No-Code GIS for Big Spatial Data: A Platform for Interactive Logistics Visualization

Sefa Sarı[1], Ali Osman Kabil[1], Mustafa E. Kamaşak[2]

[1]Ar-Ge Merkezi, Yurtiçi Kargo Servisi A.Ş., İstanbul, Türkiye – {sefa.sari, ali.kabil}@yurticikargo.com
[2]Bilgisayar Mühendisliği Bölümü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye – kamasak@itu.edu.tr

Keywords: No-code GIS, Spatial Decision Support, Geovisualization, Cloud-based Architecture, Location Intelligence, Logistics Optimization

Abstract

This study introduces a scalable, cloud-native, no-code Geographic Information System (GIS) platform tailored for large-scale spatial decision-making in logistics operations. The platform aims to enable non-technical users such as field operators and branch managers—to design, deploy, and visualize spatial queries and dashboards without requiring software development skills. Its architecture integrates a multi-layered microservices framework with a PostGIS backend and supports dynamic visualization of millions of spatial points in real time. Advanced optimization modules, such as clustering (k-means, DBSCAN) and location-routing algorithms (p-median, set covering), are encapsulated as configurable plug-ins within the platform. Designed and tested in a national courier network, the system addresses major technical challenges such as query performance on fragmented big data, visual responsiveness under heavy load, and inter-database integration via engines like Trino. Experimental evaluations demonstrate sub-second response times for complex spatial aggregations over datasets exceeding 1.5 million records. The platform fosters improved operational visibility, reduced IT dependency, and faster decision loops in dynamic geospatial environments. The architecture and findings of this work present a practical contribution to enterprise-grade GIS systems and open new research directions in user-centric geospatial analytics.

1. Introduction

In the logistics and transportation industry, spatial reasoning plays a central role in planning, routing, and operational efficiency. With thousands of shipments moving daily across complex urban and regional networks, decisions related to branch placement, delivery zoning, and route optimization are inherently geographic. Consequently, the ability to model, analyze, and visualize spatial dynamics in real time is a critical capability for maintaining competitiveness and service quality in the sector (Li et al., 2021, Yang et al., 2020). Despite this centrality, spatial data remains underutilized in many logistics organizations due to fragmented tooling, limited access to technical GIS expertise, and high costs associated with commercial platforms. Traditional Geographic Information Systems (GIS) often require domain-specific programming, complex data integration workflows, and dedicated IT support to translate raw geospatial data into actionable insights (Cheng et al., 2023). These constraints pose a significant bottleneck for non-technical users, such as field operators or regional managers, who lack the tools or technical background to interact directly with geographic data. In recent years, the rapid growth of spatial data and its critical role in decision-making processes have created an increased demand for Geographic Information Systems (GIS) that are more accessible, scalable, and responsive to domain-specific needs. This is particularly true in logistics and transportation, where real-time spatial analysis and operational agility can significantly influence competitiveness (Malczewski, 2006). Traditional GIS platforms, while powerful, often require specialized technical expertise, extended development cycles, and substantial infrastructure (Wang et al., 2021). These constraints limit their usability by non-developer stakeholders, such as operations managers or field coordinators. Moreover, as spatial datasets grow in size and complexity—with millions of delivery points, customer locations, and road segments—existing tools struggle with rendering performance and analytical scalability. Many logistics firms resort to using multiple disconnected systems: one for mapping, another for analysis, and a third for business reporting. This creates a fragmented ecosystem that introduces latency, inconsistency, and high maintenance overhead (Sun et al., 2022). Another notable challenge is the lack of intuitive mechanisms to visualize spatio-temporal patterns such as delivery heatmaps, missed delivery clustering, or zone-based performance metrics, which often require extensive data wrangling or scripting (Zhang et al., 2022). To address these issues, the recent emergence of low-code GIS platforms offers a promising paradigm shift. By abstracting complex backend processes and enabling SQL-based scripting for spatial logic, low-code solutions empower both technical and non-technical users to co-develop dynamic dashboards and maps tailored to logistics workflows (Wang et al., 2022). These platforms often combine cloud-native scalability with customizable visualization engines, providing a unified interface to connect data from heterogeneous sources—such as PostgreSQL/PostGIS, Oracle, and MongoDB—and render millions of geographic records in real time.

