Check for updates

RESEARCH ARTICLE

REVISED Edge-Cloud Architectures for Urban Mobility and Safety

[version 2; peer review: 3 approved with reservations, 1 not approved]

Enrico Rossini[1], Marcello Pietri[1], Marco Picone[1], Natalia Selini Hadjidimitriou[1], Marco Mamei[1], Panagiotis Moraitis[2], Petros Zervoudakis[2]

[1]DISMI, University of Modena and Reggio Emilia, Reggio Emilia, Italy
[2]Netcompany, Athens, Greece

v2 **First published:** 06 Jan 2026, **6**:14
https://doi.org/10.12688/openreseurope.22191.1
Latest published: 13 Mar 2026, **6**:14
https://doi.org/10.12688/openreseurope.22191.2

Abstract

This paper introduces the NOUS Smart City Architecture (NSCA), an extensible middleware designed to enable intelligent, interoperable urban services across the edge–cloud continuum. Developed within the NOUS project, NSCA addresses a central challenge in urban digitalization: supporting scalable, efficient communication among heterogeneous actors while integrating with emerging data space infrastructures. On the **edge side**, NSCA builds on the lightweight and widely adopted MQTT protocol, enabling reliable, low-latency, and semantically structured communication among distributed assets such as edge devices, vehicles, roadside units, and localized digital services. On the **cloud side**, it integrates a Data Space Ecosystem based on the SIMPL Open architecture, providing a scalable foundation for cross-stakeholder data exchange and governance. The two layers are bridged through a standard MQTT inter-broker connector that can be deployed on either edge nodes or cloud hosts, offering flexibility in how topics are forwarded, filtered, and federated. NSCA supports both real-time data dissemination and service composition at scale, allowing new applications to be integrated with minimal configuration. This makes the architecture suitable for a wide range of smart city verticals, including road safety, mobility management, environmental monitoring, and citizen services. The platform's practical value is demonstrated through two operational services deployed in a real-world smart city testbed. The **Vulnerable Road User (VRU) Safety Service** leverages edge-to-cloud communication to detect pedestrians, cyclists, and other users at risk, issuing proactive collision warnings based on real-time situational awareness. The **GeoPerception Service** enables cooperative sensing by sharing localized perception among vehicles and smart

Open Peer Review

Approval Status ✗ ? ? ?

	1	2	3	4
version 2 (revision) 13 Mar 2026	✗ view		? view	? view
version 1 06 Jan 2026	? view	? view		

1. **Shiva Shankar Reddy**, Sagi Rama Krishnam Raju Engineering College (A), Bhimavaram, India
2. **Eirini Eleni Tsiropoulou**, Arizona State University, Tempe, USA
3. **Antonio Puliafito**, University of Messina, Messina, Italy
4. **Valter Vieira Camargo**, Computing Department, UFSCar - Universidade Federal de São Carlos, São Carlos, Brazil

Any reports and responses or comments on the article can be found at the end of the article.

infrastructures, improving environmental understanding and mobility intelligence. Together, these services showcase NSCA's ability to integrate IoT data flows, edge analytics, and data-space–enabled cloud services, supporting next-generation, interoperable urban applications.

Keywords
Edge computing, Cloud computing, Multi-Access Edge Computing, Mobility, Safety, Smart City, Data Spaces



This article is included in the Horizon Europe gateway.

Corresponding author: Marco Mamei (marco.mamei@unimore.it)

Author roles: Rossini E: Conceptualization, Methodology, Software, Writing – Review & Editing; **Pietri M**: Conceptualization, Software, Writing – Review & Editing; **Picone M**: Conceptualization, Software, Supervision, Writing – Review & Editing; **Hadjidimitriou NS**: Conceptualization, Funding Acquisition, Investigation, Project Administration, Writing – Review & Editing; **Mamei M**: Conceptualization, Funding Acquisition, Investigation, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing; **Moraitis P**: Conceptualization, Funding Acquisition, Project Administration, Writing – Review & Editing; **Zervoudakis P**: Conceptualization, Project Administration, Software, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135927 (A catalyst for EuropeaN ClOUd Services in the era of data spaces, high-performance and edge computing - NOUS)
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Rossini E *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License , which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Rossini E, Pietri M, Picone M *et al.* **Edge-Cloud Architectures for Urban Mobility and Safety [version 2; peer review: 3 approved with reservations, 1 not approved]** Open Research Europe 2026, **6**:14
https://doi.org/10.12688/openreseurope.22191.2

First published: 06 Jan 2026, **6**:14 https://doi.org/10.12688/openreseurope.22191.1

REVISED **Amendments from Version 1**

The revised version of the manuscript incorporates the corrections suggested by the reviewers. The Introduction and Related Work section have been expanded, incorporating further bibliographic references. Technical aspects of the architecture and experimental set-up have been clarified. In the new version of the article, an extensive grammar check is done, improving the clarity and coherence of the manuscript.

