Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084386

ARTICLE

IoT-Enabled Middleware for Smart City Environmental Monitoring with Integrated Analytics and Visualization

Zulfiqar Ali, Azhar Mahmood*, Shaheen Khatoon and Seyed Ali Ghorashi

School of Architecture, Computing and Engineering, Department of Computer Science and Digital Technologies, University of East London, University Way, London, UK

*Corresponding Author: Azhar Mahmood. Email: a.mahmood3@uel.ac.uk

Received: 21 April 2026; Accepted: 26 June 2026; Published: 13 August 2026

ABSTRACT: Smart city systems increasingly depend on data analytics and visualization to support informed and timely decision making in complex urban environments. However, existing middleware solutions predominantly focus on data acquisition and communication, while analytical processing and visualization are typically delegated to external applications, resulting in increased development complexity, reduced reusability, and fragmented system architectures. This study presents analytics and visualization-centric middleware named "Service-Oriented Middleware for Smart City Applications" (SOMSCA), in which these capabilities are embedded directly within the middleware layer. SOMSCA adopts a service-oriented approach, exposing analytics and visualization functionalities as reusable platform services, and incorporates a data-type-oriented visualization strategy along with a template-assisted dashboarding mechanism to enable dynamic and flexible application development. To validate the proposed approach, a prototype implementation is developed using React, FastAPI, MySQL, and TimescaleDB and evaluated using air-quality data collected from the London Air Quality Network, comprising more than one million observations across multiple pollutants and monitoring locations. The implementation supports real-time, historical, and aggregated analytical services together with dynamic dashboard generation. The results demonstrate the practical feasibility of middleware-integrated analytics and visualization through reusable service creation, flexible dashboard configuration, and interactive environmental monitoring capabilities. These findings highlight the potential of middleware-level intelligence to simplify application development and support data-driven decision making in smart city environment.

KEYWORDS: Middleware; smart city; internet of things (IoT); data analytics; data visualization; service-oriented architecture; dashboarding; environmental monitoring; air quality; decision support

1 Introduction

The rapid pace of urbanization continues to impose significant demands on modern cities, necessitating intelligent systems capable of efficiently managing large volumes of heterogeneous data generated from diverse urban infrastructures. Smart city environments increasingly rely on Internet of Things (IoT) technologies to enable continuous data collection across domains such as environmental monitoring, transportation, energy, and public services. However, the true value of such data lies not merely in its acquisition, but in its transformation into actionable insights that support timely and informed decision-making.

Several frameworks and systems have been proposed in the literature that focus on data analytics and visualization across various domains with some specifically targeting environmental monitoring [1,2].

Although these are reasonable solutions for serving a specific domain, and typically developed as domain-specific applications rather than as middleware, limiting their generalizability and reuse. Consequently, frameworks designed for environmental monitoring cannot be readily adapted or extended to other domains, such as transportation or energy management.

This lack of extensibility represents a significant limitation, as it leads to fragmented solutions and duplicated development efforts across domains. The integration of analytics and visualization capabilities within the middleware enables their reuse across diverse applications and helps ensure flexibility, extensibility, and scalability.

Middleware has been widely adopted as a fundamental architectural layer in smart city systems, primarily to address challenges related to heterogeneity, interoperability, and scalability. Existing middleware solutions effectively facilitate communication between distributed IoT devices and application services, enabling seamless data exchange and system integration. Nevertheless, these platforms largely remain focused on data acquisition, message brokering, and system-level coordination [3–7], while analytical processing and visualization capabilities are typically delegated to the application layer.

This architectural separation introduces several limitations. First, it leads to duplication of effort, as similar analytical logic and visualization components must be re-implemented across multiple applications. Second, it increases system complexity and development overhead, reducing reusability and slowing down application deployment. More critically, this approach restricts the ability to perform real-time analytics and monitoring at the middleware level, thereby limiting visibility into system behavior and delaying actionable insights.

Despite the growing importance of data-driven urban intelligence, analytics and visualization have often been overlooked within middleware design. This is primarily due to the traditional perception of middleware as a passive integration layer, where the emphasis is placed on data transmission and protocol abstraction rather than on intelligence and insight generation. Furthermore, concerns related to performance overhead, scalability, and architectural complexity have discouraged the incorporation of computationally intensive analytics and visualization functionalities within middleware environments.

However, the integration of analytics and visualization directly into middleware presents a significant opportunity. It enables real-time data interpretation, improves system observability, supports early detection of anomalies, and enhances decision-making capabilities across smart city applications. By embedding these capabilities within the middleware layer, it becomes possible to provide reusable, standardized services for analytics and visualization, thereby reducing redundancy and improving overall system efficiency.

This raises a critical research question in middleware design: how can data analytics and visualization be effectively integrated into middleware architectures without compromising performance, scalability, or system simplicity? Addressing this challenge is essential to fully leverage the potential of urban data and to advance the development of intelligent, responsive smart city systems.

In response, this study builds upon the Service-Oriented Middleware for Smart City Applications (SOMSCA) and focuses specifically on its data analytics and visualization components. While an overview of the middleware architecture is provided, the primary emphasis is on the design, implementation, and evaluation of integrated analytics and visualization services within the middleware. Through an environmental monitoring use case, the study demonstrates how such integration can enable efficient data processing, reusable service design, and interactive visualization, ultimately supporting more effective and scalable smart city applications.

The main contributions of this study are as follows:

1. Development of a middleware-integrated analytics and visualization module along template-assisted dashboarding for smart city applications.
2. Conceptualization and validation of data-type-oriented strategy for a large-scale environmental monitoring use case of a smart city.

The article is structured as follows. Section 2 presents the literature review. Section 3 introduces the proposed analytics- and visualization-centric middleware. Section 4 details the implementation of the data analytics and visualization components, including their architecture, service layers, and the end-to-end workflow, as well as their internal integration. Section 5 presents the environmental monitoring use case and corresponding results. Section 6 provides a discussion of the findings. Finally, Section 7 summarizes the conclusions and outlines directions for future work.

2 Literature Review

Internet of Things technologies have become a fundamental enabler of smart city initiatives by facilitating continuous monitoring of urban environments through interconnected sensors, communication networks, and intelligent services. Environmental monitoring represents one of the most widely adopted smart city applications, supporting measurement of air quality, temperature, humidity, noise levels, and other environmental indicators. The large volume of data generated by these systems requires efficient mechanisms for collection, storage, processing, and interpretation [7,8].

Middleware technologies have therefore emerged as a critical architectural component in IoT-based smart city systems. They provide abstraction between heterogeneous devices and applications while addressing challenges related to interoperability, scalability, and resource integration. Although considerable research has focused on communication and integration capabilities, support for integrated analytics and visualization remains comparatively limited within existing middleware solutions [6,9,10].

The role of middleware in smart city applications has been extensively investigated, particularly in addressing challenges such as heterogeneity, interoperability, and scalability. However, despite significant progress in these areas, most existing middleware frameworks provide limited support for integrated data analytics and visualization, which are essential for enabling intelligent and data-driven urban decision-making.

A number of middleware solutions incorporate context-awareness and semantic reasoning as core capabilities. For example, in [10], ontology-based reasoning to support context-aware IoT applications is employed, whilst in [11], event-driven semantic processing for adaptive systems is introduced. Although these frameworks enable higher-level abstraction of sensor data, they do not provide embedded analytical pipelines or visualization mechanisms, thereby requiring external systems for insight generation. Similarly, SMArc focuses on semantic reasoning within specific domains, yet lacks support for cross-domain analytics and visualization, limiting its applicability in complex smart city environments [12].