Technological innovations such as cloud-native architectures, microservice based modular designs, and WebGL-powered frontends have further accelerated the development of lightweight, scalable GIS platforms (Yang et al., 2020). These architectures enable distributed processing of large-scale spatial data while maintaining high responsiveness and fault tolerance (Li et al., 2020). Additionally, the integration of real-time data sources—such as vehicle telemetry, shipment updates, and customer interactions into spatial dashboards enhances situational awareness and shortens decision cycles (Chen et al., 2019). In this context, PostgreSQL/PostGIS databases, together with visualization engines like Mapbox and Deck.gl, provide the necessary foundation for interactive, browser-based GIS solutions (Gonzalez et al., 2019). Such technologies not only enable rich cartographic rendering but also support analytical operations such as spatial joins, clustering, and proximity analysis at scale. Furthermore, modular system design facilitates the implementation of various geospatial analysis models, including clustering (e.g., k-means, DBSCAN), location optimization (e.g., p-median, set covering), and route planning (e.g., vehicle routing problem, travelling salesman problem) (Ashtari et al., 2020). These models can be embedded as configurable add-ons, making the system adaptable to use cases across sectors beyond logistics, such as urban planning or public service optimization (Balram and Dragicevic, 2003).Despite these advancements, the literature still lacks implementations of general purpose, cloud-based GIS platforms specifically designed for logistics organizations that do not rely on proprietary software or demand heavy IT involvement. This study proposes a novel, no-code GIS platform architecture tailored for operational decision-making within such organizations. The platform, developed in collaboration with an industry-leading logistics company, is evaluated based on scalability, usability, integration capacity, and analytical performance.

In this context, we present a domain-specific, low-code GIS platform tailored to the operational requirements of a national courier company. Developed within the IT and RD departments of Yurtiçi Kargo, the system provides a unified environment for spatial querying, visualization, clustering, and decision support. It bridges the gap between IT-intensive GIS development and operational agility, allowing logistics stakeholders to create their own geospatial panels using SQL-based definitions and reusable configuration objects. The remainder of this paper is structured as follows. Section 2 presents the system architecture, including design principles, core components, and implementation challenges. Section 3 provides an evaluation of system performance and usability based on real-world deployment scenarios. Finally, Section 4 concludes the paper with a summary of findings and future directions.

2. System Architecture and Design

2.1 System Overview

The developed system is a low-code, integrated Geographic Information System (GIS) platform designed to meet the spatial data analysis needs of large-scale logistics operations, specifically tailored for the corporate context of Yurtiçi Kargo. Unlike conventional GIS tools that rely on disparate third-party software and require extensive development effort, this platform consolidates visualization, analysis, and decision support functions into a unified architecture. It employs a modular microservices based design following a client-server model, enabling multiple dynamic panels to interact with a centralized computation engine through SQL queries and API calls. Core technologies include C#, .NET Core MVC for backend logic, a hybrid database layer consisting of MongoDB, Oracle, and PostgreSQL for handling high-volume spatial data, and JavaScript powered front-end interfaces enhanced with mapping engines like Google Maps, OSM, and Here. The system enables both technical personnel to define advanced spatial logic through SQL based scripting and non-technical users to interact with data through user-friendly visualization tools. Key features such as address parsing, spatial clustering (k-means, p-median), heatmap generation, and dynamic zoning are delivered through containerized add-on modules, allowing horizontal scalability and flexible integration. By providing a centralized, customizable environment for spatial intelligence, the platform eliminates the need for fragmented software ecosystems, reduces IT dependency, and enhances operational agility across organizational units.

2.1.1 Modules and Functional Components

The platform is composed of several modular components, each responsible for a specific function within the spatial analysis and decision support workflow. These modules are loosely coupled and communicate through RESTful APIs and internal service buses, ensuring scalability, reusability, and maintainability across deployments. The main modules are outlined below:

- **Data Ingestion and Harmonization Module:** This component is responsible for importing, cleaning, and standardizing data from various internal sources, including MongoDB (historical parcel tracking), Oracle (real-time operational data), and PostgreSQL/PostGIS (spatial datasets). The system leverages the Trino SQL query engine to allow cross-database querying in ANSI-compliant syntax.
- **SQL Scripting Engine:** Designed as a low-code interface, this engine allows technical users to define analytical panels using SQL-based instructions. The scripting layer supports modular queries, variable definitions, and parameterized execution, enabling dynamic dashboards that respond to user input or data triggers.
- **Visualization Engine:** This module handles rendering of spatial outputs such as point clusters, choropleth maps, heatmaps, travel-time-based buffers, and temporal-spatial animations. Mapping is powered by external APIs (Google Maps, OSM, Here) and rendered using client-side libraries in the frontend layer.
- **Clustering and Optimization Add-ons:** This set of containerized add-ons provides access to complex algorithms such as k-means clustering, p-median and p-center location models, DBSCAN, and set covering formulations. These models are used for strategic tasks such as branch placement, region assignment, and route planning.
- **Access Control and Security Layer:** To ensure data privacy across departments, the system includes a role-based access control (RBAC) model. Each user or user group is assigned specific panel visibility and query execution privileges, enforced at both backend and frontend layers.
- **Dashboard Orchestration Layer:** Dashboards are not hard-coded but generated dynamically based on metadata stored in a configuration layer. Each dashboard consists of multiple panels, each tied to a SQL command and a rendering directive (e.g., map, table, chart), which are interpreted and displayed at runtime.

