructure is designed with resilience and monitoring to ensure system availability during emergencies and high-load events.
- **Accountability**: All API access and data operations are logged with traceable user identities and timestamps, ensuring full auditability.

DATA MANAGEMENT INTEGRITY: ACID COMPLIANCE

All structured data operations performed via the PostgreSQL/PostGIS backend comply with **ACID** properties:

Table 9. ACID Compliance

Property	Implementation Detail
Atomicity	Transactions are executed fully or not at all using PostgreSQL transaction blocks.
Consistency	Schema constraints, data validation, and referential integrity ensure valid states before commit.
Isolation	Concurrent data operations are isolated to avoid race conditions using serializable or repeatable read isolation levels.
Durability	All committed data is immediately written to disk and backed up using WAL (Write-Ahead Logging).

This guarantees that simulation results, demographic overlays, and emergency routing outputs remain reliable, even in concurrent or high-load environments.

SECURE DATA TRANSMISSION: SSL/TLS ENCRYPTION

All data in transit across the PANTHEON system is encrypted using **TLS 1.2+**, including:

- API requests/responses
- Internal service-to-service communication
- Kafka data streaming
- Authentication redirections via Keycloak

Certificates are managed using Let's Encrypt or institutional CA providers, and mutual TLS is optionally available for critical internal communications.

API SECURITY: TOKEN-BASED ACCESS CONTROL

All REST API endpoints enforce authentication and authorization through **token-based access control** mechanisms:

- **Access Tokens**: JWT tokens issued by Keycloak, included in the `Authorization: Bearer <token>` header.
- **Scopes and Roles**: Role-based access control (RBAC) enforced per user group (e.g., administrator, planner, researcher).
- **Endpoint Protection**: Public, protected, and admin-only endpoints are defined with clear policies.

Sample Authorization Flow:

- User logs in via PANTHEON web interface → Redirected to Keycloak.
- Upon successful authentication, Keycloak issues a JWT.
- All subsequent API calls include the JWT in headers.
- Backend services validate the token's signature and embedded roles.

AUTHENTICATION & IDENTITY MANAGEMENT VIA KEYCLOAK

Keycloak serves as the central Identity and Access Management (IAM) system:

- **Single Sign-On (SSO)**: Unified login for web dashboards, APIs, and simulation control interfaces.
- **OAuth 2.0 & OpenID Connect**: Standards-based protocols for secure identity federation.
- **Multi-Factor Authentication (MFA)**: Optional for privileged accounts and critical components.
- **Fine-Grained Role Management**: Custom realm roles and client scopes define access to scenario-specific APIs or datasets.

ADDITIONAL SECURITY MECHANISMS

Table 10. Additional; security mechanisms

Category	Implementation
Audit Logging	All API requests, dataset access, and admin actions are logged and monitored.
Input Validation	API payloads are validated against JSON schemas or model definitions.
CORS Policies	Strict cross-origin settings configured per frontend app.
CSRF Protection	Enabled for authenticated web clients using secure cookies.
Container Security	Services are deployed in isolated Docker containers with least-privilege execution.
Backup & Recovery	Encrypted, periodic backups of all critical data.

4.7 AUTHENTICATION AND AUTHORIZATION

Keycloak provides a centralized and standards-based identity and access management (IAM) solution for securing the PANTHEON platform. It enables Single Sign-On (SSO), role-based access control, and token issuance for web interfaces, APIs, and backend systems such as Kafka.

WEB FRONTEND (USER INTERFACE) ACCESS

Authentication Workflow

- **Redirect to Login**: When a user accesses a protected web application (e.g., dashboard), they are redirected to the Keycloak login page.
- **Login via UI**: Users authenticate using username/password (or MFA if configured).
- **Token Issuance**: On success, Keycloak issues:
    - **ID Token**: Used by the frontend for user info display.
    - **Access Token (JWT)**: Used to authenticate subsequent API requests.
    - **Refresh Token**: For silent session renewal.

Authorization

- The frontend uses **OpenID Connect (OIDC)** implicit or authorization code flow.
- User roles are embedded in the access token (e.g., `role:planner`, `role:responder`).
- Role-based UI rendering (e.g., hide admin panel if user lacks `admin` role).
- Optional **realm-level roles** or **client-specific roles** control what datasets a user can access or actions they can take.

REST API ACCESS

Authentication

- APIs require a valid **Bearer token (JWT)** in the `Authorization` header:
- Token is typically obtained by the frontend or backend using the **Authorization Code Flow** or via **client credentials** (for machine-to-machine access).

Authorization Mechanism

- API endpoints are protected using **resource-based or role-based policies**:
    - Example: Only users with role:admin can call POST /api/v1/scenario/upload
    - Users with role:planner may access GET /api/v1/heatwave/allocation/final_results
- Fine-grained access policies can be defined using **Keycloak Authorization Services**, which support:
    - Role-based access control (RBAC)
    - Attribute-based access control (ABAC)
    - Resource permissions and scopes

3. KAFKA ACCESS (SECURE STREAMING)

Kafka is integrated with Keycloak to enable **fine-grained authentication and authorization** for **producers**, **consumers**, and **stream processors**.

Authentication Workflow

Kafka clients (e.g., Python, Java) authenticate using **OAuth 2.0** with client credentials:

- The client requests an **access token** from Keycloak using `client_id` and `client_secret`.
- The token is added to Kafka client configuration.
- Kafka brokers validate the token using **JWT introspection** or a public key.

Sample OAuth2 token request:

```
curl -X POST \
  https://keycloak.pantheon-platform.eu/realms/pantheon/protocol/openid-
connect/token \
  -d "grant_type=client_credentials" \
  -d "client_id=kafka-client" \
  -d "client_secret=ABC123XYZ"
```

Authorization Mechanism

Kafka brokers are configured to use **SASL/OAUTHBEARER** mechanism:

```
sasl.mechanism=OAUTHBEARER
security.protocol=SASL_SSL
```

Authorization is based on:

- Client roles embedded in the token (e.g., `kafka_producer`, `kafka_consumer`)
- Topic-level permissions configured in Kafka ACLs or enforced via **Kafka-Authorizer + Keycloak adapter**

Example Policy:

- Client with `role:kafka_fire_sim` can:
- Produce to `ds-ath-b-fire-propagations`
- Consume from `ds-ath-b/uav`

SESSION AND TOKEN MANAGEMENT

Table 11. Session and Token management

Feature	Description
Token Expiry	Short-lived access tokens (5–15 min), refreshable via refresh token
Revocation	Admins can revoke tokens and sessions via Keycloak Admin Console
Session Timeout	Configurable idle and max session lifetimes
Audit Logging	All logins, token requests, and permission checks are logged

SECURITY SUMMARY TABLE

Table 12. Security summary

Access Type	Protocol	Authentication	Authorization
Web UI	OIDC	User login	Role-based views + data filtering
REST API	OAuth2 / OIDC	Bearer Token	Role/Resource-based policies
Kafka Client	OAuth2	Client Token	Topic-level ACLs via roles

4.8 DATA ENCRYPTION STRATEGIES

To ensure the **confidentiality**, **integrity**, and **compliance** of sensitive emergency response data across the PANTHEON platform by applying robust encryption mechanisms at all stages of the data lifecycle: **in transit**, **at rest**, and **during processing** when applicable.