Several frameworks emphasize modularity and scalability through distributed and service-oriented architectures. InterSCity adopts a microservices-based approach that facilitates scalable development and deployment of urban applications [6]. SmartCityWare and MEx Middleware extend this paradigm by supporting integration across cloud and fog layers, enabling flexible resource utilization [5,13]. Despite these architectural advantages, such frameworks typically externalize analytics and visualization functionalities, resulting in fragmented system designs and increased development overhead.

In parallel, adaptive and edge-oriented middleware solutions have been proposed to enhance responsiveness and localized decision-making. AUSOM utilizes a MAPE-K-based adaptive loop for runtime system

management, while GAMBAS and MinT enable context-aware processing and cooperative data sharing at the edge [14–16]. While these approaches improve performance and reduce latency, their focus remains on system-level optimization rather than comprehensive analytical processing or visualization support.

Beyond middleware research, recent studies have explored advanced analytics in smart city contexts. A detailed investigation [17] provides anomaly detection techniques across IoT networks and surveillance systems, highlighting the growing importance of intelligent data processing in urban environments. In [3], a hybrid AI-IoT framework integrated with digital twins for predictive infrastructure management is proposed, demonstrating the potential of combining real-time data with predictive analytics. In [18], authors review recent technological advancements in smart city management, emphasizing the increasing role of data-driven and AI-enabled solutions. While these contributions demonstrate significant progress in analytics and intelligent systems, they are predominantly developed as standalone or domain-specific solutions and are not integrated within middleware architectures.

Recent research has explored the integration of federated learning with IoT infrastructures to enable distributed intelligence while preserving privacy and reducing communication overhead [19]. Federated learning-based approaches have demonstrated potential for supporting collaborative analytics across geographically distributed smart city environments. Although such techniques are not incorporated in the current implementation, they represent a promising direction for extending middleware-integrated analytical services in future versions of SOMSCA. Among existing platforms, CityPulse represents a notable effort to incorporate analytics for event detection and decision support in urban systems [7]. However, its capabilities remain domain-specific and do not provide flexible or reusable visualization mechanisms. Similarly, other analytics frameworks operate outside the middleware layer, lacking seamless integration with data acquisition, service orchestration, and visualization. To provide a structured comparison, Table 1 summarizes existing middleware frameworks with respect to their support for analytics and visualization.

Table 1: Middleware comparison based on analytics and visualization integration.

Middleware	Embedded Analytics	Embedded Visualization	Reusable Services	Dashboard Support	Integration Level	Limitation
CA4IOT [10]	Partial	No	No	No	External	No visualization
MASSIF [11]	Partial	No	No	No	External	Limited analytics
InterSCity [6]	No	No	No	No	External	No intelligence
CityPulse [7]	Partial	Limited	Limited	Limited	External	Domain-specific
SmartCityWare [13]	No	No	No	No	External	No analytics
MEx Middleware [5]	Partial	No	No	No	External	Limited analytics
SMArc [12]	No	No	No	No	External	Limited analytics
GAMBAS [14]	No	No	No	No	External	Limited analytics
AUSOM [15]	No	No	No	No	External	Limited analytics
MinT [16]	No	No	No	No	External	Limited analytics
SOMSCA	**Yes**	**Yes**	**Yes**	**Yes**	**Internal**	**None**

It is important to distinguish SOMSCA from data analytics and visualization platforms such as ELK Stack, Grafana-based dashboards, and cloud-native monitoring environments. While these platforms provide powerful monitoring, storage, and visualization capabilities, they generally operate as supporting infrastructure or application-level tools. In contrast, SOMSCA integrates analytics and visualization directly within the middleware layer and exposes these capabilities as reusable services that can be consumed by multiple applications. This approach reduces duplication of analytical logic, promotes reuse, and simplifies the development of intelligent smart city applications.

The analysis reveals that, although certain frameworks provide partial support for analytics, visualization capabilities are largely absent, and integration is typically achieved through external systems. This results in limited reusability, increased development complexity, and a lack of unified intelligence within middleware platforms. Unlike existing systems, SOMSCA uniquely integrates both analytics and visualization as reusable middleware services.

The literature review highlights several limitations in existing approaches. Most middleware platforms focus primarily on interoperability, communication management, and data integration while providing limited support for embedded analytical processing. Visualization functionality is typically implemented within external applications rather than within middleware architectures, resulting in duplicated development effort and reduced reusability. Furthermore, existing analytical solutions are frequently domain-specific and difficult to reuse across multiple smart city applications. These limitations motivate the development of SOMSCA, which integrates analytics and visualization directly into the middleware layer through reusable service-oriented components.

In the following sections, an overview of the proposed SOMSCA middleware is presented, along with in-depth details of the data analytics and visualization capabilities. Section 3 describes the SOMSCA middleware and its capabilities along with the strategy to address the challenge of flexibility and extensibility by using the data-type-oriented strategy for visualizations, while Section 4 presents the implementation details, including the end-to-end process of creating services and visualizations using the SOMSCA middleware.

3 Analytics and Visualization-Centric Service-Oriented SOMSCA

Smart city systems typically follow a layered IoT architecture consisting of sensing, communication, middleware, and application layers. The sensing layer comprises environmental sensors and IoT devices responsible for generating observations. The communication layer provides connectivity and data transmission, while the middleware layer performs integration, processing, storage, and service management. The application layer delivers end-user services and decision-support functionality. SOMSCA operates primarily within the middleware layer and extends traditional middleware capabilities through integrated analytics and visualization services.

The SOMSCA middleware facilitates the development of smart cities applications with significantly reduced effort while ensuring cost-effective implementation. The architecture of SOMSCA is structured into three logically distinct modules. Each module encapsulates a coherent set of responsibilities that together enable the middleware to support real-time data acquisition, secure and scalable storage, and intelligent decision-making using data analytics and visualizations. These modules are: *The Core Services Module*, the *Data Persistence Module*, and the *Data Analytics and Visualization Module*. The design of each module has been guided by principles of modularity, separation of concerns, and extensibility, all of which are considered essential for supporting a wide range of smart city use cases. In addition to these, an Application Programming Interface (*API) Listener* and *Request Dispatcher* component serves as the entry point for both external and internal service calls. This component is responsible for validating request structure and authentication, dispatching requests to appropriate modules, performing access control, scheduling tasks, and monitoring usage patterns. By centralizing request handling, it ensures consistent communication between components and simplifies integration with external applications as shown in Fig. 1.

The *Core Services Module* is responsible for handling external interactions, coordinating internal requests, enforcing security protocols, and performing initial data preprocessing. It includes components that manage user authentication, request dispatching, sensor data acquisition, and rule-based filtering or transformation of incoming data streams.