The functional structure enables collaborative yet secure use of the system across technical and non-technical users. Analysts can focus on defining spatial logic and data relationships using SQL, while operational users can interpret results through rich visualizations. This dual-layer design ensures that domain knowledge and data engineering efforts are decoupled, allowing faster iteration and deployment of geospatial decision support tools.

2.1.2 Implementation and Technical Challenges

The implementation of the platform encountered several technical and architectural challenges due to the hybrid requirements of flexibility, performance, scalability, and user accessibility in a single integrated environment. The most critical issues and corresponding mitigation strategies are summarized below:

- **Cross-Database Querying:** One of the fundamental challenges was the need to integrate data from three distinct sources (MongoDB, Oracle and PostgreSQL) without duplicating or migrating data. This was addressed by deploying a *Trino SQL Query Engine*, which provides ANSI SQL-based federated querying capabilities. Through custom connectors and metadata mapping, users can write unified SQL scripts that fetch, join, and filter spatial and non-spatial data in a coherent and performant manner.
- **Low-Code Script Interpreter:** To allow analysts to define interactive panels using SQL without traditional coding, a custom-built parser engine was developed. This engine performs grammar checks (syntax validation), semantic interpretation and runtime optimization of SQL queries. The design supports user defined parameters and conditional logic, mimicking procedural behavior while retaining declarative SQL structure.
- **Performance Bottlenecks in Large Datasets:** The system is designed to handle millions of records across spatial and tabular formats. Index optimization, caching mechanisms, query result reuse, and asynchronous execution were employed to minimize query latency. Additionally, spatial queries were optimized using spatial indexing (R-trees, quadtrees), simplification techniques (geometry generalization), and spatial joins handled server-side to reduce frontend load.
- **Dynamic Rendering of Visual Components:** As the dashboard content is not hard-coded but generated dynamically from user scripts, real-time rendering of charts, maps, and grids required an abstract representation of each visual component. A rendering engine was built to convert script output into interactive components using JavaScript libraries Leaflet, Highcharts, Vega-lite), ensuring consistency and responsiveness across devices.
- **Concurrency and Multi-Tenancy:** Supporting simultaneous access by different business units necessitated a robust multi-user architecture. The system supports multi-tenancy through role-based authentication and per-user configuration caching. Each user accesses only the panels and datasets assigned to their profile, and conflicting edits or execution tasks are sandboxed in isolated containers.
- **Fault Tolerance and System Recovery:** Given the platform's integration into operational processes, downtime had to be minimized. Microservice containerization via Docker and Kubernetes allowed isolated deployments, restarts and health monitoring of components. Logging and alerting mechanisms are tied to a centralized ELK stack (Elasticsearch–Logstash–Kibana), ensuring early detection of anomalies.
- **Scalability of Add-ons and Analytical Extensions:** Optimization models and clustering tools packaged as add-ons require computational resources that differ from standard SQL query execution. These services are deployed as independent containers and triggered via task queues, with results written back to the main database for visualization. This design enables elastic scaling of analytical modules without impacting core performance.

These challenges reflect the inherent complexity of building a domain-agnostic, low-code spatial analytics system in a logistics context. However, by combining scalable cloud-native components with spatial intelligence and interactive visual design, the platform successfully bridges the gap between technical and operational stakeholders.

3. Evaluation and Results

To validate the functionality, performance, and usability of the platform, a series of controlled experiments and field tests were conducted using real-world logistics data from Yurtiçi Kargo's operational environment. The evaluation focused on both technical performance metrics and user-level effectiveness.

3.1 Performance Testing

The system was deployed in a staging environment with the following specifications: Windows 10, Intel Core i7-7700HQ CPU @ 2.80GHz, 16GB RAM, and SSD storage. Datasets included delivery records, shipment logs, address databases, and geospatial boundary files from major Turkish provinces. Key findings are summarized as follows:

- **Point Visualization:** Rendering and interaction with spatial datasets containing up to **2 million points** was successfully achieved with smooth panning and zooming, aided by spatial clustering simplification techniques.
- **Query Execution Speed:**
  - Standard SQL-based thematic queries on 500K+ records executed within **3–6 seconds**.
  - Heatmap and density-based thematic visualizations over 1 million records rendered within **5–8 seconds**, depending on geographic spread and zoom level.
- **Cross-Database Joins:** Federated queries using Trino over MongoDB, PostgreSQL, and Oracle completed with no critical performance bottlenecks; indexed fields ensured acceptable latency.
- **Concurrent Access:** The system sustained **20+ simultaneous user sessions** executing independent panels and map-based queries with no measurable degradation in performance.