DATA CLASSIFICATION & SENSITIVITY

PANTHEON handles data of varying sensitivity:

Table 13. Data sensitivity

Data Type	Sensitivity Level	Encryption Required
Population grids & demographics	High	At rest + in transit
Road networks & building data	Medium	In transit
Simulation outputs (routes, fire)	High	At rest + in transit
Satellite imagery & UAV data	High	At rest + in transit
User credentials & access tokens	Critical	At rest (hashed) + in transit

DATA IN TRANSIT

All data transmitted between clients, APIs, internal services (Kafka, MinIO, PostgreSQL), and external sources (e.g., weather APIs) will use **TLS v1.2 or higher** to ensure secure communications.

Implementation:

- **HTTPS/SSL** for all web APIs and web clients
- **Kafka + SASL_SSL** for secure message streaming
- **Keycloak** and all auth redirects secured with HTTPS

KEY MANAGEMENT AND ROTATION

Table 14. Key management and rotation

Aspect	Recommendation
Key Length	Minimum 256-bit AES keys
Key Storage	External Key Management System (KMS or Vault)
Key Rotation	Automatic every 90–180 days
Audit Logging	All key usage and changes are logged and audited
Access Control	Restricted to security admins with MFA

COMPLIANCE ALIGNMENT

Encryption strategy aligns with:

GDPR: Personal data protection (Articles 5, 32)

NIS2: Cybersecurity and infrastructure resilience

ISO/IEC 27001: Information Security Management Systems

IMPLEMENTATION ROADMAP

Table 15. Security implementation roadmap

Phase	Deliverables
Phase 1	TLS enforcement, HTTPS migration, Kafka SSL
Phase 2	MinIO SSE setup, PostgreSQL encryption enabled
Phase 3	Keycloak integration with Vault/KMS
Phase 4	Field-level encryption and key rotation policy
Phase 5	Compliance audit and penetration testing

4.9 SYSTEM-SPECIFIC SECURITY MECHANISMS

The PANTHEON platform employs a distributed architecture integrating PostgreSQL, Kafka, MinIO, and Neo4j — each secured using tailored, system-specific security configurations. These mechanisms collectively ensure **data confidentiality**, **integrity**, **availability**, and **auditability** across all operational components.

POSTGRESQL / POSTGIS SECURITY

Authentication

- Supports **SCRAM-SHA-256** or **client certificate-based** authentication.
- Integrated with **Keycloak** via OIDC-proxy (optional) or service accounts.
- Database roles mapped to application users with limited privileges.

Authorization

- Role-based access control (RBAC) using `GRANT`, `REVOKE`, and schema-level permissions.
- Row-Level Security (RLS) enabled for sensitive tables to restrict access based on user role or region.

Encryption

- **In transit**: SSL/TLS enabled (`ssl = on`, certificates required for all connections).
- **At rest**:
  - Filesystem-level encryption using LUKS/dm-crypt or
  - Encrypted tablespaces and pgcrypto for field-level AES-256 encryption.

Audit

- Enabled via **`pgaudit`** extension or `log_statement = all` for admin actions.

- Logs include login attempts, DDL/DML operations, failed queries, and access violations.

APACHE KAFKA SECURITY

Authentication

- Uses **SASL/OAUTHBEARER** mechanism with Keycloak for client authentication.
- Brokers validate tokens using Keycloak's public key or token introspection endpoint.

Authorization

- Kafka ACLs configured per **topic**, **consumer group**, and **operation**:
  - User:kafka-fire -> Write -> Topic:ds-ath-b-fire-propagations
  - User:kafka-uav -> Read -> Topic:ds-ath-b/uav
- Dynamic topic protection via Kafka Authorizer + Keycloak adapter.

Encryption

- **In transit**: TLS encryption for all client-broker and inter-broker communication (`security.protocol = SASL_SSL`).
- Optionally supports **mTLS** for trusted internal communications.

Audit

- Broker logs track every authentication attempt, ACL denial, and connection status.
- Integration with **Confluent Control Center**, **Grafana**, or **Prometheus exporters** for monitoring.

MINIO OBJECT STORAGE SECURITY

Authentication

Access via **Keycloak JWT tokens** using OIDC provider configuration.

Supports **service accounts** with access key and secret for automated components.

Authorization

- Fine-grained **bucket-level and prefix-level** policies:
- Example:

```
"Statement": [{
  "Action": ["s3:GetObject"],
  "Resource": ["arn:aws:s3:::fire-data/*"],
  "Effect": "Allow",
  "Condition": {"StringEquals": {"jwt:role": "fire_viewer"}}
}]
```

Encryption

- **Server-side encryption (SSE)**:
  - **SSE-S3**: Internal AES-256 with auto-rotation.
  - **SSE-KMS**: External key management via HashiCorp Vault or AWS KMS.
  - **Client-side encryption** supported for highly sensitive datasets (e.g., field-collected UAV data).

Audit

- All API actions (GET, PUT, DELETE) logged with:

- Caller identity (JWT subject or access key ID)
- Object accessed
- Timestamp

- Integration with centralized logging tools (e.g., Fluentd, ELK stack).

NEO4J (GRAPH DATABASE) SECURITY

Authentication

- Native authentication using hashed credentials (bcrypt).
- Optional integration with **Keycloak** via **OIDC plugin** or **LDAP bridge** for SSO.

Authorization

- Role-based security model:
  - reader, editor, admin, and **custom roles** (e.g., infra_analyst)
- Graph-level access control via **fine-grained privileges** on labels, properties, and procedures:
  - GRANT TRAVERSE ON GRAPH pantheon NODES infra_element TO infra_analyst;

Encryption

- **Encrypted bolt protocol** (`bolt+s`) for Neo4j drivers.
- HTTPS enabled for Neo4j Browser and API access.
- Certificate pinning and secure CAs for production deployments.

Audit

- Query logs enabled with:
  - Full command tracing
  - Access source (IP, user, timestamp)
  - Failure reason (if any)
- Supports external audit plugins and integration with log aggregators.

SUMMARY TABLE

Table 16. Security summary for Postgres, Kafka, Minio, Neo4j

Component	Authentication	Authorization	Encryption	Auditing
PostgreSQL	SCRAM / Client Certs	SQL GRANTs + RLS	SSL, AES, pgcrypto	pgaudit, logging
Kafka	Keycloak (OAuth2)	Kafka ACLs per role/topic	TLS, SASL_SSL	Broker logs + Prometheus
MinIO	Keycloak OIDC / Tokens	Policy-based (S3-compatible)	SSE-S3/KMS, TLS	Object access logs
Neo4j	Native / OIDC	Graph privileges	Bolt+TLS, HTTPS	Query logs, plugin extensions

5. TECHNICAL CONSIDERATIONS

5.1 OVERVIEW OF REQUIRED TECHNICAL CHARACTERISTICS

A secure data repository must have at least the following four technical characteristics: must be compliant with the regulation, especially with respect to the personal data restrictions, it must also be scalable, this is able to work for small and large datasets, it must be also accessible within reasonable time frame and shall have the capacity to recover previous versions and keep track of all the changes. The following sections will briefly describe the pantheon storage system with respect to these four technical characteristics.