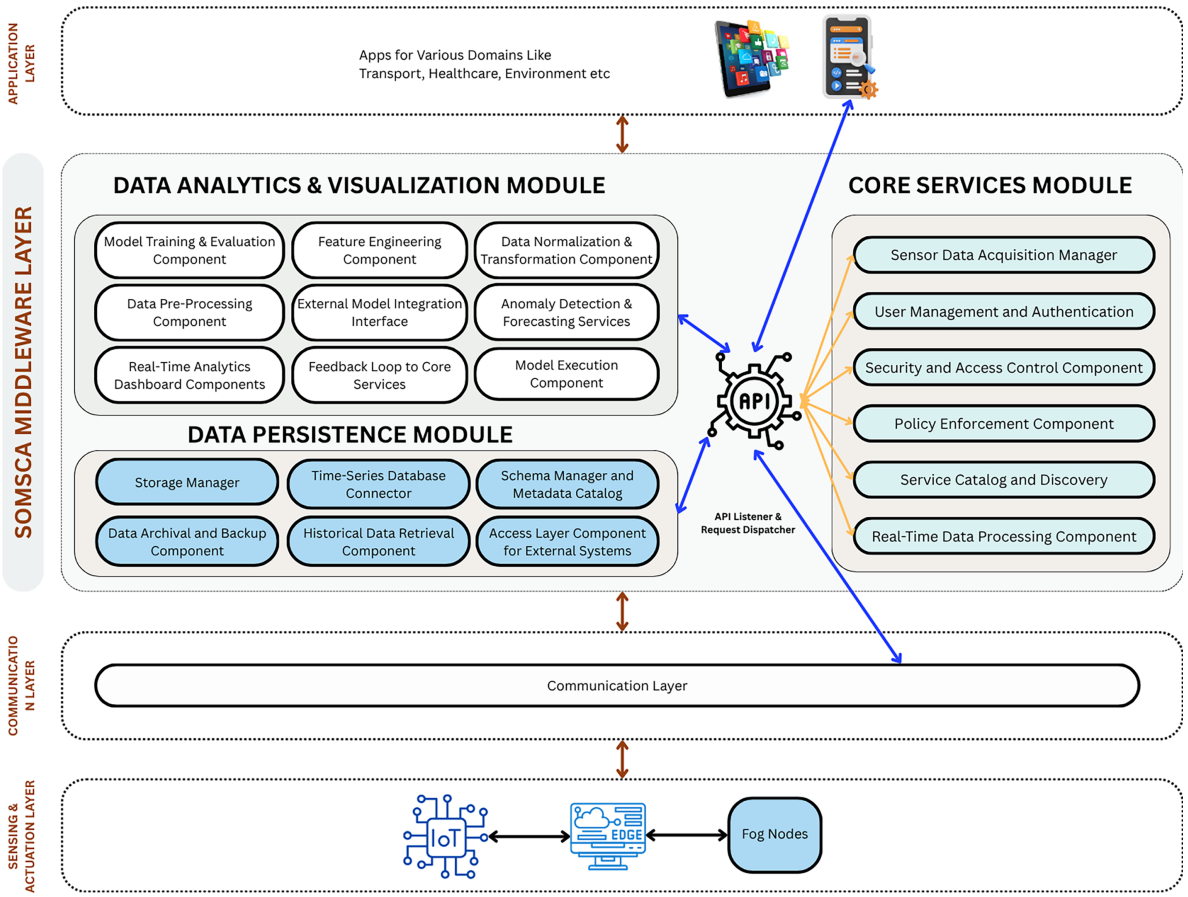

Figure 1: Architecture of SOMSCA middleware.

The *Data Persistence Module* provides the functionality required for long-term storage and structured access to data. It ensures that data ingested from the environment, once processed, is stored in a consistent and query able format. This module includes components for managing database interfaces, schema and metadata registries, archival routines, and historical query services.

The *Data Analytics and Visualization Module* enable the middleware to perform higher-level reasoning through the application of machine learning models and statistical analysis. It supports model training and inference workflows, data transformation routines, anomaly detection services, and the generation of stakeholder-facing visual dashboards. It also includes mechanisms to feed insights back to the *Core Services Module* for triggering automated system actions or alerts. The current study focuses on this module and details how SOMSCA enables the statistical analysis and visual dashboards. Development of predictive models using this module is not in the scope of this study, however, their conceptual application on the visual dashboards is discussed in the following sections. The remainder of this section provides a detailed description of data analytics and visualization module and how it enables the creation of visual dashboards to enable decision making.

3.1 Data Analytics and Visualization

The analytics and visualization functionality is implemented as part of the *Data Analytics and Visualization Module* in SOMSCA middleware, which operates alongside core services and data persistence layers. This module is responsible for:

i. Processing real-time and historical data
ii. Performing statistical and predictive analysis
iii. Generating dynamic visualizations
iv. Supporting reusable dashboard components

The architecture follows a service-oriented design, where analytics and visualization are exposed as reusable platform services. It enables the creation of dashboards for deriving insights from heterogeneous data using a data-type-oriented visualization strategy. The module supports both template-based dashboard generation for various use cases, such as environmental monitoring, transport monitoring, and energy consumption monitoring, as well as the creation of custom dashboards by configuring individual visualizations. In the latter approach, users can select platform services and attach relevant resources, such as sensors, to visualize raw data or perform statistical analysis at runtime. The features of this module are discussed in detail in Section 4.

Design Requirements

The design of SOMSCA is guided by both functional and non-functional requirements commonly encountered in smart city environments.

Functional requirements include acquisition and management of heterogeneous sensor data, support for analytical processing and statistical aggregation, dynamic visualization generation, reusable dashboard creation, and service sharing across multiple applications.

Non-functional requirements include scalability to support large data volumes, extensibility for future analytical capabilities, interoperability with heterogeneous devices and services, maintainability through modular architecture, and reusability of analytical services and visualization components.

These requirements guided the development of the proposed service-oriented analytics and visualization framework.

3.2 Data-Type-Oriented Visualization Strategy

A key challenge in smart city systems is handling heterogeneous data types. SOMSCA addresses this by introducing a data-type-oriented visualization strategy, where visualization options are dynamically determined based on the data type. This ensures that users are only presented with relevant visualization options, improving usability and reducing complexity. A few of the use cases are presented as examples in Table 2. This approach enables flexible visualization across domains without requiring predefined chart configurations.

Table 2: Data-type oriented use case examples with recommended visualizations.

Data Type	Use Case Example	Recommended Visualizations
Time Series	Particulate Matter ≤ 2.5 μm (PM2.5), Nitrogen Dioxide (NO_2)	Line Chart, Area Chart, Heat Map
Aggregated	Borough averages	Bar Chart, Key Performance Indicator (KPI)
Distribution	Pollution spread	Histogram
Geospatial	Ward-level Air Quality Index (AQI)	Heatmap

3.3 Template-Assisted Dashboarding

To simplify application development, SOMSCA introduces *template-assisted dashboarding*. Predefined templates provide reusable configurations for common analytical scenarios. These templates serve as blueprints that developers can customize, enabling rapid creation of dashboards without extensive coding. The template system supports:

i. KPI dashboards
ii. Time-series monitoring dashboards
iii. Aggregation-based analysis dashboards
iv. Environmental and other domains like traffic, energy usage monitoring dashboards, etc.

Dashboard design within SOMSCA focuses on usability, flexibility, and decision support. Users can create dashboards either from predefined templates or by manually configuring visual components. The data-type-oriented visualization strategy ensures that only suitable visualization options are presented for a particular data type, reducing configuration complexity and improving usability.

The dashboard interface supports interactive exploration of analytical results through filtering, time-range selection, and dynamic service execution. This enables decision-makers to investigate environmental conditions from multiple perspectives while maintaining consistency across analytical outputs and visual representations.

3.4 Service-Oriented Analytics and Visualization

A key contribution of SOMSCA lies in its adoption of a service-oriented approach to analytics and visualization, where analytical operations are encapsulated as reusable platform services rather than being implemented independently within each application. This design fundamentally shifts the development paradigm from application-specific implementations to a reusable service ecosystem.

In conventional smart city systems, developers are required to manually construct queries, aggregation logic, and visualization pipelines for each application. This not only increases development effort but also leads to inconsistencies and limited reuse. In contrast, SOMSCA abstracts these operations into standardized services that can be invoked across multiple applications, dashboards, and interfaces. These services are defined through reusable templates and exposed via the middleware as API-driven endpoints. As a result, developers can simply configure and consume these services without needing to implement the underlying analytical logic. The types of analytical services supported within this framework are mentioned in Table 3. By encapsulating these operations as services, SOMSCA enables a unified and standardized approach to analytics and visualization.

