

Article

Integrating Near Real-Time Hydrological Data for Monitoring and Alerting: The RoWaterAPI Framework

Mihnea Cristian Popa ¹, Daniel Constantin Diaconu ^{1,2,*}, Adrian Gabriel Simion ¹, Ioan Florin Voicu ^{1,3} and Costache Romulus ^{4,5,6}

- ¹ Interdisciplinary Center for Advanced Studies, University of Bucharest, 010041 Bucharest, Romania; popamihneacristian@gmail.com (M.C.P.); simion.adrian14@gmail.com (A.G.S.); ioan.voicu@s.unibuc.ro (I.F.V.)
- ² Department of Meteorology and Hydrology, Faculty of Geography, University of Bucharest, Bd. Nicolae Bălcescu No 1, 1st District, 010041 Bucharest, Romania
- ³ Interdisciplinary School of Doctoral Studies (ISDS), University of Bucharest, 031216 Bucharest, Romania
- ⁴ Research Institute of the University of Bucharest, 90-92 Sos. Panduri, 5th District, 050663 Bucharest, Romania; romulus.costache@unitbv.ro
- ⁵ National Institute of Hydrology and Water Management, București-Ploiești Road, 97E, 1st District, 013686 Bucharest, Romania
- ⁶ Department of Civil Engineering, Transilvania University of Brașov, 5, Turnului Street, 500152 Brașov, Romania
- * Correspondence: daniel.diaconu@unibuc.ro

Abstract

The paper addresses the limitations of fragmented and delayed hydrological information systems in supporting timely disaster risk mitigation. The paper introduces the RoWaterAPI, a framework that integrates near real-time hydrological measurements with geospatial analytics to improve awareness during flood-related events. The methodology utilizes open-source technologies, including Django, Kafka, and PostGIS, to support scalable data ingestion and hazard mapping. Initial baseline evaluation under a simulated bursty workload indicates an end-to-end latency of $\approx 1\text{--}3$ s and a peak throughput of $\approx 6000\text{--}8500$ messages/s. This performance supports real-time alerts for data variations, bridging advanced geoprocessing with user-centered design for public and institutional stakeholders. Ultimately, RoWaterAPI provides a transferable implementation model that can be adapted to any national context facing similar constraints in data fragmentation and operational accessibility.

Keywords: API; geospatial analytics; water monitoring; digital transformation



Academic Editor: Eungyu Park

Received: 23 December 2025

Revised: 1 February 2026

Accepted: 18 February 2026

Published: 19 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Sustainable water management has become an increasingly important concern as environmental pressures continue to intensify [1–3]. Ongoing climate changes are reshaping water availability and variability, highlighting the limitations of traditional management approaches and showing the need for more adaptive strategies.

Effective water management relies on the availability of timely and reliable hydrological data, particularly when quick decision-making is required during fast-onset events such as floods, and when sustained, adaptive management actions are needed during slow-onset events such as droughts, which typically require sustained, adaptive management decisions over a longer period of time [4]. It enables authorities and stakeholders to respond effectively, minimizing damage and optimizing water distribution. In the broader

perspective, it supports long-term planning and development strategies, ensuring water security and resilience against the impacts of climate change and population growth.

The scientific community recognizes the complexity of implementing effective hydrological monitoring systems, especially in areas like South and Southeast Asia, where diverse weather patterns impact agriculture and livelihoods. The integration of meteorological forecasts, downscaling methods, and hydrological models plays a crucial role in forecasting systems' accuracy. However, the effectiveness of these components can vary, highlighting the ongoing debate regarding optimal methodologies for hydrological monitoring. This conversation underscores the necessity for advanced, region-specific systems capable of addressing the unique challenges posed by weather variability and climate change [5–9].

The integration of environmental science and information technology through platforms like RoWaterAPI highlights an expanding field of study that bridges the gap between advanced computational techniques and practical environmental management strategies. RoWaterAPI is a research-driven framework developed within the Interdisciplinary Center for Advanced Studies, University of Bucharest, and is currently maintained and operated by the authors. The framework was initiated in response to recurring operational constraints in national hydrological information delivery, including fragmented data flows, limited interoperability, and the lack of programmatic access mechanisms that would allow hydrological information to be reliably integrated into connected decision-support and alerting systems. At present, publicly available hydrological information/data in Romania is available as bulletins and static reports (hydrological bulletins are published as documents for public access, while flood hazard and risk information is provided through a dedicated map portal). RoWaterAPI is proposed as an integration and access layer that exposes hydrological measurements and hazard information via standardized endpoints, complementing, rather than replacing existing monitoring infrastructures.

In general, operational hydrological management systems are part of public services, with governmental policies and guidelines influencing the field of interest. Recent trends show efforts to increase commitment to the sustainable management of water resources, disaster prevention and mitigation, as well as the need for integrated water resource management, as climate and social changes place increasing demands on resources.

Despite ongoing international efforts to modernize hydrological services, many national systems continue to rely on fragmented infrastructures and delayed data flows, limiting their effectiveness during emergency situations. This highlights the need for a modern, scalable, and accessible platform tailored to national needs.

Although several international standards such as WaterML 2.0, the OGC SensorThings API, and ISO-based Observations and Measurements (O&M) exist to support hydrological data interoperability, their practical implementation in national systems like Romania remains limited. Factors such as legacy infrastructure, inconsistent data formats, and insufficient integration capabilities often hinder adoption. This paper explores an alternative approach that aims to complement these standards while addressing operational realities.

The objective of this study is to design and evaluate RoWaterAPI, an event-driven hydrological monitoring and alerting platform that integrates near real-time hydrological observations with geospatial processing capabilities. The system is implemented using a modular microservices architecture to support scalable data ingestion, spatial analysis, and alert dissemination workflows.

RoWaterAPI is designed to support both institutional and public use cases by providing standardized access to hydrological data and warning information. Through this approach, the platform addresses operational requirements related to water resource management, early warning, and situational awareness during hydrological events.

Frameworks such as RoWaterAPI exemplify a novel integration in tackling these pressing challenges. This initiative, emerging against a backdrop of global demographic pressures and unpredictable climate phenomena, underscores the need for innovative approaches to hydrological monitoring and data accessibility.

However, existing approaches often prioritize either scientific modeling or visualization, while offering limited support for operational alerting and real-time data dissemination at a national scale.