5.2 REGULATORY CONSIDERATIONS (GDPR)

The General Data Protection Regulation1 (GDPR)[3] is a comprehensive data protection law enacted by the European Union to safeguard the personal data and privacy of individuals from the EU and the European Economic Area. It also addresses the export of personal data outside these regions. For any secure data repository handling personal or sensitive information, adherence to GDPR is essential to ensure legal compliance, protect user privacy, and maintain trust. This includes implementing appropriate technical and organizational measures to ensure data integrity, confidentiality, and accessibility, while also enabling subjects to demand for their rights such as access, rectification, erasure, and portability.

The PANTHEON digital twin does not use any personal data, neither for the simulations nor for the training, thus no secured data repository for personal data subject to GDPR has been needed.

5.3 SCALABILITY

Scalability is a critical consideration in the development and deployment of a digital twin. In PANTHEON the digital twin is designed for training first responders and for supporting emergency response planning. The area extension and resources to manage in the digital twin can have huge variations from one execution to another, from one end-user to the next. The complexity and scope of simulated scenarios may range from localized incidents to large-scale, multi-region natural or hand-made disasters. For this reason, scalability is a major issue in PANTHEON.

The PANTHEON digital twin must be capable of scaling both in terms of data volume and computational resources. This includes:

- accommodating diverse geographic areas with many useful data layers to feed into a geographical information system (GIS),
- managing the number of simulated agents (e.g., responders' teams, drones, trucks, communities, roads, meeting points, evolution of the disaster itself), and
- adding external data sources such as weather systems, IoT sensors, and communication networks.

[3] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016*

A scalable system shall ensure that functionalities remain realistic and responsive under various load conditions. In PANTHEON this means that emergency planners shall be able to effectively test and adapt response strategies across a range of disaster scenarios and magnitudes.

Cloud-based and distributed architectures offer significant advantages over traditional centralized local solutions to meet these scalability challenges. Cloud platforms provide elastic compute and storage resources that can dynamically scale based on real-time demand, enabling the digital twin to simulate large-scale disasters, incorporate streaming with environmental data, and support concurrent execution of its components as part of a distributed pipeline. All together will provide the requested functionality by expanding the components across distributed connected platforms. Distributed solutions also enhance system resilience and fault tolerance, both critical aspect in emergency contexts, where continuity and responsiveness are vital.

PANTHEON architecture leverages on containerization, in particular Docker, a very robust technology within cloud environments. Containerization allows modular scaling of individual system components, such as geospatial processing, AI-driven decision support systems, simulators of the disaster scenario, or drone fleet management. The PANTHEON modularity supports incremental expansion and facilitates integration with existing systems through API to GIS platforms, sensor networks, or traffic databases.

By adopting a scalable, cloud-native architecture, the digital twin becomes a robust, future-proof tool capable of adapting to evolving training needs and emergency planning requirements across a broad spectrum of disaster scenarios.

Scalability can be measured for different dimensions: maximum number of users that can simultaneously interact with the digital twin without degradation in performance (capacity), number of geographic regions able to integrate (coverage), volume of data the system can ingest and process per time unit (throughput), number of concurrent simulation entities (complexity), system's ability to automatically allocate or release cloud resources based on a time variable load (elasticity) and number of third-party systems (integration).

These indicators can be tested through performance benchmarks, load testing, simulation stress tests, and monitoring metrics in production or pilot environments. Typically, these tests generate a scalability evaluation matrix in which each dimension is assigned a simple scoring system (1 to 5), such that 1 means no scalability and 5 means full scalability. The system scalability can be computed from this matrix using the weighted average of the indicators, giving higher weights to the higher priority dimension for the system scalability according to the context.

5.4 PERFORMANCE

Performance refers to the system’s ability to deliver a smooth, responsive experience during simulation, training, and planning tasks under expected operational conditions. The performance of a system is given by a set of indicators that measure the limits of such a system. Most typical performance indicator is the execution time, which, in case of an interactive system such as PANTHEON digital twin, can also be named as response time or system latency.

For a digital twin used in emergency response, high performance is critical to ensure timely scenario rendering, data synchronization, user interaction, and decision support. To better differentiate with the previous section, performance is about speed, while scalability is about maintaining speed as demands grow. In summary:

- Performance refers to how well the system operates under current conditions, measured in terms like speed, responsiveness, latency, and throughput.
- Scalability refers to how well the system maintains or adapts performance as load increases in number of users, data, complexity, or size of the scenarios.

To measure the performance of a system, one can calculate time information in several ways. In particular:

1. System Latency: System latency measures the time delay between an input (such as a user input or a data update) and the response of the system.
    - Example: How quickly a user sees changes after modifying disaster parameters.
2. Frame Rate / Rendering Speed: For 3D environments, it measures how smoothly the interface and simulations are visualized.
    - Example: Frame rate consistency during complex visual simulations.
3. Data Processing Time: It measures how quickly data feeds (for instance, IoT sensor or weather data) are ingested, processed, and/or displayed.
    - Example: Real-time update delay of incoming drone or IoT data streams.
4. Simulation Load Time: This is the time required to launch or restart a new simulation, or switch between one scenario to another scenario.
    - Example: Delay when switching from a wildfire simulation to a flood response simulation.
5. System Reliability: Reliability measures capacity of a system to be available and running for unplanned loads or inputs.
    - Example: Up-time of the system for long training sessions.
6. Pipeline Responsiveness: Given a distributed system in which the data flows from one component to another the responsiveness measures the time breakdown of how much latency is due to network, server computation, or device limitations for each of the components.

Example: Delays added due to cloud communication time.

5.5 VERSIONING AND AUDITABILITY

Versioning and auditability are critical technical characteristics of a digital twin designed for high-stakes domains such as emergency response training and disaster planning. Both features ensure that all changes in models, simulation parameters, data inputs, or user actions are traceable, reversible, and accountable. In the case of a training simulation these are fundamental issues since they allow to track back the responders' actions and revisit them collectively to learn from potential mistakes. In a planning scenario they are also very useful because can help to understand how the plan adapts to the different simulation parameters and which are the limits of the resources.

Versioning enables the system to maintain a historical record of the temporal states of the digital twin, including scenario configurations, data models, and training modules. This capability allows planners and trainers to revert to previous versions, compare outcomes across iterations, and ensure consistency across sessions. It also supports continuous improvement by enabling structured testing and validation of new simulation updates without compromising existing configurations.

Auditability ensures that every interaction with the system, such as user inputs, data uploads, simulation runs, or role-based decisions, is logged in a secure and tamper-evident manner. This is essential for post-

incident analysis, regulatory compliance, and training accreditation. In multi-agency environments, audit trails provide transparency and trust, allowing stakeholders to verify how decisions were made and by whom during planning or simulated operations.

Together, versioning and auditability enhance the digital twin's reliability, governance, and long-term maintainability. When implemented within a secure, role-based access framework, they also support GDPR compliance and other data protection standards by allowing controlled oversight of data use and system behaviour.