Table 3: Supported analytical services.

Service Type	Description
Time-Series Services	Retrieve temporal sensor data for trend analysis
Aggregation Services	Compute statistical measures such as average, minimum, and maximum
Grouped Analysis Services	Perform spatial or categorical aggregation (e.g., by borough or ward)
Distribution Services	Generate frequency distributions and histograms

This architectural decision offers several important advantages. This significantly enhances reusability, as the same service can be consumed across multiple applications without modification. It ensures *consistency*, since all applications rely on the same underlying analytical logic. Finally, it also improves

scalability, as services can be independently optimized and extended without impacting application-level implementations.

3.5 Analytical Capabilities

The SOMSCA middleware supports a comprehensive set of analytical capabilities designed to address the diverse requirements of smart city applications. These capabilities are integrated directly within the middleware, enabling both real-time and historical data analysis without reliance on external systems. It also enables continuous monitoring of incoming sensor data, allowing real-time insights into urban conditions such as air quality levels. In addition, it supports historical data analysis, enabling users to identify trends, seasonal variations, and long-term patterns.

Beyond basic monitoring, the system provides advanced analytical operations, including statistical aggregation and comparative analysis across different spatial or categorical dimensions. For instance, pollution levels can be compared across boroughs or wards, enabling administrators to identify high-risk areas and prioritize interventions. Furthermore, SOMSCA supports distribution-based analysis, allowing users to examine the spread and frequency of pollutant values. This is particularly useful for identifying anomalies and understanding the variability of environmental conditions. These analytical capabilities collectively enable comprehensive decision support, allowing city administrators to move beyond reactive responses and adopt proactive strategies for urban management. The next section describes the implementation of data analytics and visualization module, tools used and different layers of the implementation used to validate and evaluate the proposed analytics and visualization capabilities.

4 Implementation of Data Analytics & Visualization Module

To validate the proposed approach analytics and visualization capabilities, a complete prototype of SOMSCA is implemented. The implementation focuses on demonstrating how service-oriented analytics and visualization can be operationalized within a real-world smart city environment. The system was designed to support the full lifecycle of data processing, from ingestion and storage to analysis and visualization. Particular emphasis was placed on the integration between service templates, platform services, and visualization components, forming a cohesive and reusable architecture.

The complete workflow of the system demonstrates how analytics and visualization are seamlessly integrated within the middleware. Initially, service templates are defined to represent reusable analytical logic. These templates are then used to create platform services by associating them with specific data sources and configurations. Once created, these services are linked to dashboard widgets, which define how the results should be visualized. This flow of dashboard creation is shown in Fig. 2. At runtime, when a user interacts with a dashboard, the system executes the corresponding platform services, retrieves the processed data, and renders the visualizations dynamically.

This workflow minimizes manual query development and simplifies the construction of data-driven applications. It enables fully dynamic dashboard generation, where analytics and visualization are driven entirely by configuration rather than custom development. The prototype for data analytics and visualization module follows a multi-layered architecture that separates concerns while maintaining strong integration between components.

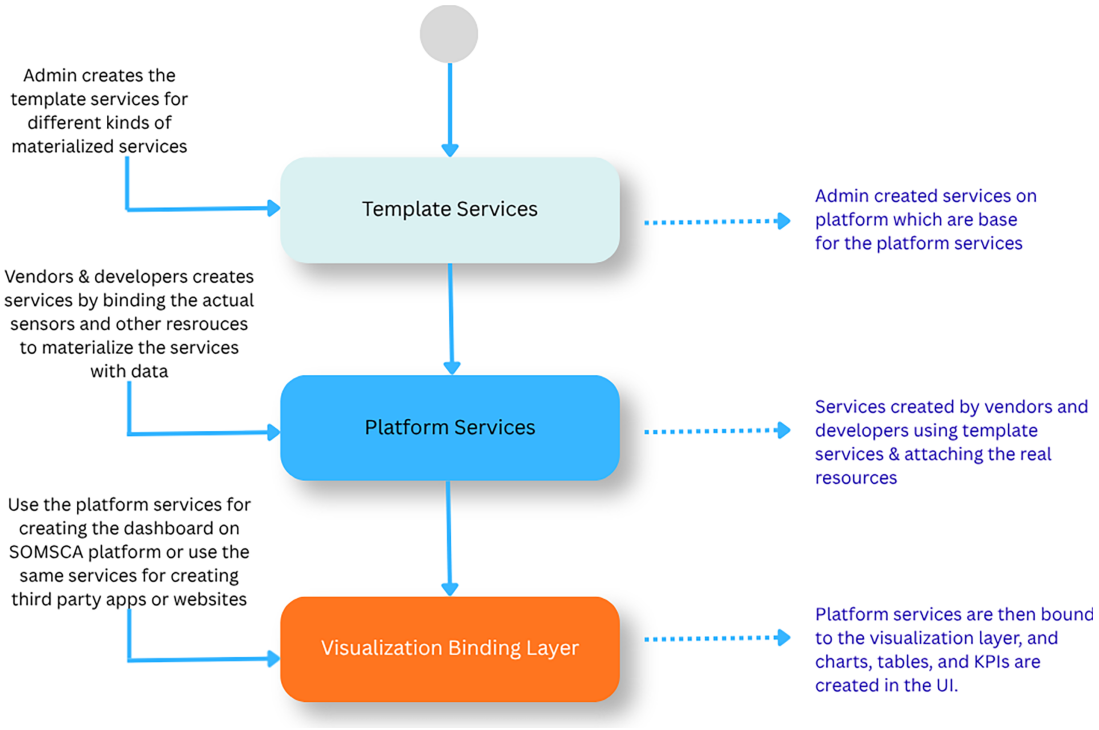

Figure 2: End to end flow of visualization and service integration.

The frontend is implemented using React JS, a modern JavaScript framework, enabling dynamic and interactive visualization [20]. Visualization rendering is performed using ECharts, a charting library that supports a wide range of graphical representations, including time-series charts, bar charts, and heatmaps [21]. The backend is implemented using FastAPI, a lightweight, high-performance API framework, which exposes endpoints for data ingestion, service execution, and dashboard rendering. This layer acts as the orchestration engine, coordinating interactions between data sources, analytical services, and visualization components [22]. A hybrid approach is adopted for data storage where a relational database MySQL is used to manage metadata, including devices, projects, and service configurations, while a time-series database TimescaleDB is employed to efficiently store and query high-frequency sensor data [23,24]. This architecture ensures scalability, modularity, and efficient handling of both structured and temporal data.

4.1 Template Service Layer

The implementation begins with the definition of *service templates*, which form the foundation of the analytical framework. These templates encapsulate reusable analytical logic and define how data should be processed and returned.

Each service template specifies the required input parameters, such as time ranges, pollutant types, and grouping dimensions. It also defines the query logic used to retrieve and process data from the underlying databases, as well as the structure of the output returned to the client.

Different types of templates were developed to support various analytical scenarios. These include templates for retrieving time-series data, computing aggregated statistics, performing grouped analysis, and generating distribution-based insights such as histograms and heat maps. Table 4. lists some of these services with the recommended and additional supported visualization types. By abstracting analytical

logic into templates, the system ensures that complex operations can be reused across multiple services without duplication.

Table 4: Template services with recommended and supported visualizations.