By design, RoWaterAPI supports adaptation to different water management contexts and regulatory environments, providing transferability to similar hydrological information systems. This approach encourages sharing knowledge and working together internationally, improving water management, disaster readiness, and environmental protection.

Although developed in the context of Romania, the RoWaterAPI framework addresses challenges that are common to many national hydrological monitoring systems, including data fragmentation and limited real-time accessibility.

The remainder of this paper is structured as follows: Section 2 discusses the theoretical background and related work. Section 3 outlines the proposed methodology and technical architecture of RoWaterAPI. Section 4 presents the results, followed by a discussion in Section 5. Finally, Section 6 concludes the paper and outlines future work.

2. Theoretical Background and Related Work

An Application Programming Interface (API) is a set of protocols, routines, and tools for building software applications, acting as an intermediary that allows different software programs to communicate with each other [10,11].

The main characteristic of an API is its ability to define a set of protocols or tools designed to communicate with an operating system, a database, a network, and other lower-level aspects of a software system [12–14].

The benefits offered by an API include flexibility and efficiency in data access, customization based on user requirements, and code reusability to enable more productive workflows [14]. The transition from Simple Object Access Protocol (SOAP) to APIs, particularly RESTful APIs, was driven by the need for simplicity, flexibility, and performance in web services. While SOAP relied on rigid protocols and XML-based communication, APIs introduced lightweight formats like JSON and a more flexible, resource-oriented architecture. This shift allowed for faster data exchange, easier integration with modern applications, and greater adaptability to diverse platforms, making APIs the preferred choice for scalable and efficient web services.

Designed by Guido van Rossum and first released in 1991, Python 3.14's philosophy emphasizes code readability and syntax that allows programmers to express concepts in fewer lines of code compared to languages like C++ or Java. One of Python 3.14's key strengths lies in its vast ecosystem of third-party modules and frameworks, which extend its capabilities far beyond the standard library [15].

In the context of web-based and data-driven applications, frameworks such as Django enable the efficient development of robust and scalable services, while scientific libraries including NumPy, Pandas, Matplotlib, and Scikit-Learn support data analysis, visualization, and predictive modeling tasks. Additionally, Python 3.14's cross-platform allows deployment without significant changes needed, an important requirement for such infrastructures [16,17].

PostgreSQL, extended with PostGIS, serves as the core data storage layer, providing reliable management of large volumes of spatio-temporal hydrological data and supporting advanced spatial queries essential for national-scale water monitoring applications [18].

PostgreSQL, extended with the PostGIS spatial module, serves as the primary data storage layer within RoWaterAPI. This combination enables efficient management of large volumes of spatio-temporal hydrological data, supporting complex spatial queries required for analyzing river networks, water bodies, and flood-prone areas [19,20]. By adhering to Open Geospatial Consortium (OGC) standards, PostGIS ensures interoperability with external geospatial tools and services commonly used in hydrological analysis.

Over the past decade, several standards have been developed to support the interoperable exchange of hydrological data. Among these, WaterML 2.0, an Open Geospatial Consortium (OGC) standard based on the Observations and Measurements (O&M) model, provides a structured representation of hydrological time series data. The SensorThings API and OGC API—Connected Systems further define RESTful approaches for accessing and managing sensor observations. While these standards have been adopted in large-scale initiatives such as the US National Water Model and the UK Hydrology API, their implementation at the national level remains uneven, particularly in Eastern Europe.

Amazon Web Services is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS), and packaged software as a service (SaaS). AWSs can offer organizations tools such as computing power, database storage, and content delivery services. Its broad set of tools and services includes computing, storage, networking, database, analytics, application services, deployment, management, machine learning, mobile, developer tools, and tools for the Internet of Things (IoT), making it possible to lower costs and become more agile [21,22].

Amazon EC2 is a central part of AWSs, providing scalable computing capacity in the Amazon Web Services cloud. It is designed to make web-scale cloud computing easier for developers, allowing them to obtain and configure capacity with minimal effort. With Amazon EC2, we will host our applications and microservices on a scalable and secure cloud computing environment. EC2 provides the flexibility to choose the instance types, operating systems, and software packages that best match the needs of our applications, ensuring optimal performance and reliability. This setup allows us to efficiently manage and scale our microservices architecture in response to changes in demand, without worrying about the underlying hardware or the maintenance of physical servers. By leveraging EC2's capacity to dynamically adjust resources, we can ensure that our applications remain responsive and available, while also optimizing costs based on actual usage [21].

Amazon MSK (Managed Streaming for Apache Kafka) is a fully managed service that makes it easy to build and run applications that use Apache Kafka to process streaming data. MSK provides the infrastructure for Apache Kafka without requiring users to become experts in Kafka operations, making it easier to ingest and process streaming data in real time. It automates the provisioning, setup, scaling, and management of Apache Kafka clusters in the cloud, offering high availability and security. This service simplifies the process of setting up a Kafka cluster, monitoring cluster health, and scaling the number of messages that can be processed [23]. We will integrate Amazon MSK with the Kafka component of our application, leveraging the managed service to streamline our event-driven architecture. By utilizing Amazon MSK, we can connect our application's Kafka component directly to a fully managed Kafka environment. Through this setup, we ensure that our Kafka component is optimally managed, allowing us to focus more on application development and less on the operational overhead associated with running Kafka clusters.

Amazon RDS (Relational Database Service) is a managed relational database service that provides six popular database engines, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. RDS makes it easier to set up, operate, and scale a relational database in the cloud by handling routine database tasks such as

provisioning, patching, backup, recovery, failure detection, and repair. Amazon RDS offers cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups. It frees developers to focus on their applications by managing the database infrastructure, providing a high level of availability, security, and compatibility with existing tools and applications [24].

3. Proposed Methodology and System Architecture

RoWaterAPI is built on an event-driven microservices architecture designed to support the integration and near real-time processing of hydrological data. By decoupling data ingestion, processing, and alert dissemination, this architecture enables scalable handling of heterogeneous data streams and supports low-latency propagation of hydrological observations and warning events.

Event-driven architectures (EDAs) are particularly suited to hydrological monitoring systems, where rapidly changing environmental conditions require timely system responses. In RoWaterAPI, the use of EDA facilitates dynamic processing of incoming data and supports adaptive system behavior under variable data loads, contributing to operational scalability and reliability [25,26].

The proposed architecture was designed to address two operational constraints commonly identified in hydrological early warning systems: data latency and limited scalability during extreme hydrological events. By decoupling data ingestion, processing, and alert generation, the system enables faster propagation of critical alerts to decision-makers (Figure 1).