TOOLS:

- git

The PANTHEON system currently employs git for version control, allowing the save versioned storage of all scenario configurations, simulation scripts, and training modules. All changes are tracked with full commit history, including metadata such as author, timestamp, and change description. This allows teams to revert to previous configurations, compare iterations, and ensure consistency across multi-agency simulations.

- Docker

Simulation models and supporting services are containerized using Docker enabling consistent deployment of versioned environments across different stages (development, testing, training). Each container is tagged and stored in a private registry, allowing reliable rollback and parallel deployment of different model versions for comparative analysis or regression testing.

- Kafka

Data streams, such as real-time sensor feeds, event updates, or agent telemetry, are managed through Kafka, which provides a high-throughput, fault-tolerant messaging layer. Kafka's topic-based architecture inherently supports data stream versioning, with message retention policies and consumer group tracking ensuring that historical data can be replayed or audited as needed. Kafka also includes snapshot mechanism, allowing complete states (environment, agents, configuration) to be captured, tagged, and archived. These snapshots are stored in cloud-based object storage with version metadata, making them retrievable for training audits, after-action reviews, and performance comparisons.

- MinIO

MinIO is a cloud storage service designed to allow the consolidation of data storage on a single, private cloud namespace. Using its server-side encryption (SSE) MinIO protects objects as part of write operations, allowing applications to take advantage of server processing power to secure objects at the storage layer (encryption-at-rest). SSE also provides key functionality to regulatory and compliance requirements around secure locking and erasure. In addition, MinIO uses a key encryption service from an external key management service for performing secured cryptographic operations at scale. MinIO also supports client-managed keys, this is, the application takes full responsibility for creating and managing encryption keys for use with MinIO SSE. MinIO versions each object independently following Amazon's S3 structure/implementation. When each new version of an object is written, it is assigned a unique version ID. Applications can specify a version ID to access a specific version of that object at a specific time. For auditory any object in MinIO can enable locking and then no version of that object will be deleted. Lock objects are immutable and read-only, thus they cannot be deleted and will be available for any auditor's legal requirement.

Together, these tools create a robust and transparent versioning framework, ensuring the digital twin remains accountable, adaptable, and fully auditable. These features are critical for high-reliability applications in emergency response planning and for inter-agency coordination.

6. MAINTENANCE AND MONITORING

6.1 MONITORING TOOLS AND DASHBOARDS

Effective and proactive monitoring is essential to ensure the reliability, performance, and responsiveness of the PANTHEON platform. This monitoring spans both **technical infrastructure metrics** and **business-level operational indicators**, providing comprehensive visibility into system health and scenario execution status.

To achieve this, PANTHEON integrates **Prometheus**, **Kibana**, and **Metricbeat/Filebeat/Custom Beats**, offering layered observability for DevOps and business stakeholders alike.

The monitoring infrastructure enables **real-time visibility**, **root cause diagnosis**, and **performance optimization** across both system-level and use-case-specific activities. By combining **Prometheus**, **Metricbeat**, and **Kibana**, PANTHEON supports both DevOps engineers and domain experts with tailored insights and responsive alerting mechanisms.

MONITORING ARCHITECTURE OVERVIEW

The monitoring stack consists of:

- **Prometheus**: Core time-series database and metrics scraper for system components.
- **Metricbeat**: Collects host-level and service-level system metrics (CPU, memory, disk I/O, etc.).
- **Filebeat**: Ships logs from services (e.g., PostgreSQL, Kafka, Neo4j) to Elasticsearch.
- **Custom Beats / Exporters**: Tailored metrics extraction from PostgreSQL, MinIO, Kafka, Neo4j, and application-level services.
- **Elasticsearch**: Central log and metrics repository.
- **Kibana**: Visualization layer for dashboards, alerts, anomaly detection.
- **Grafana** *(optional)*: Alternative visualization interface, especially for Prometheus-focused metrics.

TECHNICAL MONITORING METRICS

These metrics focus on **infrastructure and platform health**:

Table 17. Technical monitoring metrics

Component	Metrics Tracked
CPU & Memory	Per-node CPU load, RAM usage, swap activity (via Metricbeat)
Network	Incoming/outgoing traffic, connection errors, latency
Storage	Disk usage per mount, IOPS, free space alerts
PostgreSQL	Read/write queries/sec, slow queries, locks, replication lag (via pg_exporter)
Kafka	Topic write/read rates, consumer lag, broker health (via kafka_exporter)
MinIO	Bucket read/write rates, object count, size, latency (custom MinIO exporter)
Neo4j	Query latency, active connections, transactions/sec (via neo4j-exporter)

All metrics are scraped by Prometheus and visualized in **Kibana Dashboards** and optionally in **Grafana**.

BUSINESS-LEVEL MONITORING METRICS

These indicators monitor **platform usage, data processing, and scenario execution**, tailored for decision-makers and scenario operators.

Table 18. Business level metrics

Metric Category	Example Metrics
User Engagement	Number of active users (past hour/day/week), user login trends
Data Input Metrics	Number of files uploaded, volume (MB/GB), formats ingested
Data Output Metrics	Number of outputs exported, size, formats (CSV, JSON, GeoJSON)
Scenario Utilization	Active scenarios, state (running, paused, error), duration
Alerting & Feedback	Number of active alerts, resolved/unresolved status, user feedback entries
Workspaces	Count of workspaces per scenario, data volume by workspace
Uptime Metrics	Application/service uptime %, error rates, request latency (API-level)

All business metrics are tagged by `workspace_id`, `scenario_type`, `user_id`, and timestamps.

NOTIFICATION AND ALERTING MECHANISMS

Alerts are based on **Prometheus alert rules**, **Elasticsearch Watcher**, and **custom thresholds**. Alerts can be routed to:

- Web UI notifications
- Email/SMS for admins
- Kafka topics (`alerts.system`, `alerts.business`)
- Slack/MS Teams integration (optional)

Table 19. Alert Triggers

Example Alert Triggers	Action
CPU > 85% for 10 min	Show in DevOps Dashboard, notify via Kafka
PostgreSQL write latency > 300ms	Email admin and push to `alerts.db` topic
Scenario failed 3 times in a row	Display in scenario monitoring dashboard
Data export > 1GB from same user in <10min	Raise anomaly alert, mark in audit log
Unused workspace > 90 days	Alert owner for potential archival

DASHBOARD EXAMPLES

Technical Dashboard (Kibana / Grafana)

- System resource heatmap (CPU, memory, disk)
- Kafka topic throughput per scenario
- Database write/query heatmap
- Storage usage per bucket (MinIO)

Business Dashboard (Kibana)

- Active users over time
- Scenario execution timelines
- Workspace storage usage over time
- Input/output file volume distribution
- Alert trends by scenario type

SCALABILITY AND MAINTENANCE

Prometheus scraping interval: 15–30s depending on metric

Elasticsearch shard rotation: Weekly for logs, monthly for metrics

Retention policies:

- Metrics: 30–90 days
- Logs: 6–12 months

Archival to MinIO or cold storage (if configured)

6.2 LOGGING AND ALERTS

The PANTHEON platform integrates a unified, multi-layer **logging and alerting system** to ensure observability, traceability, and situational awareness across all operational components. This system tracks all significant events — user actions, data flows, system operations, and model outputs — while triggering contextual alerts based on configurable conditions and scenario-specific thresholds.