Template Service Name	Best Visualization	Other Supported Visualization (s)
sensor-data-latest-single	KPI Card	—
sensor-data-latest-multiple	Bar Chart	Table
sensor-data-timeseries-single	Line Chart	Area Chart
sensor-data-timeseries-multiple	Line Chart	Area Chart
sensor-data-average-single	KPI Card	—
sensor-data-average-multiple	Bar Chart	Table
sensor-data-min-single	KPI Card	—
sensor-data-min-multiple	Bar Chart	Table
sensor-data-max-single	KPI Card	—
sensor-data-max-multiple	Bar Chart	Table
sensor-data-sum-single	KPI Card	—
sensor-data-sum-multiple	Bar Chart	Table
sensor-data-count-single	KPI Card	—
sensor-data-count-multiple	Bar Chart	Table
grouped-pollutant-latest-summary	Table	Bar Chart
grouped-pollutant-average-time series	Line Chart	Area Chart
grouped-pollutant-ranking	Bar Chart	Table
grouped-pollutant-stat-summary	Table	Bar Chart
grouped-pollutant-threshold-violations	Bar Chart	Table
grouped-sensor-coverage	Bar Chart	Table
grouped-pollutant-peak-summary	Bar Chart	Table
grouped-pollution-time-of-day	Line Chart	Bar Chart
pollutant-value-distribution	Bar Chart (Histogram)	Table

4.2 Platform Service Layer

Building upon the service templates, the next layer involves the creation of platform services, which bind templates to actual data resources such as sensors and projects. A platform service represents a fully configured analytical endpoint that can be consumed by applications. During its creation, developers select the appropriate template and associate it with specific sensors, pollutant categories, or geographical areas. Additional parameters, such as default time ranges or aggregation levels, can also be configured.

This layer also incorporates access control and visibility settings, ensuring that services can be securely shared across different users and applications. As a result, each platform service acts as a reusable and secure API endpoint that encapsulates both analytical logic and data access.

4.3 Visualization Binding Layer

The final stage of the implementation involves linking platform services to visualization components. This layer is responsible for transforming analytical results into visual representations that can be displayed on dashboards and applications.

When a visualization is requested, the system dynamically executes the associated platform service and retrieves the analytical results. These results are then normalized into a format suitable for rendering, ensuring compatibility with the visualization engine. The system supports a wide range of visualization types, allowing users to select the most appropriate representation for their data shown in Table 5. This dynamic binding between services and visualizations enables flexible and interactive dashboard creation without requiring manual data processing.

Table 5: Visualization type and purpose they serve.

Visualization Type	Purpose
KPI Cards	Display key metrics
Line and Area Charts	Show temporal trends
Bar Charts	Compare values across categories
Heatmaps	Visualize temporal or spatial intensity
Tables	Present detailed structured data

5 Evaluation and Validation

In this section a use case for environmental monitoring is presented to validate the results and capabilities of the middleware's data analytics and visualization module. Environmental monitoring represents a critical domain in the development of smart cities, as it directly impacts public health, urban sustainability, and policy decision-making. Urban environments are increasingly affected by air pollution, necessitating continuous monitoring of multiple environmental factors such as particulate matter (e.g., PM2.5, PM10), nitrogen oxides (NO, NO_2, NO_x), and other pollutants. Effective management of such environments requires not only data acquisition but also timely analysis and intuitive visualization to support informed decision-making by city administrators.

In practice, environmental monitoring systems are expected to provide a comprehensive set of capabilities, including real-time dashboards, key performance indicators (KPIs), temporal trend analysis, spatial comparisons across regions (e.g., boroughs or districts), and statistical summaries such as averages, minimums, and maximums. These capabilities enable decision-makers to identify pollution hotspots, analyze temporal patterns, and take corrective actions such as traffic re-routing, emission control measures, or public advisories.

To support these requirements, visualization dashboards typically incorporate multiple components, including aggregated statistical tables, time-series charts, distribution plots, and spatial representations. These components collectively enable both high-level situational awareness and detailed analytical insights. In the following sections, we demonstrate how the proposed SOMSCA middleware fulfills these requirements through integrated data analytics and visualization capabilities.

5.1 Experimental Setup

A prototype implementation of SOMSCA was developed to validate the proposed framework. The frontend was implemented using React and ECharts, while FastAPI was used to provide backend service execution and API management. Metadata was stored within MySQL, whereas TimescaleDB was employed for efficient storage and retrieval of time-series sensor observations.

The evaluation was conducted using air-quality data obtained from the London Air Quality Network. The dataset contains more than one million observations collected during the year 2025 and includes multiple

pollutants monitored across different wards and locations. The implementation environment supports service creation, analytical execution, dashboard generation, and visualization rendering, thereby enabling end-to-end validation of the proposed framework.

5.2 Dataset

To evaluate the proposed system, a large-scale environmental monitoring use case was implemented using real-world air quality data collected from the London Air Quality Network [25]. The dataset spans a full year, from 01 January 2025 to 31 December 2025, and contains over one million records. It includes multiple pollutants such as nitrogen oxides and particulate matter, collected from monitoring stations distributed across various wards in London.

Only stations with valid and continuous data were selected to ensure data quality and consistency. During dataset preparation, stations with incomplete or invalid observations were excluded to improve consistency of the evaluation dataset. The present work does not implement dedicated noise filtering, anomaly detection, or missing-value imputation techniques, as the primary objective is the evaluation of middleware-integrated proposed functionality. Investigation of robust data-quality management strategies remains an important direction for future work. After downloading the data from the relevant boroughs and wards in London, an extract-transform-load (ETL) pipeline is created using Python code. Using this ETL all the downloaded CSV files are ingested to the MySQL database. In the first step, respective sensor devices are created in the database in the same ETL process and then by integrating the sensor device to relevant readings, these are then ingested to the TimescaleDB. After this process data became available for creating the relevant services on the platform and bind those services to the user interface (UI) for visualization and data analytics.

5.3 Use Case Based Analytics and Visualizations

Once dataset is ingested in TimescaleDB, it is then used to create the platform services based on related template services to provide a range of analytical services. These services enable multiple perspectives on environmental conditions across the city.

The system supports analysis at both city-wide and localized levels, allowing users to examine overall pollution trends as well as variations across different boroughs and wards. It also enables the identification of peak pollution levels, distribution patterns, and temporal variations such as time-of-day effects. More details about the outputs is provided in the subsequent section. This multi-dimensional analysis provides a comprehensive understanding of environmental conditions and demonstrates the system's ability to support multi-level decision-making.

5.4 Visualization Outputs

The implemented dashboards present analytical results through a variety of visual components. These include key performance indicators that summarize current pollution levels, time-series charts that illustrate trends over time, and bar charts that compare pollution levels across different regions. In addition, heatmaps are used to visualize temporal patterns, while tabular views provide detailed data for further analysis.

These visualizations support administrators in making informed decisions. For instance, anomalies can be identified and investigated using line charts of sensor values. The aggregation of threshold violations can be analysed using bar charts, as shown in Fig. 3. Similarly, administrators can examine different areas within the vicinity, along with the number of sensors deployed, as illustrated in Fig. 4. Further details of these

figures are provided in the following paragraphs. Moreover, when predictive models are integrated with these visualizations, administrators can act proactively, addressing potential issues before they arise.