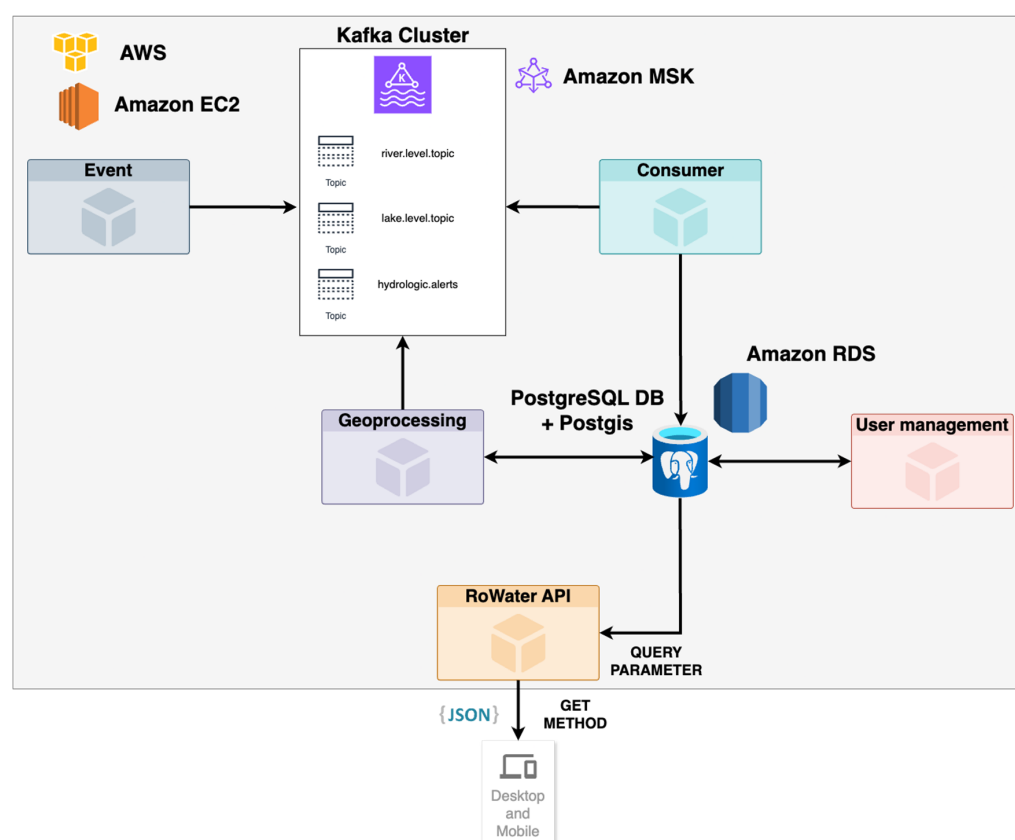


Figure 1. Architecture diagram.

Within RoWaterAPI, the API layer functions as the primary interface for accessing hydrological observations and alert information, supporting consistent integration across

institutional and public-facing applications. This approach facilitates near real-time access to hydrological data required for monitoring, analysis, and early warning workflows.

By formalizing data exchange through well-defined endpoints and access controls, the API supports broad yet controlled availability of hydrological information to multiple stakeholder groups, including governmental agencies, environmental organizations, and the public. Such standardization enhances interoperability and reliability, which are essential for operational water resource management and environmental decision-making.

The backend implementation of RoWaterAPI is based on Python 3.14, utilizing the Django and GeoDjango frameworks to support web-based service development and geospatial data handling. Django provides a stable application framework for implementing API endpoints and data access logic, while its object-relational mapping capabilities support structured interaction with the underlying spatial database [27,28].

Django is used within RoWaterAPI as the core web application framework for implementing API endpoints and coordinating backend services. Its structured architecture supports modular development and simplifies the integration of data access, authentication, and request handling within a unified application layer.

The framework's object-relational mapping (ORM) capabilities enable consistent interaction with the underlying relational and spatial database, supporting structured data access while reducing direct dependency on database-specific query implementations. This approach contributes to maintainability and scalability, which are essential for operational hydrological monitoring systems deployed at a national scale.

Python 3.14 was chosen due to its mature geospatial ecosystem and its suitability for rapid prototyping of data-driven workflows.

The RoWaterAPI platform leverages Python 3.14, Django, and GeoDjango to support scalable data ingestion, geospatial processing, and API-based data dissemination. Python 3.14's mature geospatial ecosystem facilitates the implementation of complex data processing workflows, while Django provides a stable and extensible framework for building web-based hydrological services. GeoDjango extends this framework with native support for spatial data models and geospatial querying, enabling the integration of hydrological observations with spatial analyses required for monitoring and alerting applications. Through its integration with GEOS, GDAL, and PostGIS, the platform supports spatial indexing and efficient geospatial queries within a relational database environment, ensuring consistent handling of vector-based hydrological data [16].

To support near real-time data ingestion and alert dissemination, RoWaterAPI employs Apache Kafka as an event-streaming backbone. Kafka enables asynchronous communication between system components, allowing hydrological observations and warning events to be processed and distributed with low latency [29–31]. This event-driven approach is well-suited to dynamic hydrological monitoring contexts, where rapid changes in water conditions require the timely propagation of alerts to downstream services and end users [6,32].

In its current implementation, RoWaterAPI relies on a lightweight JSON-based schema optimized for operational performance and system integration, while remaining extensible toward WaterML 2.0 and SensorThings API mappings in future development stages.

To ensure that no hydrological data is lost during the ingestion of parsed reports, RoWaterAPI utilizes an at-least-once delivery strategy. The Kafka consumers are configured with manual offset commits, meaning a record is only acknowledged in the event stream after it has been successfully persisted into the Amazon RDS spatial database. This reduces the risk of data loss under bursty workloads, at the cost of possible reprocessing; therefore, database writes are designed to be idempotent (e.g., via unique event identifiers and upserts) to prevent duplicates.

Compared to existing hydrological information systems, RoWaterAPI adopts a modular, event-driven architecture that prioritizes operational deployment at a national scale, extensibility, and public accessibility. Whereas many existing solutions focus primarily on scientific modeling or data visualization, RoWaterAPI integrates real-time alerting, geospatial processing, and API-based data access within a unified, production-oriented framework adapted to the constraints of Romanian hydrological infrastructure.