The system provides both **real-time visibility** (via the web UI) and **programmable access** (via APIs and Kafka topics), ensuring decision-makers and technical operators are promptly informed of critical developments.

LOGGING ARCHITECTURE

All layers (from data ingestion to frontend interactions) emit structured logs. These are stored and processed centrally, following a **log aggregation and enrichment pipeline**:

Types of Logged Events

- **User Actions**: Logins, role changes, API calls, data downloads, scenario executions.
- **Data Operations**: Dataset ingestions, deletions, metadata changes.
- **Simulation Events**: Model execution start/stop, input/output status, exceptions.
- **Security Events**: Authentication failures, token expiries, unauthorized access attempts.
- **System Metrics**: Kafka topic throughput, MinIO I/O, API latency, PostgreSQL queries.

Technologies Used

- **Logstash**: Log forwarding and filtering.
- **Elasticsearch / PostgreSQL**: Log storage and indexing.
- **Kibana / Custom Dashboards**: Visual exploration and audit trail interfaces.
- **Kafka Topics**: Real-time log streams for advanced processing (e.g., anomaly detection).

Example Log Entry

```
{
  "timestamp": "2025-06-04T10:21:15Z",
  "user": "planner_vienna",
  "action": "run_simulation",
  "scenario": "heatwave",
  "component": "allocation_engine",
  "status": "completed",
  "duration_ms": 5123,
  "ip": "192.168.20.11"
}
```

ALERTING MECHANISM

The alert system processes logged events against configurable rules and conditions, generating **contextual alerts** related to platform operations and scenario impacts.

Alert Types

Table 20. Alert Types

Alert Type	Trigger Example	Use Case Scenario
Data Quality	Missing coordinates in ingestion file	Wildfire (fuel map errors)
Impact Threshold	High population exposure (>10,000 persons)	Earthquake, Heatwave
System Failure	Model crash, Kafka timeout	All
Security	Unauthorized access or token misuse	All
Scenario-specific	Fire perimeter intersecting hospitals	Wildfire
Real-time source	Unusual spike in wind speed from ERA5	Wildfire

Alert Generation Pipeline

1. **Log Stream Ingestion** via Kafka
2. **Rule Evaluation Engine**: Evaluates rules defined per scenario/component
3. **Alert Enrichment**: Adds context (scenario, severity, source, timestamp)
4. **Dispatch**: Alerts pushed to:
    a. Kafka topics (e.g., `alerts.high_impact`)
    b. REST API endpoint: `GET /api/v1/alerts?filter=scenario:heatwave`

KAFKA-BASED ALERT DISTRIBUTION

Every generated alert is also published to **Kafka topics** for:

- Real-time AI processing (e.g., predictive adjustments)
- Operator dashboards
- Federation with third-party DRM platforms

Kafka Topic Structure:

```
alerts.earthquake.high_impact
alerts.wildfire.fire_perimeter
alerts.heatwave.population_risk
alerts.security.unauthorized_access
```

Each alert message is a structured JSON object containing:

```
{
  "alert_id": "A-2025-HEAT-042",
  "severity": "high",
  "message": "Over 15,000 elderly persons affected in districts 4, 6, and 9",
  "timestamp": "2025-06-04T10:34:00Z",
  "source": "allocation_module",
  "scenario": "heatwave",
  "resolved": false
}
```

API ACCESS

Alerts can be retrieved programmatically via a dedicated API:

Endpoint:

```
GET /api/v1/alerts
```

Query Parameters:

- `scenario=wildfire`
- `severity=high`
- `date=2025-06-04`
- `status=unresolved`

Response Format (JSON or CSV):

Alert ID, timestamp, message, component, resolution status, severity

SCENARIO-BASED LOGGING & ALERTS INTEGRATION

Table 21. Alert examples

Scenario	Unique Alert Examples
Earthquake	Damaged routes exceed 70%; allocation impossible
Heatwave	All parks in District 3 over capacity
Wildfire	Fire crosses major road; UAV required for recon

Cyberattack	Comms node inaccessible; fallback network activated

ALERT LIFECYCLE

1. **Generated**: From logs or model outputs
2. **Dispatched**: UI, Kafka, API
3. **Acknowledged**: Manually by an operator or auto-resolved
4. **Archived**: Retained in PostgreSQL and Elasticsearch for audits

COMPLIANCE AND AUDITING

- All alerts are stored with source logs for traceability.
- Modifications (acknowledgment, resolution) are logged.
- Alert rules are version-controlled and monitored for changes.

6.3 BACKUP/RESTORE PROCEDURES

PRINCIPLE: NO DATA LOSS (NDL)

The PANTHEON platform follows a **No Data Loss** (NDL) policy, meaning:

- All critical datasets and stateful services are backed up regularly and redundantly.
- The system supports **point-in-time recovery (PITR)** and **incremental recovery** to minimize downtime and data loss risk.
- Backups are **verified**, **encrypted**, and **replicated** off-site or cross-region.

Backup Frequency Overview

Table 22. Backup Frequency

Backup Type	Frequency	Coverage	Purpose
Incremental	Daily	Changed files/records only	Fast, space-efficient daily updates
Cumulative	Weekly	All changes since last full	Intermediate snapshot
Full	Monthly	Entire dataset/system	Long-term restore point

FILE SYSTEM BACKUP STRATEGY

Targets:

- Local configuration files
- Temporary cache folders
- Model binaries and environment scripts

Process:

- **Daily Incremental**: Use `rsnapshot` or `rsync` to sync changes to backup volume.
- **Weekly Cumulative**: Consolidated archive of the week's modified files (`tar.gz`).
- **Monthly Full**: Full mirror of system folders (excluding temp/volatile data).

Storage:

- Encrypted external disk or MinIO bucket
- Optional cloud sync (AWS S3, Azure Blob)

Recovery:

- File-based restore or directory tree rehydration

POSTGRESQL / POSTGIS BACKUP STRATEGY

Tools:

- `pg_basebackup`, `pg_dump`, `pgBackRest` (preferred for incremental backups)

Schedule:

- **Daily Incremental**: WAL archiving via `pgBackRest` with `archive_mode = on`.
- **Weekly Cumulative**: Differential backup with full WAL chain.
- **Monthly Full**: Complete snapshot of all clusters using `pg_basebackup`.

Features:

- **Point-in-time recovery (PITR)** supported
- **Per-database role backup** via `pg_dump` for partial recovery

Storage:

- Write to encrypted volumes or S3-compatible storage (MinIO bucket: `s3://backups/postgres/`)

Security:

- AES-256 encryption
- Encrypted credentials and restricted roles

Restore Mechanisms

- **Point-in-Time Recovery (PITR)**:
  - Restore from a base backup and replay WAL (Write-Ahead Logs) to a specific timestamp.
  - Tools: `pgBackRest`, `pg_basebackup`, `pg_wal`
- **Full Restore**:
  - Restore a full monthly backup using `pg_basebackup` or compressed SQL dumps (`pg_dump` + `pg_restore`).
  - Suitable for full server failures or corruption.
- **Selective Table/Data Restore**:
  - Use `pg_dump -t` and restore specific tables or schemas.
  - Useful for partial corruption or accidental deletes.