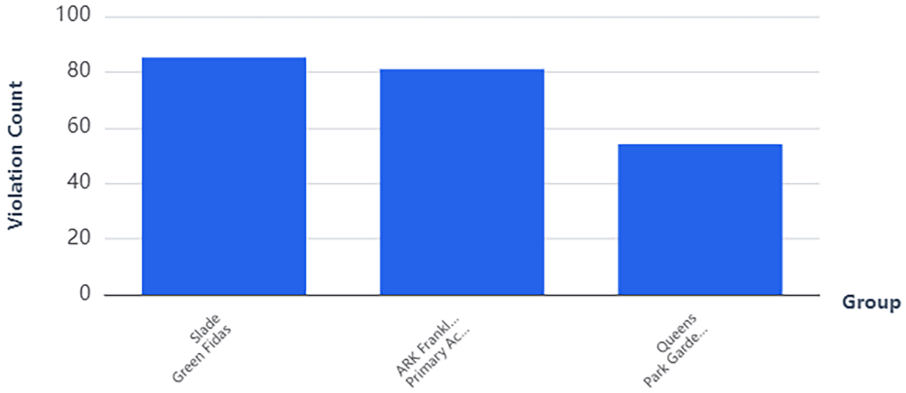

Figure 3: Total pollutant threshold violations group by ward.

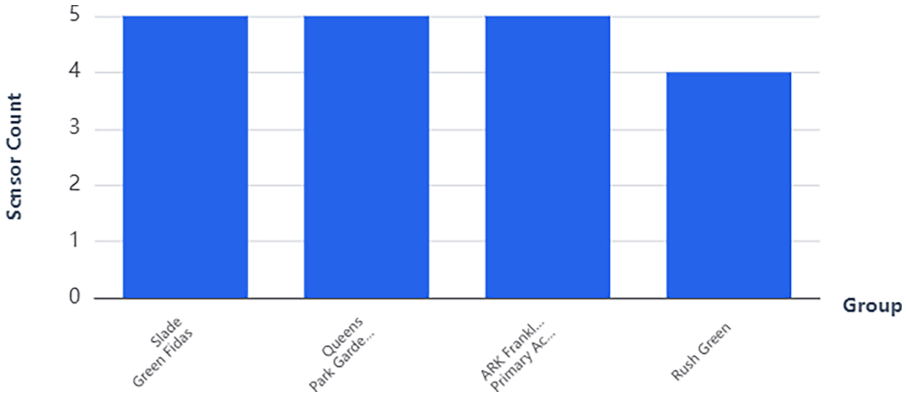

Figure 4: Number of sensors installed in each ward.

In the current use case, however, predictive models have not been incorporated. The visualizations therefore present raw and aggregated data only, without future predictions. The integration of predictive analytics, along with multi-domain monitoring to support cross-domain correlation analysis, is intended for future work.

These visualizations collectively provide both high-level summaries and detailed insights, enabling users to explore the data from multiple perspectives. Given the broad range of analytical and visualization capabilities supported by the proposed system, presenting all possible configurations and outputs is not feasible within the scope of this paper. Therefore, this section focuses on illustrating key representative components, including statistical tables, time-series charts, and distribution visualizations, which collectively demonstrate the effectiveness of the system for environmental monitoring use cases. Figs. 3–8 along with Tables 6 and 7 present selected visualization outputs generated for the environmental monitoring use case. A variety of visualization types are employed, including bar charts, line charts, and tables.

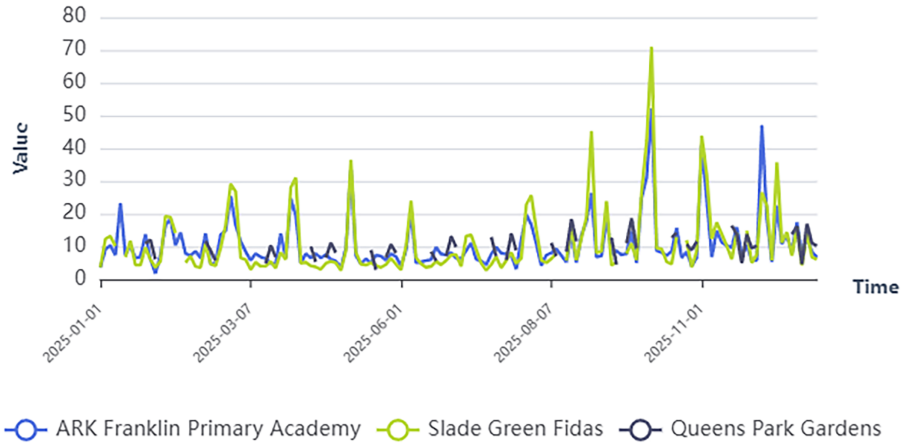

Figure 5: Average value time series group by ward.

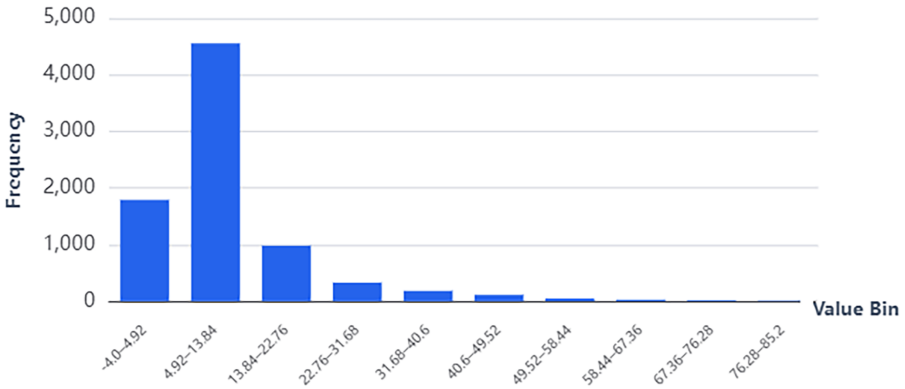

Figure 6: Pollutant value distribution grouped in range bins.

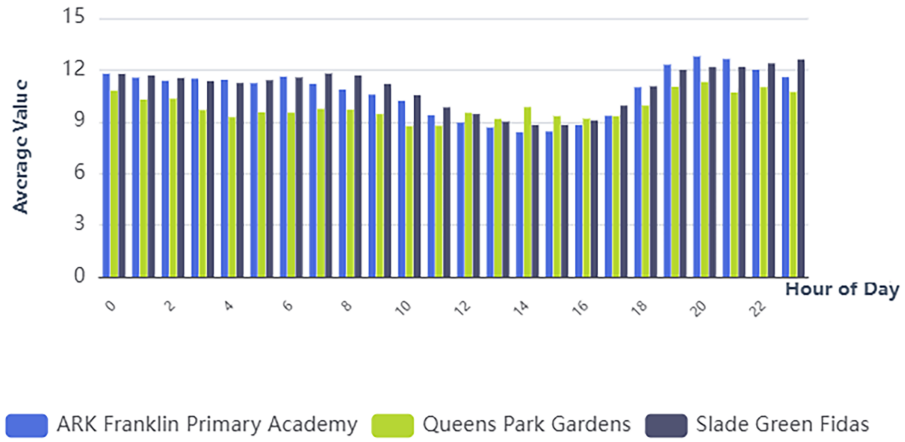

Figure 7: Average value of pollutants group by ward and time-of-day.