Design Goals and Data Model

The system design was guided by requirements for low-latency access, scalability, and modularity. RoWaterAPI is built around a simplified data model comprising three core entities: hydrometric stations, time-series measurements (including water level, discharge, and water quality), and warning events derived from hydrological and meteorological nowcasting services. This structure supports both the ingestion of raw data from field monitoring devices and the generation of post-processed geospatial outputs for analysis and visualization.

The internal data schema was developed to accommodate heterogeneous data flows while maintaining consistency across spatial and temporal representations. The API follows RESTful design principles, enabling standard GET and POST operations with JSON-based payloads. Although the current implementation is not fully compliant with WaterML 2.0, the data model has been designed to allow future extension or transformation into standardized hydrological data formats.

Access control is implemented through token-based authentication, whereby authenticated clients receive signed JSON Web Tokens (JWTs) containing identity and role-based claims. These claims are validated at the API gateway level to enforce differentiated access policies for administrators, institutional users, and public users. This approach enables fine-grained control over data visibility and permitted operations, addressing common limitations in hydrological information systems related to access filtering and accountability. RoWaterAPI is designed to serve multiple stakeholder categories through differentiated access policies and information products. In this paper, we distinguish the general public, water users, and water management authorities (Table 1).

Table 1. Stakeholder categories and degree of detail provided by RoWaterAPI.

Category	Description
General public	General hazard information (read-only) Map-based visualization Active warnings/events
Water users	More detailed station-level information (authenticated) Time-series access Metadata Data export
Water management authorities	Highest detail (authenticated) Full operational views Administrative functions Data management data

The general public is provided with simplified, read-only hazard awareness information, such as the current status of hydrometric stations and active warning events, primarily through interactive maps and dashboards. Authenticated operational water users can access a higher level of detail (station-level time series, metadata, and export capabili-

ties) for monitoring and decision-support workflows. Water management authorities are provided with the highest level of detail and operational capabilities, including access to administrative and data management functions.

The system architecture illustrates how such design choices can support more efficient operational workflows across hydrological monitoring, emergency response, and territorial planning contexts. In the following sections, the functional components of RoWaterAPI are described, including its real-time data access mechanisms, geoprocessing capabilities for spatial analysis, and tools for flood hazard mapping and assessment. In its current implementation, RoWaterAPI provides hazard-related information. Full flood risk assessment would require the integration of exposure and vulnerability layers, which are outside the scope of this work and are considered future work.

This system leverages event-driven microservices architecture, integrating near real-time hydrological data, advanced geoprocessing techniques, and interactive user interfaces to provide comprehensive water monitoring and alert solutions. The demonstration model for RoWaterAPI includes a comprehensive platform that features:

- An API for near real-time hydrological data, facilitating access to water levels, flow rates, quality measurements, and other important hydrological variables from a network of hydrometric stations across the region (Figure 2).
- A geoprocessing microservice, capable of generating data for interactive maps and dashboards that underscore flood risks and hydrological warnings, thus improving the decision-making process for both authorities and the general population.
- A user-friendly interface, specifically designed for the beneficiary institutions, equipped with advanced data management tools and enhanced security measures to safeguard sensitive information.

```

{
  "meta": {
    "status": 200,
    "author": "National Administration Romanian Waters",
    "license": "OGL-ROU 1.0",
    "version": "1.0"
  },
  "data": [
    {
      "watercourse": "Lotru",
      "lake": "Vidra",
      "measurement_period": "25.06.2024 16:57:42",
      "normal_retention_volume": 340.3,
      "variation": 50,
      "forecast_description": "Decreasing level",
      "trend": "Decreasing"
    },
    {
      "watercourse": "Lotru",
      "lake": "Vidra",
      "measurement_period": "29.06.2024 13:24:13",
      "normal_retention_volume": 353.5,
      "variation": 6,
      "forecast_description": "Decreasing level",
      "trend": "Decreasing"
    }
  ]
}

```

```

{
  "meta": {
    "status": 200,
    "author": "National Administration Romanian Waters",
    "license": "OGL-ROU 1.0",
    "version": "1.0"
  },
  "data": [
    {
      "station": "Danube-br.Borcea",
      "station_code": 2,
      "watercourse": "Danube",
      "measurement_period": "25.06.2024 16:57:42",
      "attention_level": 550.0,
      "flood_level": 620.0,
      "danger_level": 750.0,
      "qa_flow_at_attention_level": 11680.0,
      "qa_flow_at_flood_level": 13080.0,
      "qa_flow_at_danger_level": 16000.0,
      "water_level": 106.0,
      "flow_rate": 4600.0,
      "variation": 375,
      "forecast_description": "Decreasing flow",
      "trend": "Decreasing"
    }
  ]
}

```

Figure 2. API response example.

RoWaterAPI introduces a novel approach to hydrological monitoring by integrating real-time data acquisition with advanced processing and user-focused design. This system distinguishes itself from existing national and international frameworks by offering:

- **Advanced Geoprocessing Capabilities:** Employing advanced geospatial analysis for enhanced flood prediction and hazard assessment, RoWaterAPI delivers more precise and actionable insights compared to traditional monitoring systems.
- **Accessible and Interactive Interfaces:** With modern web technologies, RoWaterAPI makes hydrological information and alerts accessible to a wider audience, including those without specialized knowledge.
- **Real-Time Data Integration:** RoWaterAPI provides a dynamic and responsive monitoring solution by ensuring the availability of near real-time hydrological data.

Following the description of the backend architecture, this section addresses the design and implementation of the user interface of RoWaterAPI. The focus is placed on how the front-end layer enables user interaction with hydrological data and geospatial outputs generated by the system.

The user interface was designed to support intuitive access to hydrological information while maintaining consistency with the underlying real-time and geospatial data structures. The front-end implementation is based on a component-oriented web architecture, facilitating maintainability, type safety, and structured interaction with the API.

The front-end layer of RoWaterAPI is implemented using React and TypeScript within a component-based web application architecture. This approach supports modular development, type safety, and long-term maintainability of the user interface, while enabling structured interaction with backend services. The use of strongly typed components facilitates more predictable data handling and reduces implementation errors in data-intensive applications [33,34].

The user interface design emphasizes interactive visualization of real-time hydrological data and derived geospatial outputs, allowing users to explore measurements and risk-related information without page reloads. React's component-oriented model supports efficient rendering and dynamic updates of map-based visualizations, which is particularly relevant for real-time and event-driven applications [35,36]. This design aligns the front-end layer with the system's real-time processing and geospatial analytics capabilities.