Policy

- Always restore to **staging/test** instance first to validate data integrity.
- Cross-verify spatial indexes and constraints (especially for PostGIS layers).
- Maintain at least 3 verified restore points per environment: latest daily, last weekly, and last monthly.
- Automated alerting if restore takes longer than expected (`alerts.restore.postgres.delayed`).

Validation

- Check row counts, primary keys, and foreign key constraints.
- Verify GIS functionality using sample spatial queries (`ST_Within`, `ST_Intersects`).

MINIO (OBJECT STORAGE) BACKUP STRATEGY

Targets:

- GeoTIFFs, UAV footage, simulation outputs, large datasets

Tools:

- `mc mirror` (MinIO client) for incremental and full backups
- S3-compatible backup utilities (e.g., Velero, Restic)

Schedule:

- **Daily Incremental**: `mc mirror --watch` for changed/new files (eventual consistency)
- **Weekly Cumulative**: Snapshot of buckets changed in the last 7 days
- **Monthly Full**: Snapshot entire MinIO cluster (or per-bucket)

Redundancy:

- Backups replicated across multiple zones or remote MinIO instances

Metadata Handling:

- Object metadata and versions stored with backups to support restoration

Restore Mechanisms

- **Per-Object Restore**:
  - Use `mc cp` or `mc restore` to selectively recover specific objects from backup storage.
- **Bucket-Level Restore**:
  - Restore entire buckets using `mc mirror` from backup mirror locations or snapshots.
- **Disaster Recovery**:
  - Re-deploy MinIO with metadata backup and rehydrate all objects into new instance.
- **Versioned Restore**:
  - If versioning is enabled, restore to previous versions using object metadata.

Policy

- Always log restored object IDs and metadata hashes.
- Retain backup bucket for 30 days post-restore to verify object integrity.
- Critical simulation datasets must be checksum-verified after restoring.

Validation

- Confirm file size and hash match (`sha256sum` or `etag`).
- Validate metadata (`mc stat`) and ensure object visibility in API layer.

NEO4J (GRAPH DB) BACKUP STRATEGY

Tools:

- `neo4j-admin backup`

- `neo4j-admin copy` for full offline snapshots

Schedule:

- **Daily Incremental**: Use `neo4j-admin backup` with `--fallback-to-full=false`
- **Weekly Cumulative**: Graph snapshot with store files and transaction logs
- **Monthly Full**: Full database export and index rebuild checkpoint

Storage:

- Compressed and encrypted backups stored in dedicated backup volume or object store (`s3://backups/neo4j`)

PITR:

- Supported via transactional logs and checkpoint files

Restore Mechanisms

- **Offline Full Restore**:
  - Stop Neo4j service and replace graph store directory with full backup using `neo4j-admin restore`.
- **Online Incremental Restore**:
  - Restore latest transaction logs and incremental data using `neo4j-admin backup` tools.
  - PITR support via transaction log replay (Enterprise Edition).
- **Selective Graph Element Restore** (manual):
  - Use Cypher scripts or `apoc.import.graphml` to re-import specific nodes/relationships.

Policy

- Always back-up and restore graph databases **offline** unless using HA cluster with rolling restarts.
- Restore to separate instance for comparison before switching production.
- Maintain change logs for schema updates and custom procedures.

Validation

- Run health checks: node counts, edge types, and graph integrity (`apoc.meta.stats()`).
- Run impact analysis queries to verify graph logic (e.g., connectivity, traversal results).

APACHE KAFKA BACKUP STRATEGY

Scope:

- Topic data (retained logs), configuration, state store (if using KStreams)

Approach:

- **Offset + message store backups** for replay/recovery

Schedule:

- **Daily**: Export message logs of high-priority topics to MinIO
- **Weekly**: Snapshot configuration and topic states
- **Monthly**: Archive complete topic partitions (where applicable)

Considerations:

- Keep log retention aligned with backup window (e.g., 30 days for high-value topics)

- Capture Zookeeper or KRaft metadata if using Zookeeper mode

Restore Mechanisms

- **Replay from Backup Topics**:
  - Restore Kafka logs from MinIO/S3 or HDFS using `kafka-console-producer` or a custom script.
- **MirrorMaker 2 Failover**:
  - If operating with cross-cluster replication, switch consumer/producer to failover Kafka cluster.
- **Kafka Streams Restore**:
  - Restore KTables or state stores from **internal changelogs** or RocksDB snapshots.
- **Configuration Restore**:
  - Restore `server.properties`, ACLs, and topic metadata from Git or Zookeeper snapshots (if applicable).

Policy

- High-priority topics (e.g., alerts, fire propagation) must support at least **7-day replay window**.
- Topic-level retention should align with backup intervals (e.g., `retention.ms = 604800000` for weekly).
- Topics are replayed into temporary namespaces (`restored.topic_name`) before re-routing traffic.

Validation

- Confirm message count parity with backup.
- Consume sample messages to verify schema integrity and completeness.
- Compare consumer group lags pre- and post-restore.

RETENTION AND ROTATION POLICY

Table 23. Retention Policy

Backup Type	Retention Duration	Purpose
Daily	7 days	Short-term corrections
Weekly	4 weeks	Interim recovery
Monthly	12–18 months	Long-term auditability

- Old backups are auto-rotated using retention policies.
- Verification checksums (`sha256sum`) are logged with each backup.
- Backup logs are themselves backed up and auditable via the logging system.

6.4 RECOVERY MECHANISM

GENERAL RECOVERY WORKFLOW

Recovery Staging Workflow

- Identify failure scope (file-level, service, full system).
- Isolate affected data and time window.
- Select appropriate restore type (incremental, cumulative, full).
- Restore to staging environment.
- Perform data integrity tests.
- Synchronize restored data with active services (if required).
- Re-enable traffic and monitoring.

Security in Restore

- Only authorized DevOps or recovery roles can initiate restores.

RECOVERY TESTING AND VERIFICATION

- **Monthly test restores** are performed in staging environments.
    - All backup jobs include:
    - **Completion status**
    - **Duration**
    - **Byte size**
    - **Integrity check logs**
- Alerts for failed backups are issued via:
    - Web UI alerts
    - Kafka topic: `alerts.backup.failures`
    - API: `GET /api/v1/backup/status?component=minio`

6.5 SYSTEM-SPECIFIC BACKUP IMPLEMENTATION

CONCEPT: SCENARIO AS A WORKSPACE

Each executed **scenario** (e.g., earthquake in Athens, wildfire in Fyli) generates and uses a set of interconnected datasets, models, results, and metadata. These are grouped into a **scenario workspace**, which includes:

- Input datasets (e.g., population, infrastructure)
- Simulation outputs (e.g., routes, risk zones)
- Logs, alerts, and audit trails
- Graph elements and relationships (Neo4j)
- Streaming artifacts (Kafka topic state snapshots)

The **workspace ID** (e.g., `scenario_2025_ath_eq001`) acts as a cross-system identifier for organizing, backing up, and restoring all related content.