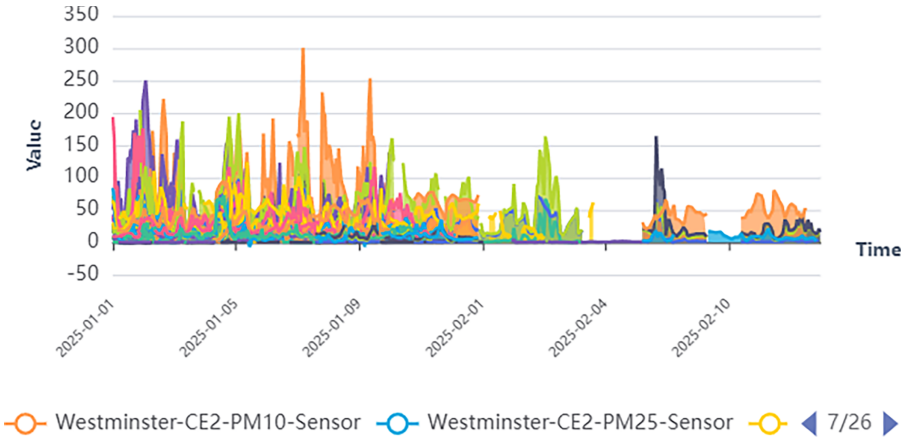

Figure 8: Time series line chart of selected sensors.

Table 6: Pollutant stat summary group by ward.

Group Value	Avg Value	Min Value	Max Value	Sample Count
Slade Green Fidas	10.91	0.4	85.2	3273
ARK Franklin Primary Academy	10.7	−4	81	3443
Queens Park Gardens	9.87	−2	54	1355

Table 7: Peak value and peak value time for each ward.

Group Value	Peak Value	Peak Time
Slade Green Fidas	85.2	2025-10-03 09:00:00
ARK Franklin Primary Academy	81	2025-12-01 00:00:00
Queens Park Gardens	54	2025-09-11 00:00:00

For all visualizations, the data is grouped at the ward level, illustrating sensor devices deployed within each ward along with their aggregated statistical summaries, value ranges, and raw time-series data. In all the visualization outputs, the data corresponds to the year 2025, is grouped by ward, and is filtered for the PM2.5 pollutants except for the Fig. 8, which includes the sensors of different pollutants along with PM2.5. The details of the individual visualizations are described as follows:

Table 6 presents a tabular representation of each ward, including the total number of samples collected by sensors, along with their average, minimum, and maximum values. Likewise, Table 7 shows the maximum observed value in each ward along with the corresponding timestamp.

In Fig. 3, a bar chart illustrates the violation count across different wards. The violation count represents the number of observations exceeding a predefined threshold level. This visualization is generated using the gp-pollutant-threshold-violations template service, which is then instantiated as a platform service by attaching relevant sensor devices based on the specified filters. Similarly, Fig. 4 presents a bar chart showing

the number of sensor devices deployed in each ward. This visualization is created using the gp-sensor-coverage template service, followed by the application of filters to associate the relevant sensors and generate the corresponding platform service.

In Figs. 5–8, multiple visualizations are presented using both line and bar charts. Fig. 5 illustrates a line chart showing the average sensor readings over time. This visualization is generated using the gp-pollutant-average-time series template service, with the date range covering the entire year 2025. Similarly, Fig. 6 presents a bar chart depicting the distribution of sensor values, created using the gp-pollutant-value-distribution template service. Fig. 7 utilizes the gp-pollution-time-of-day template service to display the average pollution levels for each hour across different wards.

Finally, Fig. 8 employs the sensor-data-time series multiple template service to plot raw sensor readings by selecting multiple sensors from different wards. For demonstration purposes, sensors measuring multiple pollutants are used. However, this visualization can also be utilized to analyze trends of a single pollutant across different areas of the city by selecting only the relevant sensors, for example, by filtering for PM2.5 or NO.

The presented results demonstrate that the proposed SOMSCA middleware effectively fulfills the core requirements of an environmental monitoring system. The integration of data acquisition, analytics, and visualization within the middleware layer enables seamless generation of key performance indicators, statistical summaries, and interactive visualizations. The system supports both temporal and spatial analysis, allowing city administrators to monitor pollution trends, compare regions, and identify critical conditions in a timely manner.

Furthermore, the ability to dynamically select sensors, apply aggregation functions, and generate multi-dimensional visualizations highlights the flexibility and extensibility of the proposed approach. These capabilities are essential for real-world smart city deployments, where data heterogeneity and scalability are significant challenges.

Although this study focuses on environmental monitoring as a representative use case, the same architectural principles and service-oriented design can be applied to other smart city domains, such as traffic management, energy optimization, healthcare monitoring, and infrastructure management. This demonstrates the generality of SOMSCA as a unified middleware solution for integrated data analytics and visualization across diverse urban applications.

6 Discussion

The implementation of SOMSCA demonstrates several significant advantages in the context of smart city application development. The system exhibits a high degree of flexibility, allowing developers to dynamically select data sources, analytical operations, and visualization types without modifying the underlying code. This flexibility is particularly important in smart city environments, where requirements frequently evolve. The template-based approach greatly simplifies development by eliminating the need to repeatedly implement similar analytical logic. As a result, applications can be developed more rapidly and with reduced effort.

In term of cost perspective, the integration of proposed module within the middleware eliminates the need for separate analytical infrastructures, leading to reduced operational and development costs. Furthermore, the system reduces dependency on specialized skill sets. Developers without expertise in data analytics or visualization can still build sophisticated applications by leveraging predefined services and templates. Finally, the service-oriented architecture ensures scalability, enabling the system to handle large datasets and support multiple concurrent users without performance degradation.

Although security, privacy, and access-control services form part of the broader SOMSCA architecture, their implementation and evaluation were not the primary focus of this study. Future work will investigate authentication mechanisms, privacy-preserving analytics, secure service execution, and resilience against cyber threats commonly encountered in smart city environments.

Limitations

Although the proposed system demonstrates the feasibility of integrating analytics and visualization within a middleware environment, several limitations remain. First, the current evaluation focuses on a single environmental monitoring use case and does not yet include quantitative benchmarking of latency, throughput, or resource consumption. Second, while the architecture is designed to support scalability through reusable services and modular deployment, large-scale stress testing under high concurrency workloads remains future work. Finally, the present implementation focuses primarily on structured sensor data and does not yet address multimedia or unstructured data sources. These limitations provide opportunities for future investigation and broader validation.

In addition, the current study does not include formal usability evaluation involving end users, domain experts, or city administrators. Although the dashboards demonstrate the ability to present analytical results through multiple visualization formats, future studies should investigate usability, decision-support effectiveness, and user interaction performance through structured user studies.

7 Conclusion & Future Direction

This study presented a middleware-integrated analytics and visualization framework based on SOMSCA and demonstrated how analytical processing and visualization generation can be incorporated directly within the middleware layer through reusable services. The proposed approach combines service-oriented analytics, data-type-oriented visualization selection, and template-assisted dashboarding to reduce development complexity and improve reuse across smart city applications. Validation using a large-scale environmental monitoring use case demonstrated the practical feasibility of the framework and its ability to support dynamic analytics and visualization generation using real-world sensor data.

From a practical perspective, the proposed approach enables developers and city administrators to create analytical applications without repeatedly implementing analytical logic and visualization pipelines. By exposing these capabilities as reusable middleware services, the framework supports more efficient development of intelligent smart city solutions.

The future work will focus on extending the SOMSCA to support text-based analytics, enabling analysis of unstructured data such as user feedback and social media content. In addition to this, the integration of video and image analytics may allow the system to process multimedia data from sources such as surveillance cameras. Advanced geospatial visualization techniques will be explored to provide more detailed spatial insights, while cross-domain analytics may enable the correlation of data. Finally, the incorporation of real-time anomaly detection will enhance the system's ability to identify and respond to critical events as they occur. Future work will also include systematic performance benchmarking, including latency, throughput, scalability, and resource utilization analysis under varying workloads. In addition, the middleware will be evaluated across multiple smart city domains, including transportation, energy management, and healthcare monitoring, to further validate its generality and applicability. Future extensions will also investigate predictive analytics, anomaly detection, privacy-preserving analytics, and distributed intelligence approaches such as federated learning.