The interface is being developed to accommodate a diverse range of users, from hydrology experts and governmental bodies to the public, by offering data management panels and data filtering options (Figure 3).

The solution incorporates security mechanisms, including role-based authentication and data encryption, to protect sensitive information and regulate access to different data layers based on user roles. In addition, the system provides data management functionalities that allow authorized users to upload, modify, and remove datasets in a controlled and secure environment, ensuring data integrity and accountability.

Interactive map visualization is implemented using the Leaflet library, which provides a flexible and lightweight framework for web-based geospatial representation. Leaflet is an open-source JavaScript library widely used for developing interactive maps and has been adopted in numerous geospatial applications due to its efficiency and extensibility [37–39]. Its integration enables the visualization and manipulation of spatial datasets relevant to hydrological analysis, including river networks, hydrometric station locations, water levels, and flood-prone areas.

The mapping component supports the overlay of dynamic, near real-time data layers, such as flood warnings and hydrological alerts, providing an up-to-date spatial overview of evolving conditions. Generated maps allow direct user interaction through standard navigation functions, including zooming and panning, as well as thematic highlighting of areas based on selected criteria such as flood hazard/warning levels or water quality indicators. Users may further customize the visualization by enabling or disabling specific

data layers and selecting alternative base map styles, supporting task-oriented exploration of hydrological information.

Leaflet is characterized by a lightweight core architecture, with a compact JavaScript footprint that minimizes the impact on page load times while supporting interactive map-based applications. Despite its relatively small size, the library provides a broad range of functionalities, including tile layering, vector data representation, markers, popups, and standard user interactions such as zooming and panning. Its application programming interface is designed to be platform-independent, supporting consistent behavior across major desktop and mobile environments.

RoWaterAPI

Visualize data

Manage data

Settings

Search data

Hydrologic data

Hydrologic station Watershed Date

Enter hydrologic station name Enter watershed name Enter date

☐	Id	Watershed	Hydrologic station	Water level (cm)	Trend	Date
☐	1	Danube	Bratul Borcea	314	Rising	18.03.2024
☐	2	Olt	Hoghiz	87	Decreasing	18.03.2024
☐	3	Danube	Calafat	344	Rising	18.03.2024
☐	4	Mures	Alba Iulia	147	Normal	18.03.2024
☐	5	Arges	Malu Spart	108	Rising	18.03.2024
☐	6	Bahlui	Iasi	84	Normal	19.03.2024
☐	7	Siret	Lespezi	42.3	Decreasing	19.03.2024
☐	8	Jiu	Podari	139	Normal	19.03.2024
☐	9	Prut	Ungheni	87.3	Normal	19.03.2024

Figure 3. Data management interface design.

In addition to its core functionality, Leaflet supports an extensible plugin ecosystem that enables advanced capabilities such as geocoding, geolocation, and enhanced graphical visualization of spatial data. These features make it suitable for data-intensive geospatial applications requiring flexibility and modularity [39].

The RoWaterAPI application ecosystem is deployed on cloud infrastructure provided by Amazon Web Services (AWSs). This deployment supports scalability, availability, and performance requirements by enabling elastic resource allocation and reliable service operation under variable usage conditions.

We will use Amazon RDS as the relational database backbone for our application, taking advantage of its managed service to handle our database operations with enhanced efficiency and reliability. Our application will connect to RDS for all data storage, retrieval, and management tasks, leveraging the service's performance to ensure fast and reliable access to our data. The use of Amazon RDS will significantly streamline our development

process, reduce operational costs, and increase the overall stability and scalability of our application's data layer.

4. Results

The framework was validated using a heterogeneous dataset sourced from multiple hydrometric stations across several river basins in Romania, and the test data covered various monitoring periods throughout 2024. Given the operational constraints of current reporting, the system was evaluated by parsing legacy formats, specifically Word, Excel and PDF, which were transformed into structured JSON payloads.

The results obtained from the current implementation of RoWaterAPI indicate that the system can effectively ingest, aggregate, and process near real-time hydrological data from multiple sources. Performance tests and initial analyses demonstrate the system's capacity to handle substantial data volumes while maintaining low-latency access to hydrological measurements and derived geospatial outputs. Under simulated stress workload with small payload messages and no per-message geospatial transformations, the solution exhibited an average end-to-end ingestion-to-API persistence latency on the order of seconds (approximately 1–3 s) and sustained peak ingestion rates on the order of thousands of messages per second (approximately 6000–8500 messages per second). These results suggest the event-driven pipeline can handle bursty workloads without becoming the dominant processing bottleneck. These results reflect the baseline ingestion-and-access path. For larger payloads and geospatial processing, RoWaterAPI is designed to offload computationally intensive tasks to dedicated microservices, preserving low-latency ingestion while treating geoprocessing as a scalable downstream stage.

The integration of heterogeneous datasets from multiple hydrometric stations enables the generation of event-based alerts for critical hydrological conditions, including flood events and abrupt variations in water quality parameters. These alerts are produced in near real time and are delivered through the API and visualization components, supporting timely access to relevant information for institutional users and other stakeholders.

The results further highlight the role of the methodological framework in translating complex hydrological data into operationally usable outputs. In particular, the structure of the API responses and the design of the user interface contribute to streamlined data delivery and facilitate interaction with spatial and temporal information. As illustrated in Figures 2 and 3, these components support the transformation of raw measurements into interpretable visual and analytical representations, bridging the gap between data acquisition and practical application.

The API response structure illustrated in Figure 2 demonstrates how RoWaterAPI standardizes the delivery of near real-time hydrological data. The structure is designed to support consistent representation of heterogeneous data types, including water level measurements, discharge values, and water quality indicators collected from hydrometric stations. Data are provided in a JSON-based format, enabling straightforward integration with external applications and analytical workflows used by researchers, developers, and institutional stakeholders.

This response structure facilitates interoperability by allowing hydrological data to be consumed across different platforms and systems without additional transformation. In addition, the streamlined organization of the response supports efficient data transfer, contributing to reduced latency and timely access to information during hydrological events that require rapid interpretation and response.

The interface design presented in Figure 3 illustrates how users interact with hydrological data through the RoWaterAPI platform. The interface provides functionalities for filtering, editing, and visualizing data directly within the application, supporting struc-

tured interaction with complex hydrological datasets. These functionalities are intended to reduce workflow complexity and facilitate access to both spatial and temporal information.