3.2 Usability Feedback and Functional Validation

A test group comprising logistics planners, regional managers, and business analysts were onboarded with no prior coding experience. A brief training session (under 2 hours) was sufficient to enable users to:

- Create and execute custom visualizations using SQL snippets.
- Build conditional filters, apply joins, and define dynamic UI elements like dropdowns and sliders.
- Export visualizations, perform proximity and grid-based analyses, and share dashboards with team members.

Qualitative feedback indicated high satisfaction with:

- The freedom to operate without IT mediation,
- The visibility over spatial business data,
- The reusability of created panels across departments.

To demonstrate the user interface and visualization capabilities of the developed platform, a sample heat map generated from spatial data analysis is presented in Figure 1. The heat map illustrates the spatial distribution of the selected parameter values across the study area, where the color scale represents varying intensity levels. This visualization not only provides an intuitive understanding of spatial patterns but also enables users to identify clusters, anomalies, and high-impact zones in real time. Such visual analytics enhance decision-making by offering immediate insights into the underlying geospatial phenomena.

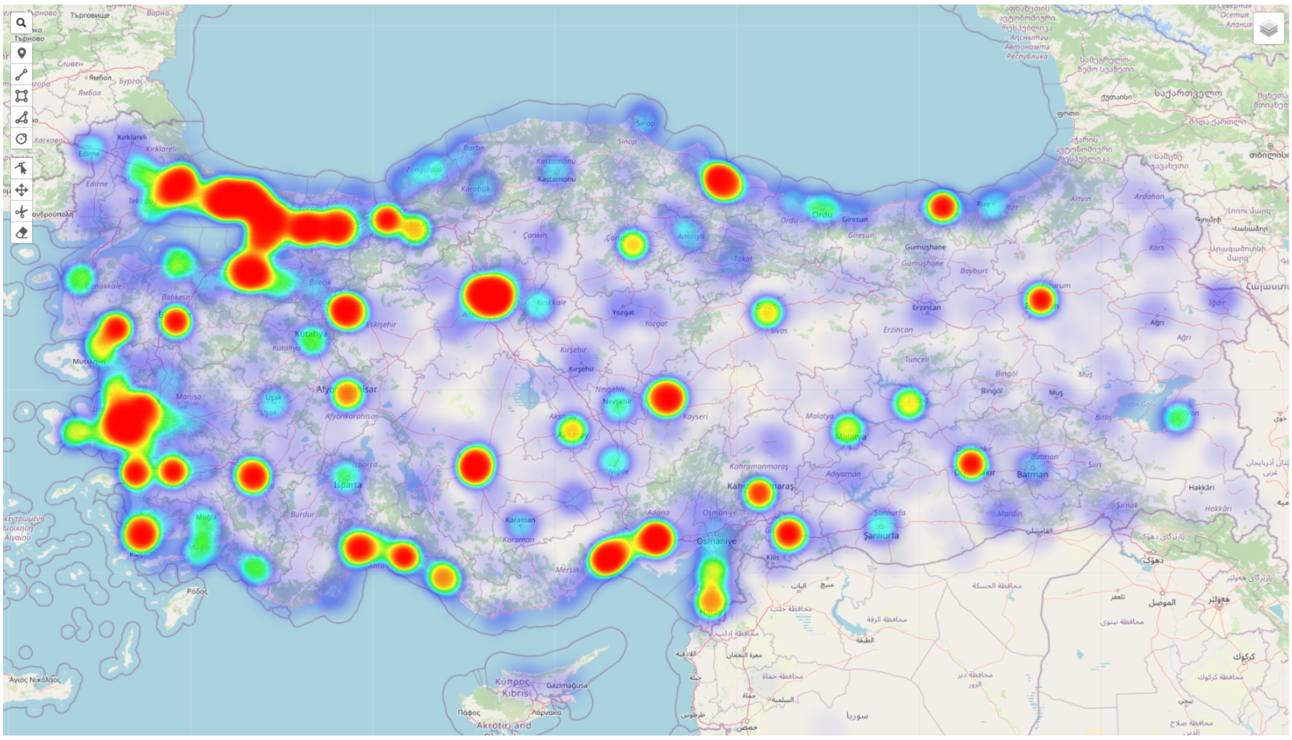

Figure 1. Sample heat map displayed in the platform's user interface. The color gradient represents the intensity of the analyzed parameter values across the study area.