Acknowledgement: AI-assisted tools were used solely for language editing, including improvements to grammar, sentence structure, readability, and overall clarity of the manuscript. These tools were not used for study design, methodology development, data processing, model development, experimentation, result generation, analysis, interpretation, or conclusions. All scientific content and findings are the original work of the author. The author carefully reviewed and verified all AI-assisted edits to ensure accuracy, originality, and compliance with academic and ethical standards.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Zulfiqar Ali, Azhar Mahmood; methodology, Zulfiqar Ali, Azhar Mahmood, Shaheen Khatoon, Seyed Ali Ghorashi; validation, Zulfiqar Ali, Azhar Mahmood; formal analysis, Azhar Mahmood, Shaheen Khatoon, Seyed Ali Ghorashi; investigation, Zulfiqar Ali, Azhar Mahmood, Shaheen Khatoon; writing & original draft preparation, Zulfiqar Ali, Azhar Mahmood; writing & review, Shaheen Khatoon, Seyed Ali Ghorashi; editing, Azhar Mahmood, Shaheen Khatoon. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The original data presented in the study is publicly available at https://www.londonair.org.uk/.

Ethics Approval: Not applicable.

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

List of abbreviations

Abbreviation	Description
SOMSCA	Service-Oriented Middleware for Smart City Applications
IoT	Internet of Things
API	Application Programming Interface
KPI	Key Performance Indicator
ETL	Extract, Transform and Load
AQI	Air Quality Index
PM2.5	Particulate Matter ≤2.5 μm
PM10	Particulate Matter ≤10 μm
NO	Nitric Oxide
NO_2	Nitrogen Dioxide
NO_x	Nitrogen Oxides
LAQN	London Air Quality Network

References

1. Murugan R, Palanichamy N. Smart city air quality prediction using machine learning. In: Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS); 2021 May 6–8; Madurai, India. p. 1048–54. doi:10.1109/iciccs51141.2021.9432074.
2. Sonawane P, Dhanawade S, Barangule V, Kulkarni A, Mahalle P. Air quality analysis & prediction using machine learning: Pune smart city case study. In: Proceedings of the 2023 IEEE 8th International Conference for Convergence in Technology (I2CT); 2023 Apr 7–9; Lonavla, India. p. 1–6. doi:10.1109/i2ct57861.2023.10126304.
3. Alourani A, Alam M, Ali A, Khan IR, Samal CK. Hybrid AI-IoT framework with digital twin integration for predictive urban infrastructure management in smart cities. Comput Mater Contin. 2026;86(1):1–32. doi:10.32604/cmc.2025.070161.
4. dos Santos D, Geyer CFR, Dos Anjos JCS, Sousa AB, Donsez D, Leithardt VRQ. MiddleFog: a middleware for protocol interoperability in heterogeneous IoT environments. TechRxiv. 2024. doi:10.36227/techrxiv.170792976.68508243/v1.

5. Cavalcanti D, Rosa N. Customizable and adaptable middleware of things. Int J Commun. 2024;37(15):e5887. doi:10.1002/dac.5887.
6. Del Esposte AMD, Kon F, Costa FM, Lago N. InterSCity: a scalable microservice-based open source platform for smart cities. In: Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems. Setubal, Portugal: SCITEPRESS; 2017. p. 35–46.
7. Puiu D, Barnaghi P, Tonjes R, Kumper D, Ali MI, Mileo A, et al. CityPulse: large scale data analytics framework for smart cities. IEEE Access. 2016;4:1086–108. doi:10.1109/access.2016.2541999.
8. Ali Z, Mahmood A, Khatoon S, Alhakami W, Ullah SS, Iqbal J, et al. A generic Internet of Things (IoT) middleware for smart city applications. Sustainability. 2023;15(1):743. doi:10.3390/su15010743.
9. Pereira J, Batista T, Cavalcante E, Souza A, Lopes F, Cacho N. A platform for integrating heterogeneous data and developing smart city applications. Future Gener Comput Syst. 2022;128(1–2):552–66. doi:10.1016/j.future.2021.10.030.
10. Perera C, Zaslavsky A, Christen P, Georgakopoulos D. CA4IOT: context awareness for Internet of Things. In: Proceedings of the 2012 IEEE International Conference on Green Computing and Communications; 2012 Nov 20–23; Besancon, France. p. 775–82. doi:10.1109/greencom.2012.128.
11. Bonte P, Ongenae F, De Backere F, Schaballie J, Arndt D, Verstichel S, et al. The MASSIF platform: a modular and semantic platform for the development of flexible IoT services. Knowl Inf Syst. 2017;51(1):89–126. doi:10.1007/s10115-016-0969-1.
12. Rodríguez-Molina J, Martínez JF, Castillejo P, de Diego R. SMArc: a proposal for a smart, semantic middleware architecture focused on smart city energy management. Int J Distrib Sens Netw. 2013;9(12):560418. doi:10.1155/2013/560418.
13. Mohamed N, Al-Jaroodi J, Jawhar I, Lazarova-Molnar S, Mahmoud S. SmartCityWare: a service-oriented middleware for cloud and fog enabled smart city services. IEEE Access. 2017;5:17576–88. doi:10.1109/access.2017.2731382.
14. Apolinarski W, Iqbal U, Parreira JX. The GAMBAS middleware and SDK for smart city applications. In: Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS); 2014 Mar 24–28; Budapest, Hungary. p. 117–22. doi:10.1109/PerComW.2014.6815176.
15. Bellur U, Narendra NC, Mohalik SK. AUSOM: autonomic service-oriented middleware for IoT-based systems. In: Proceedings of the 2017 IEEE World Congress on Services (SERVICES); 2017 Jun 25–30; Honolulu, HI, USA. p. 102–5. doi:10.1109/services.2017.25.
16. Jeon S, Jung I. MinT: middleware for cooperative interaction of things. Sensors. 2017;17(6):1452. doi:10.3390/s17061452.
17. Haq HBU, Akram W, ur Rashid Kayani H, Mahmood K, Shih C, Kharel R, et al. A deep dive into anomaly detection in IoT networks, sensors, and surveillance videos in smart cities. Comput Mater Contin. 2026;87(2):4–10. doi:10.32604/cmc.2025.073188.
18. Tavares JMRS, Karri C, Machado JJM, Jain DK, Dannana S, Gottapu SK, et al. Recent technology advancements in smart city management: a review. Comput Mater Contin. 2024;81(3):3617–63. doi:10.32604/cmc.2024.058461.
19. Ghadi YY, Mazhar T, Shah SFA, Haq I, Ahmad W, Ouahada K, et al. Integration of federated learning with IoT for smart cities applications, challenges, and solutions. PeerJ Comput Sci. 2023;9(5):e1657. doi:10.7717/peerj-cs.1657.
20. React—a javascript library for building user interfaces. 2025 [cited 2025 Dec 7]. Available from: https://react.dev.
21. Apache ECharts. 2026 [cited 2026 Mar 30]. Available from: https://echarts.apache.org/en/index.html.
22. Ramírez S. FastAPI—fast and modern web API framework for python. 2025 [cited 2025 Dec 7]. Available from: https://fastapi.tiangolo.com.
23. Oracle I. MySQL. 2026. [cited 2026 Mar 30]. Available from: https://www.mysql.com/.
24. TimescaleDB—time-series database on PostgreSQL. 2025 [cited 2025 Dec 7]. Available from: https://www.timescale.com.
25. London air quality data. 2026 [cited 2026 Mar 15]. Available from: https://www.londonair.org.uk/.