Key interface features include dynamic data filtering, real-time map-based visualization, and configurable dashboard elements that provide an integrated overview of current hydrological conditions. The design supports scalability by allowing the incorporation of additional data streams and analytical components, enabling adaptation to different hydrological monitoring contexts.

Overall, the interface design demonstrates how usability and responsiveness can be aligned with advanced geospatial analysis, supporting the practical interpretation of hydrological information in operational and decision-support settings.

In summary, the results show that RoWaterAPI can ingest and process near real-time hydrological data from multiple sources, deliver standardized API responses with low latency, and support interactive visualization and event-based alerting through the user interface, as illustrated in Figures 2 and 3. These results confirm that the implemented architecture enables the coordinated handling of data ingestion, processing, and visualization within a single operational system.

5. Discussion

RoWaterAPI illustrates an integrated approach to hydrological monitoring that leverages event-driven system design and advanced geospatial analysis to support the timely use of hydrological information. By combining real-time data processing with spatial analytics, the system contributes to improved situational awareness in water resource monitoring contexts.

This approach enables a more responsive framework for linking data acquisition with operational outputs, supporting the translation of hydrological measurements into information relevant for monitoring and early intervention in water-related events.

The modular and scalable architecture of RoWaterAPI creates opportunities to incorporate methodologies validated in other domains of geospatial research, broadening its applicability beyond hydrological monitoring. By integrating techniques such as predictive analytics, machine learning models for risk assessment, and geospatial data fusion, the platform can enhance its functionality to address challenges in fields like urban planning, agricultural resource management, and environmental conservation.

With its flexible design and modular structure, RoWaterAPI presents a potential model for addressing the global challenge of sustainably managing water resources amidst environmental changes, though it remains a participant among various efforts to bridge gaps in existing water management approaches.

Comparable hydrological information systems have been implemented in several countries, illustrating the role of centralized and accessible data services in water resource management. In the United Kingdom, for example, the Hydrology API provides access to a wide range of hydrological datasets, including river flows, water levels, water quality parameters, rainfall, and groundwater conditions.

Similarly, hydrological services in countries such as the United States and Norway provide operational access to critical monitoring data that support water management activities, environmental protection, and disaster preparedness. These examples highlight the growing reliance on integrated hydrological data platforms for operational and decision-support purposes. These examples highlight the global demand for innovative and adaptable hydrological monitoring systems to ensure sustainable water management in the face of environmental challenges and climate variability.

The RoWaterAPI, aimed at improving hydrology data management and accessibility, took a significant step forward with its introduction at the NISPAce 2022 conference. This

opportunity allowed us to share our work and gather first-hand feedback, laying a solid foundation for the project's future development.

The adoption of RoWaterAPI also highlights the potential of integrated hydrological information systems to support collaborative approaches to water resource management across institutional and societal levels. By emphasizing data accessibility and structured user interaction, such systems can contribute to broader engagement with hydrological information and support more informed use of data in operational and planning contexts.

From a longer-term perspective, the modular and transferable design of RoWaterAPI suggests its applicability beyond a single national context. The framework provides a reference model that could be adapted to different institutional settings and hydrological regimes, offering insights relevant to the development of comparable initiatives in other regions.

These findings highlight the role of continued technological development and interdisciplinary cooperation in maintaining the effectiveness and adaptability of hydrological information systems under changing environmental conditions.

6. Conclusions

While the platform demonstrates the feasibility of real-time hydrological data integration, its operational effectiveness depends on data availability, institutional adoption, and governance frameworks. Through its innovative integration of event-driven architecture, sophisticated geospatial analysis, and the utilization of modern technologies such as Python 3.14, Django/GeoDjango, PostGIS, and Apache Kafka, RoWaterAPI will bring substantial improvements over traditional hydrological information dissemination workflows by providing standardized, near real time data access to critical information on top of existing monitoring infrastructures. It offers a scalable and adaptable solution that not only addresses current environmental and water management challenges but also lays the groundwork for future advancements. The research underscores the importance of technological innovation and interdisciplinary collaboration in tackling global environmental issues. As we move forward, continued exploration, technological integration, and expansion into diverse ecological regions remain crucial. RoWaterAPI's journey from concept to implementation exemplifies the transformative potential of combining environmental science with information technology, offering a template for future initiatives aimed at sustainable environmental stewardship and proactive water resource management.

Furthermore, RoWaterAPI is poised to significantly contribute to the digitalization process in Romania, exemplifying how technological advancements can streamline and enhance environmental and water resource management. This initiative not only serves as a pivotal step towards modernizing Romania's hydrological information access and decision-support workflows, by providing standardized, low-latency access to observations and derived data. Through RoWaterAPI illustrates the global potential for digital solutions to drive progress in sustainable water management and environmental protection.

RoWaterAPI's use of real-time data delivers significant advantages across various domains. In water management, it enhances decision-making processes, while strengthening disaster management through early warning systems for floods and droughts. In agriculture, it supports precision farming by providing timely hydrological data to optimize irrigation and crop planning. Furthermore, in environmental protection, it facilitates ecosystem health monitoring and prompt interventions. As one of the few tools to comprehensively integrate real-time hydrological data, it serves as a crucial resource for tackling interdisciplinary challenges, underscoring the profound impact of environmental data on decision-making across diverse fields.

Compared to traditional hydrological systems based on static datasets and delayed reporting, the RoWaterAPI approach demonstrates an alternative workflow centered on real-time processing and geospatial analysis.

This work also sets a precedent for leveraging technologies like Django and Apache Kafka in building robust, real-time environmental data integration and dissemination services (API-first), which complement existing systems and support early warning and operation decision-making. The emphasis on democratizing data access—making hydrological information available to a wide range of stakeholders, from governmental agencies to local communities—represents a significant step towards fostering a culture of informed decision-making and proactive environmental stewardship.

The outcomes of this study substantiate the research objective, demonstrating the successful integration of real-time hydrological data processing with advanced geospatial analysis. This accomplishment highlights the system's capacity to serve as an innovative and robust framework for sustainable water resource management and improved disaster preparedness, thereby fulfilling the aims outlined in the research.

These findings validate the core hypothesis that a modular, event-driven architecture can effectively support real-time hydrological monitoring and alerting, even in settings where legacy infrastructure limits the adoption of existing standards. While RoWaterAPI does not fully implement protocols such as WaterML 2.0 or the OGC SensorThings API, its architecture is designed to remain extensible and adaptable to such integrations in future iterations. The system's focus on usability, scalability, and performance reflects a practical response to the challenges identified at the beginning of this paper.