BACKUP WORKFLOW BY USE CASE

When to Trigger a Backup

- After scenario execution completes successfully
- Before major updates to models, inputs, or system configuration
- On schedule (daily/weekly) for active or long-running scenarios

What to Backup (Per Workspace)

Table 24. Workspace Backup

Workspace Component	Description	Storage Mechanism
Scenario Metadata	Name, description, timestamps, creator, status	PostgreSQL
Input Datasets	Raw GeoJSON, CSV, NetCDF, etc.	MinIO
Simulation Outputs	Paths, damage zones, allocation summaries	MinIO, PostgreSQL
Database References	Relational data (infrastructure, stats)	PostgreSQL
Graph Elements	Neo4j subgraph: facilities, links, dependencies	Neo4j
Alerts & Logs	Logs, alerts triggered, resolution state	Elasticsearch / PostgreSQL
Stream Snapshots	Kafka topic segments (optional)	Kafka / external S3

Metadata Tagging

Each record, file, or object is tagged with the workspace ID, e.g.:

- PostgreSQL: `workspace_id` column
- MinIO: object prefix `workspace/scenario_2025_ath_eq001/`
- Neo4j: node property `workspace_id`
- Kafka: message headers (if needed)

BACKUP PROCESS

Step-by-Step Procedure

- **Initiate Backup**: Triggered manually or by scheduler/API (`POST /api/v1/backup/workspace/{id}`)
- **Collect References**: Query PostgreSQL to find all tables, MinIO keys, graph nodes, Kafka topics tagged with the workspace ID.
- **Snapshot and Export**:
  - PostgreSQL: Dump all records linked by `workspace_id`
  - MinIO: Copy or version all objects under the workspace prefix
  - Neo4j: Export graph subgraph (e.g., `apoc.export.json.query`)
  - Kafka: (optional) Export last N messages from associated topics
- **Store in Unified Archive**:
  - Package all assets into a single archive (ZIP, TAR.GZ, or logical manifest)

  - Optionally split by domain (data, model, logs, stream)
- **Store in Backup Repository**:
  - Upload to MinIO (`s3://backups/workspaces/scenario_2025_ath_eq001.tar.gz`)
  - Log backup metadata to PostgreSQL `workspace_backup_log`

RESTORE PROCESS

Restore Triggers

- Re-run historical scenario for audit or comparison
- Roll back to pre-update state
- Recover from data corruption or system failure

Step-by-Step Restore Procedure

- **Select Backup**: User selects a workspace backup via UI or API
- **Extract Backup Package**: Decompress archive or fetch manifest
- **Restore Components**:
  - **PostgreSQL**: Import SQL/JSON records for metadata, datasets, results
  - **MinIO**: Rehydrate objects to original prefix under `workspace_id`
  - **Neo4j**: Load subgraph from JSON/CSV export (`apoc.import.json`)
  - **Kafka**: Replay stream if needed, or restore offsets
- **Re-link Workspace**:
  - Set scenario status: "restored"
  - Ensure all indexes, references, and cache entries are rebuilt
- **Validate Integrity**:
  - Compare restored object counts
  - Run test queries (e.g., summary metrics, graph traversal)
  - Re-run critical business rules to ensure consistency

POLICIES

Retention Policy

- **Active Workspaces**: Daily incremental backup
- **Completed Workspaces**: Weekly cumulative, monthly full
- **Archived Workspaces**: Kept for 18–36 months (regulatory scope)

Validation Policy

- Restore operations are tested quarterly via automated staging restores
- Random checksum checks on MinIO + PostgreSQL datasets post-restore
- Broken or partial workspaces are flagged with status = “corrupted”

SUMMARY: KEY BENEFITS OF WORKSPACE-BASED APPROACH

- Cross-system consistency
- Scenario portability (move to test or training)
- Auditable restore process
- Aligned with real operational workflows
- Efficient long-term storage and comparison

6.6 AUDITING

The auditing system in PANTHEON links every scenario action to traceable and verifiable logs using the `workspace_id` model. It enhances trust, traceability, security, and compliance across all use cases, while integrating with the broader platform services via API, Kafka, and Web UI.

Auditing in the PANTHEON platform ensures full traceability, accountability, and compliance across all scenario executions and related data transformations. Given that the system is organized around **workspace-based execution units** (i.e., each use case scenario is encapsulated in a workspace), the auditing framework captures all interactions, events, and data changes tied to a given workspace. This enables retrospective analysis, forensic investigations, compliance reporting, and anomaly detection.

AUDITING SCOPE AND PRINCIPLES

The platform adheres to the following auditing principles:

- **Completeness**: All user actions, automated processes, data ingestions, and transformations are logged.
- **Contextualization**: Every audit entry is tagged with the relevant `workspace_id`.
- **Traceability**: Events form a causally-linked chain — from data input to simulation, output, and user actions.
- **Tamper-evidence**: Logs are immutable and cryptographically verifiable where required.
- **Accessibility**: Audit data is accessible via Web UI, API, and Kafka stream for integration.

AUDIT LOGGING COVERAGE

Table 25. Audit Content

Category	Logged Details
User Actions	Logins, role changes, data uploads, scenario executions, exports
Scenario Lifecycle	Creation, run, result generation, restore, backup
Data Access	Reads/writes from PostgreSQL, MinIO, Neo4j, API endpoints
Processing Events	Simulation triggers, component execution times, success/failure logs
API Calls	Endpoint name, method, parameters, status code, duration
Kafka Messages	Topic name, key, message metadata, sender ID (where applicable)
Authentication Events	Auth success/failure, token issuance/expiration (Keycloak integration)
Policy Triggers	Alert rules, risk thresholds, workspace rule violations

AUDIT LOG STRUCTURE

Each audit record is stored in the central **Audit Trail Service**, with persistence in **PostgreSQL** and optional streaming to **Kafka** for real-time monitoring.

```
{
  "timestamp": "2025-06-04T10:15:34Z",
  "workspace_id": "scenario_2025_ath_eq001",
  "user_id": "user_58",
  "event_type": "DATA_EXPORT",
  "component": "web-ui",
  "resource": "/api/v1/results/export",
  "status": "SUCCESS",
  "duration_ms": 743,
  "ip_address": "192.168.1.44",
  "metadata": {
    "format": "CSV",
    "dataset": "shortest_path",
    "records_exported": 612
  }
}
```

Audit entries are linked chronologically and indexed by `workspace_id`, `user_id`, `component`, and `event_type`.

WORKSPACE-LEVEL AUDIT SUMMARIES

Each scenario workspace has an **associated audit history** automatically displayed in the scenario's detail view in the Web UI or exported via:

- `GET /api/v1/audit/workspace/{workspace_id}`
- Kafka topic: `audit.workspace.{id}`

Audit summaries include:

- Start and end time of the scenario lifecycle
- Number of simulation runs
- Data sets uploaded, modified, or deleted
- Alerts triggered and user responses
- Backup and restore events
- Access patterns and download history

REAL-TIME AUDIT NOTIFICATIONS

The audit service streams key events to Kafka for external monitoring and automation:

Table 26. Kafka audit

Kafka Topic	Description
`audit.events`	All audit events (selectable filters)
`alerts.security.login.failures`	Multiple failed logins
`audit.workspace.{workspace_id}`	Events scoped to specific workspace
`alerts.anomaly.workspace.{id}`	Suspicious access or unexpected patterns

RETENTION AND COMPLIANCE

- **Retention Policy**:
  - 1 year for standard audit logs
  - 5+ years for critical scenario-related data (EU Civil Protection compliance)
- **Encryption**:
  - Logs stored encrypted at rest (PostgreSQL, MinIO).
- **Immutability**:
  - Write-once logs or WORM-style object locking in MinIO (optional).
- **Access Control**:
  - Only users with the role `auditor`, `admin`, or `scenario-owner` can view full logs.
  - Logs can be filtered/redacted for data protection compliance (e.g., GDPR).