3.3 Validation of Analytical Add-ons

The clustering and optimization extensions were tested with real delivery datasets. For example:

- **K-Means Clustering Add-on:** Used to identify candidate regions for new branches; the visual output confirmed spatial grouping accuracy.
- **Set Covering Model:** Applied to minimize total delivery distance while ensuring 100
- **P-Median Facility Location:** Simulated the reallocation of depot locations across İstanbul and İzmir; results validated by comparing average customer-to-facility distances before and after optimization.

These test cases demonstrate the platform's capacity to bridge the gap between large-scale geospatial datasets and actionable business insights.

4. Conclusion

This study presents a domain-specific low-code geospatial platform tailored to the operational needs of logistics enterprises, particularly Yurtiçi Kargo. Unlike traditional GIS tools that rely heavily on developer expertise or require integration of multiple commercial products, the platform provides an integrated, SQL-driven, modular environment for spatial analysis, visualization, and decision support.

Through a layered system architecture featuring microservices, a PostgreSQL/PostGIS backend, Trino for federated queries, and a rule-based panel engine, the platform supports efficient interaction with millions of geospatial records while maintaining responsiveness and security. Analytical add-ons such as clustering, facility location models, and set covering algorithms are embedded directly into the system, allowing business users to run complex spatial analyses without programming knowledge. Field tests conducted using real operational datasets confirmed the platform's stability, performance, and usability. The ability to rapidly prototype and deploy custom geospatial dashboards improved spatial visibility across departments, enabling faster and more informed decision-making.

In addition to its immediate impact on logistics planning and performance monitoring, the platform establishes a reusable infrastructure that can be extended to domains such as retail expansion, public services and urban logistics. The solution also mitigates the dependency on foreign technologies and ensures data privacy with the client's infrastructure.

Future work will focus on expanding the add-on ecosystem, integrating AI-based predictive analytics, and evaluating cross-domain applications in smart city planning and e-government services.

References

Ashtari, P., Gholamrezaee, S., Alesheikh, A., 2020. A modular GIS framework for spatial decision support in urban logistics. *Computers, Environment and Urban Systems*, 82, 101478.

Balram, S., Dragicevic, S., 2003. Towards participatory GIS: The role of map-based tools in spatial decision making. *Environment and Planning B: Planning and Design*, 30(3), 349–362.

Chen, Y., Zhang, C., Wang, F., 2019. Real-time GIS in logistics: An integrated solution with sensor networks and cloud processing. *Sensors*, 19(18), 3934.

Cheng, Y., Zhao, J., Liu, C., 2023. Enterprise-scale geospatial systems: Architecture and deployment experience. *ISPRS International Journal of Geo-Information*, 12(1), 27.

Gonzalez, C., Boulton, A., Brewer, C., 2019. Visualizing big geospatial data with Deck.gl: An evaluation. *Cartographic Journal*, 56(2), 93–104.

Li, H., Wang, X., Zhang, J., 2021. Spatial intelligence in logistics: Integrating GIS with data-driven decision making. *IEEE Transactions on Intelligent Transportation Systems*, 22(7), 4291–4303.

Li, S., Wu, Y., Liu, Y., 2020. Designing scalable spatial analytics systems using cloud microservices. *Future Generation Computer Systems*, 112, 132–144.

Malczewski, J., 2006. GIS-based multicriteria decision analysis: A survey of the literature. *International Journal of Geographical Information Science*, 20(7), 703–726.

Sun, Y., Wang, S., Huang, Y., 2022. Cloud-native GIS for real-time logistics: Design and performance evaluation. *ISPRS International Journal of Geo-Information*, 11(8), 414.

Wang, C., Li, Z., Tang, X., 2022. Towards low-code GIS platforms: An enabling framework for spatial analytics. *IEEE Access*, 10, 73119–73133.

Wang, J., Li, D., Gao, S., 2021. Review on cloud GIS platforms: Concepts, architecture, and trends. *Journal of Geographic Information System*, 13(3), 345–362.

Yang, C., Wu, H., Huang, Q., Li, Z., 2020. Big geospatial data processing: Challenges and opportunities. *Computers, Environment and Urban Systems*, 81, 101418.

Zhang, L., Yu, M., Song, Y., 2022. Interactive geovisual analytics for spatio-temporal logistics data. *International Journal of Digital Earth*, 15(6), 1291–1310.