The paper contributes to a functional, extensible, and field-tested API architecture that complements existing research on hydrological data sharing, bridging the gap between theoretical standards and operational reality. RoWaterAPI stands as a practical model for real-time information access, built on top of existing monitoring systems and capabilities and adaptable to national and regional contexts where legacy infrastructure prevails.

To support research reproducibility, the authors are committed to releasing the RoWaterAPI source code under an open-source license. The codebase is currently undergoing final documentation, dependency cleanup, and security hardening to ensure a safe public release. This phased release strategy ensures that the framework, which has already demonstrated stable performance in baseline evaluations, is packaged in a maintainable and stable form for broader reuse. It should be noted that operational deployment requires formal inter-institutional agreements for matters such as data-sharing arrangements, cybersecurity review, and maintenance responsibilities; thus, institutional validation has not yet been initiated and will be pursued through forthcoming consultations with the National Administration "Romanian Waters"; therefore, the RoWaterAPI is not yet implemented within Romania's water resources management and operational systems. To support potential adoption, the authors plan a series of meetings with representatives of the National Administration "Romanian Waters" to propose a pilot evaluation and, subject to institutional approval, to explore integration pathways into their systems. In the interim, the architectural designs, message formats, and data schemas detailed in this study provide sufficient information to support conceptual replication and independent re-implementation.

The hydrological datasets used for validation remain proprietary to the National Administration of Romanian Waters and cannot be redistributed by the authors.

Author Contributions: Conceptualization, M.C.P., D.C.D. and C.R.; methodology, M.C.P. and C.R.; software, M.C.P.; validation, M.C.P., C.R., A.G.S. and I.F.V.; formal analysis, M.C.P., D.C.D., C.R., I.F.V. and A.G.S.; investigation, D.C.D.; resources, M.C.P.; data curation, M.C.P., C.R., A.G.S. and I.F.V.;

writing—original draft preparation, M.C.P. and D.C.D.; visualization, M.C.P.; supervision, D.C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This work is dedicated to the memory of Vișan Mircea Cristian, in recognition of his influence and support as a mentor and colleague throughout the authors' academic journey.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Behzadian, K.; Ferdowsi, A.; Zolghadr-Asli, B. (Eds.) *Planning and Management for Sustainable Water Resources and Infrastructure under Climate Change*; Elsevier: Amsterdam, The Netherlands, 2025; ISBN 978-0-443-33687-4.
- Goonetilleke, A.; Vithanage, M. Water Resources Management: Innovation and Challenges in a Changing World. *Water* **2017**, *9*, 281. [\[CrossRef\]](#)
- Cosgrove, W.J.; Loucks, D.P. Water Management: Current and Future Challenges and Research Directions. *Water Resour. Res.* **2015**, *51*, 4823–4839. [\[CrossRef\]](#)
- Sofianopoulos, S.; Faka, A.; Chalkias, C. SDI-Enabled Smart Governance: A Review (2015–2025) of IoT, AI and Geospatial Technologies—Applications and Challenges. *Land* **2025**, *14*, 1399. [\[CrossRef\]](#)
- Zhou, Y.; Zaitchik, B.F.; Kumar, S.V.; Arsenault, K.R.; Matin, M.A.; Qamer, F.M.; Zamora, R.A.; Shakya, K. Developing a Hydrological Monitoring and Sub-Seasonal to Seasonal Forecasting System for South and Southeast Asian River Basins. *Hydrol. Earth Syst. Sci.* **2021**, *25*, 41–61. [\[CrossRef\]](#)
- Drost, S.; Vogt, A.; Danowski-Buhren, C.; Jirka, S.; Kirstein, V.; Pakzad, K.; Rieke, M. WaCoDiS: Automated Earth Observation Data Processing within an Event-Driven Architecture for Water Monitoring. *Comput. Geosci.* **2022**, *159*, 105003. [\[CrossRef\]](#)
- Panchanathan, A.; Ahrari, A.; Ghag, K.S.; Mustafa, S.; Haghighi, A.T.; Kløve, B.; Oussalah, M. An Overview of Approaches for Reducing Uncertainties in Hydrological Forecasting: Progress and Challenges. *Earth Sci. Rev.* **2024**, *258*, 104956. [\[CrossRef\]](#)
- Kumar, Y.Y.; Srinivasulu, V. A Review on Hazard Assessment Framework About Flood and Drought Risk Management. *Geol. J.* **2025**. [\[CrossRef\]](#)
- Pechlivanidis, I.G.; Du, Y.; Bennett, J.; Boucher, M.-A.; Chang, A.Y.Y.; Crochemore, L.; Dasgupta, A.; Di Baldassarre, G.; Luterbacher, J.; Pappenberger, F.; et al. Enhancing Research-to-Operations in Hydrological Forecasting: Innovations across Scales and Horizons. *Bull. Am. Meteorol. Soc.* **2025**, *106*, E894–E919. [\[CrossRef\]](#)
- Overeem, M.; Mathijssen, M.; Jansen, S. API-m-FAMM: A Focus Area Maturity Model for API Management. *Inf. Softw. Technol.* **2022**, *147*, 106890. [\[CrossRef\]](#)
- White, C.T.; Petrasova, A.; Petras, V.; Tateosian, L.G.; Vukomanovic, J.; Mitsova, H.; Meentemeyer, R.K. An Open-Source Platform for Geospatial Participatory Modeling in the Cloud. *Environ. Model. Softw.* **2023**, *167*, 105767. [\[CrossRef\]](#)
- Reddy, M. *API Design for C++*; Elsevier: Amsterdam, The Netherlands, 2011; ISBN 978-0-12-385003-4.
- Neigh, C.S.R.; Tucker, C.J.; Carroll, M.L.; Montesano, P.M.; Slayback, D.A.; Wooten, M.R.; Lyapustin, A.I.; Shean, D.E.; Alexandrov, O.; Macander, M.J. An API for Spaceborne Sub-Meter Resolution Products for Earth Science. In Proceedings of the IGARSS 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: New York, NY, USA, 2019; pp. 5397–5400.
- Choi, Y.-D.; Goodall, J.L.; Sadler, J.M.; Castronova, A.M.; Bennett, A.; Li, Z.; Nijssen, B.; Wang, S.; Clark, M.P.; Ames, D.P.; et al. Toward Open and Reproducible Environmental Modeling by Integrating Online Data Repositories, Computational Environments, and Model Application Programming Interfaces. *Environ. Model. Softw.* **2021**, *135*, 104888. [\[CrossRef\]](#)
- Rossum, G.V.; Drake, F.L. *The Python Language Reference*; Drake, F.L., Ed.; Python Documentation Manual/Guido van Rossum; Release 3.0.1; Python Software Foundation: Hampton, NH, USA, 2010; ISBN 978-1-4414-1269-0.
- Abrate, M.; Bacciu, C.; Hast, A.; Marchetti, A.; Minutoli, S.; Tesconi, M. GeoMemories—A Platform for Visualizing Historical, Environmental and Geospatial Changes in the Italian Landscape. *ISPRS Int. J. Geo Inf.* **2013**, *2*, 432–455. [\[CrossRef\]](#)
- Rieke, M.; Bigagli, L.; Herle, S.; Jirka, S.; Kotsev, A.; Liebig, T.; Malewski, C.; Paschke, T.; Stasch, C. Geospatial IoT—The Need for Event-Driven Architectures in Contemporary Spatial Data Infrastructures. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 385. [\[CrossRef\]](#)
- Magno, R.; De Filippis, T.; Di Giuseppe, E.; Pasqui, M.; Rocchi, L.; Gozzini, B. Semi-Automatic Operational Service for Drought Monitoring and Forecasting in the Tuscany Region. *Geosciences* **2018**, *8*, 49. [\[CrossRef\]](#)
- Ilba, M. Parallel Algorithm for Improving the Performance of Spatial Queries in SQL: The Use Cases of SQLite/Spatialite and PostgreSQL/PostGIS Databases. *Comput. Geosci.* **2021**, *155*, 104840. [\[CrossRef\]](#)
- Li, W. GeoAI: Where Machine Learning and Big Data Converge in GIScience. *J. Spat. Inf. Sci.* **2020**, *20*, 71–77. [\[CrossRef\]](#)