EXAMPLE USE CASE WORKFLOWS

Example: Earthquake Scenario Execution Audit

- `user_32` uploads input files → logged with event `DATA_UPLOAD`
- Simulation runs → events `SIMULATION_START`, `SIMULATION_COMPLETE`
- Output downloaded → event `DATA_EXPORT`
- Alerts triggered on blocked roads → event `ALERT_TRIGGERED`
- Scenario backed up → `BACKUP_COMPLETED`, including MinIO object hashes

Example: Suspicious Access Detection

- At midnight, non-admin user attempts to download datasets from 5 unrelated workspaces
- `AUDIT_EVENT` type: `ACCESS_DENIED`, `DATA_DOWNLOAD`
- Audit service detects pattern and pushes alert to:
  a. Kafka: `alerts.anomaly.user_93`
  b. UI notification for admin

7. CONCLUSIONS

The secure data repository developed under Deliverable D7.3 serves as a critical enabler for the broader vision of the PANTHEON Community-Based Smart City Digital Twin Platform. Designed to support evidence-based decision-making, scenario simulation, real-time emergency coordination, and long-term resilience planning, this repository addresses the full spectrum of technical, operational, and security requirements necessary to manage complex, multi-hazard disaster response data in urban environments. Its modular and federated design, combined with strong adherence to interoperability and open standards, ensures it is well-positioned not only to serve the immediate needs of the PANTHEON use cases in Athens/Attica and Vienna, but also to scale and evolve as a pan-European resource for disaster risk management.

The system's strength lies in its ability to ingest, validate, store, and disseminate heterogeneous datasets from a wide variety of sources—authoritative (UN, ESA, OpenStreetMap), institutional (KEMEA, municipal governments), commercial (Google APIs, OpenWeather), and project-generated (UAVs, simulations, IoT sensors). By harmonizing these datasets using standardized schemas, metadata structures, and provenance tracking mechanisms, the repository ensures that data quality, relevance, and trustworthiness are maintained across all workflows. Each scenario—earthquake, wildfire, heatwave, and cyberattack—benefits from purpose-built ingestion and processing pipelines, allowing for granular control over data fidelity and update frequency, while enabling consistent integration into decision-support systems.

Furthermore, the use of scalable technologies such as PostgreSQL/PostGIS, MinIO, Neo4j, and Apache Kafka reflects a modern, cloud-native approach that balances performance with flexibility. These technologies not only support traditional storage and retrieval but also accommodate dynamic real-time data streams and graph-based querying—both of which are essential for understanding cascading infrastructure failures, optimizing resource allocation, and evaluating interdependencies across urban systems.

A defining feature of this repository is its robust security framework, built around the principles of confidentiality, integrity, and availability (CIA), as well as full alignment with European data protection regulations such as GDPR. Keycloak enables role-based identity and access management, supporting granular control of user privileges across organizational and operational boundaries. AES-256 encryption secures data at rest, while TLS ensures secure data transmission. All interactions with the repository are logged for auditability, enabling forensic analysis and compliance verification.

From a practical perspective, the repository enhances operational efficiency across multiple user roles. Emergency planners can access high-resolution demographic and infrastructure data to design evacuation strategies and allocate resources. First responders can receive real-time updates on blocked roads, affected zones, and UAV-detected damage to improve situational awareness. Researchers can retrieve historical datasets with documented provenance and apply analytical models for hazard simulation or vulnerability assessment. Meanwhile, decision-makers at the policy level can rely on trusted, validated datasets to design interventions and allocate funding where risk is highest.

The spatial and temporal granularity of the data—down to 100-meter resolution population grids and hourly weather streams—enables modelling with a high degree of fidelity. Importantly, the repository also supports longitudinal analysis, enabling stakeholders to assess how risks evolve over time or how different interventions (e.g., installation of cooling infrastructure, wildfire mitigation planning) perform under

changing climate or urban development conditions. The alignment with the Digital Twin paradigm ensures that the repository can support "what-if" scenarios, impact forecasting, and AI-driven decision augmentation.

In the broader context of European digital transformation and civil protection, this deliverable contributes to the goals of resilience, interoperability, and preparedness. The repository is designed not only to function within PANTHEON but also to interconnect with other EU platforms, such as those developed under the Horizon Europe, Digital Europe, and Civil Protection Mechanism programs. Its emphasis on open APIs, exportable data formats, and metadata standards ensures that it can participate in broader federated data spaces—such as the proposed European Green Deal Data Space or the Common European Data Space for Smart Communities.

From a governance and sustainability standpoint, the repository includes built-in support for maintenance and monitoring, ensuring its long-term usability. Backup procedures, logging infrastructure, and automated alerts allow for continuous monitoring of system health and data integrity. The inclusion of version control and data freshness mechanisms supports the goal of maintaining current and accurate data even during fast-evolving emergency scenarios. Combined with its modular deployment model (e.g., Docker Swarm), the system is easily portable across infrastructure environments, whether in research labs, municipal IT systems, or cloud-hosted crisis centers.

Looking forward, several paths for enhancement and expansion are clear. Integration with real-time citizen reporting apps and mobile data sources could help validate model outputs and improve responsiveness. Expansion of predictive analytics using AI/ML could transform the repository into a proactive alerting system, not just a reactive data warehouse. Federated learning and privacy-preserving analytics could be implemented to allow for cross-border or cross-agency collaboration without exposing sensitive data. Moreover, introducing multilingual semantic layers could enhance accessibility for diverse users and broaden adoption across Europe.

Educational and training materials will also play a vital role in maximizing the utility of the repository. The development of user guides, API documentation, sample workflows, and sandbox environments will allow stakeholders to explore the capabilities of the system and integrate it with their own decision-support environments. By promoting knowledge transfer and capacity building, the repository can extend its impact beyond technology into organizational resilience.

In conclusion, this deliverable demonstrates that a well-structured, secure, and interoperable data repository is not merely a back-end system but a vital strategic asset for modern urban risk governance. It provides a living infrastructure that enables collaboration, adaptation, and rapid response in the face of increasingly complex and interconnected hazards. Through PANTHEON, it becomes clear that managing disasters is not just about predicting events or deploying resources—it is about creating data ecosystems that empower communities to prepare, respond, and recover more effectively. The secure data repository is a critical step in realizing that vision.

8. LIST OF ABBREVIATIONS

Abbreviation	Meaning
GDPR	General Data Protection Regulation
API	Application Programming Interface
CSV	Comma Separated Values
JSON	JavaScript Object Notation
AI	Artificial Intelligence
ML	Machine Learning