21. Vanmechelen, K.; De Munck, S.; Broeckhove, J. Conservative Distributed Discrete Event Simulation on Amazon EC2. In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), Ottawa, ON, Canada, 13–16 May 2012; IEEE: New York, NY, USA, 2012; pp. 853–860.
22. Hofmann, W.; Lang, S.; Reichardt, P.; Reggelin, T. A Brief Introduction to Deploy Amazon Web Services for Online Discrete-Event Simulation. *Procedia Comput. Sci.* **2022**, *200*, 386–393. [\[CrossRef\]](#)
23. Get Started Using Amazon MSK—Amazon Managed Streaming for Apache Kafka. Available online: <https://docs.aws.amazon.com/msk/latest/developerguide/getting-started.html> (accessed on 14 July 2025).
24. Lakshmi, N.G. Database Migration on Premises to AWS RDS. *EAI Endorsed Trans. Cloud Syst.* **2018**, *3*, 154463. [\[CrossRef\]](#)
25. Mo, M.; Qiu, X.; Shen, A. An IoT-Based System for Water Resources Monitoring and Management. In Proceedings of the 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2015; IEEE: New York, NY, USA, 2015; pp. 365–368.
26. Rosa-Bilbao, J.; Boubeta-Puig, J.; Rutle, A. CEPEDALoCo: An Event-Driven Architecture for Integrating Complex Event Processing and Blockchain through Low-Code. *Internet Things* **2023**, *22*, 100802. [\[CrossRef\]](#)
27. Django. Available online: <https://www.djangoproject.com/> (accessed on 12 July 2025).
28. GeoDjango | Django Documentation. Available online: <https://docs.djangoproject.com/en/5.1/ref/contrib/gis/> (accessed on 12 July 2025).
29. Umer, M.; Mahesh, B.; Hanson, L.; Khabbazi, M.R.; Onori, M. Smart Power Tools: An Industrial Event-Driven Architecture Implementation. *Procedia CIRP* **2018**, *72*, 1357–1361. [\[CrossRef\]](#)
30. Raptis, T.P.; Passarella, A. On Efficiently Partitioning a Topic in Apache Kafka. In Proceedings of the 2022 International Conference on Computer, Information and Telecommunication Systems (CITS), Piraeus, Greece, 13–15 July 2022; IEEE: New York, NY, USA, 2022; pp. 1–8.
31. Apache Kafka. Available online: <https://kafka.apache.org/> (accessed on 22 December 2025).
32. Martín, C.; Langendoerfer, P.; Zarrin, P.S.; Díaz, M.; Rubio, B. Kafka-ML: Connecting the Data Stream with ML/AI Frameworks. *Future Gener. Comput. Syst.* **2022**, *126*, 15–33. [\[CrossRef\]](#)
33. Freeman, A. *Essential TypeScript 4: From Beginner to Pro*; Apress: Berkeley, CA, USA, 2021; ISBN 978-1-4842-7010-3.
34. Beattie-Hood, B. *Modern TypeScript: A Practical Guide to Accelerate Your Development Velocity*; Apress: Berkeley, CA, USA, 2023; ISBN 978-1-4842-9722-3.
35. Goel, A.K.; Gupta, S.; Singh, C.K.; Agrawal, K.K. Web-ChatLine: An Innovative Chatting Platform. *Mater. Proc.* **2022**, *10*, 6.
36. Ferreira, F.; Valente, M.T. Detecting Code Smells in React-Based Web Apps. *Inf. Softw. Technol.* **2023**, *155*, 107111. [\[CrossRef\]](#)
37. Bhatia, T.S.; Singh, H.; Litoria, P.K.; Pateriya, B. Web GIS Development Using Open Source Leaflet and Geoserver Toolkit. *Int. J. Comput. Sci. Technol.* **2018**, *9*, 29–33.
38. Horbiński, T.; Lorek, D. The Use of Leaflet and GeoJSON Files for Creating the Interactive Web Map of the Preindustrial State of the Natural Environment. *J. Spat. Sci.* **2022**, *67*, 61–77. [\[CrossRef\]](#)
39. Buzzo, D. Signs of Surveillance. In *Technology, Design and the Arts—Opportunities and Challenges*; Earnshaw, R., Liggett, S., Excell, P., Thalmann, D., Eds.; Springer Series on Cultural Computing; Springer International Publishing: Cham, Switzerland, 2020; pp. 159–181. ISBN 978-3-030-42096-3.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.