Any further responses from the reviewers can be found at the end of the article

Introduction

The rapid evolution of modern cities into complex cyber-physical ecosystems has created the need for two different classes of services. On the one hand, there are real-time low-latency services associated with road safety, autonomous driving, real time communications, etc. On the other hand, there are complex multi-stakeholder applications focusing on heterogeneous data exchange and analysis: public services, energy optimization, e-governments, etc. Therefore, the smart city infrastructure should support such often contrasting requirements. Physical assets, mobile users, and digital services are increasingly required to interact seamlessly across distributed environments, generating and consuming large volumes of data under strict latency and reliability constraints. At the same time, stakeholders must engage in complex interaction protocols that involve heterogeneous and large-scale data sources.

Within this scenario, the NOUS framework aims at creating a novel architecture for edge-cloud services supporting low-latency applications as in smart cities and safety scenarios that extends to trusted, regulated and complex-multi party interactions via cloud data exchange.

We applied the NOUS framework to a concrete real-world smart city testbed: the Modena Automotive Smart Area (MASA). MASA is a living laboratory for smart urban innovation. Located in Modena, Italy, MASA integrates diverse connected assets—including Smart Cameras, Roadside Units (RSUs), vehicular on-board units, and personal mobile devices. These components operate across multiple layers of an edge–cloud continuum, ranging from low-power edge devices to Multi-Access Edge Computing (MEC) nodes and federated Cloud services, enabling large-scale experimentation with cooperative, data-driven services for smart cities.

However, managing communication and coordination in such a heterogeneous, dynamic, and distributed environment poses significant challenges. Systems must remain modular and evolvable while preserving interoperability, scalability, and low-latency performance. To address these challenges, this paper introduces the NSCA, a flexible, event-driven middleware designed to support intelligent interactions among heterogeneous urban entities distributed along the edge–cloud continuum. In particular, it supports low-latency event-based communication on the edge side, and it supports data exchange through data spaces among multiple stakeholders on the cloud side.

More in detail, NSCA revolves around two key principles:

On the edge side, one of the key architectural principles of NSCA is logical decoupling between data producers and consumers, enabling systems and services to evolve independently. To achieve this, NSCA adopts an event-driven paradigm based on the Publish/Subscribe (Pub/Sub) communication model, which naturally supports asynchronous, many-to-many interactions among distributed components. This model promotes high modularity, fault tolerance, and responsiveness, making it particularly suitable for latency-sensitive scenarios, where computation must often be executed close to data sources at the edge, while still enabling integration with cloud analytics and decision-making components. NSCA implements this paradigm through a structured messaging layer that organizes interactions into semantic categories such as telemetry, events, actions, and service-level exchanges. This semantic structuring provides fine-grained control over data flows, allowing selective subscriptions, hierarchical topic organization, and dynamic service discovery mechanisms. Furthermore, NSCA adopts an edge–cloud broker deployment strategy, enabling local message routing and processing for time-critical services, while preserving interoperability with remote observers, data aggregators, and third-party services.

On the cloud side, NCSA adopts the Data Space framework as a core architectural principle to enable secure, scalable, and sovereign data sharing across organizations. A Data Space is a distributed and federated ecosystem that allows independent stakeholders to exchange data without relinquishing control over their assets, while ensuring transparency and contractual clarity. In contrast to centralized approaches based on data lakes or data warehouses, a Data Space preserves data sovereignty by keeping data close to its original owner and enabling controlled access through standardized policies, rather than physical data consolidation. To support this paradigm, NCSA integrates

interoperability mechanisms at multiple levels—syntactic interoperability via shared APIs and communication protocols; semantic interoperability through standardized data models and ontologies; and organizational interoperability enforced by governance rules and data usage agreements. Trust is established through distributed identity management, certification mechanisms, and usage control frameworks that regulate how data can be accessed, reused, and combined into value-added services. Furthermore, NCSA enables federated data processing and AI workflows, allowing cloud services to operate on distributed datasets while respecting access permissions and usage constraints. This approach facilitates cross-domain integration—such as mobility, safety, logistics, energy and sustainability—while aligning with European initiatives like GAIA-X and the Simpl-Open, which promote openness, transparency, and digital sovereignty in data ecosystems.

The combination of these two layers allow to create complex multi-stakeholder applications for smart city data, while retaining flexibility and low-latency at the edge level to support real-time applications.

This paper provides a detailed description of the NSCA communication model, along with its topic structure, interaction semantics, and deployment strategy. The architecture is validated through two real-world services deployed within MASA:

- The Vulnerable Road User (VRU) Safety Service, which detects and prevents collision risks involving pedestrians and cyclists by leveraging cooperative sensing and real-time alerts
- The GeoPerception Service, which combines data from heterogeneous sensors to deliver localized environmental awareness and statistics on city mobility to infrastructure and users.

These services demonstrate how NSCA supports scalable, event-driven coordination and enables the deployment of intelligent, data-driven smart city applications that can enhance safety, mobility efficiency, and situational awareness in complex urban environments.

This paper is organized as follows: Section 2 presents the main NSCA edge components. Section 3 presents the main NCSA cloud components. Section 4 presents reference applications. Section 5 presents related work in the area. Section 6 concludes and presents future work.

Nous Smart City Architecture (NSCA) - Edge

The NSCA-edge is engineered to support real-time, flexible, and scalable communication among distributed components operating within complex urban environments. Its design combines a layered, event-driven messaging system with a hybrid deployment strategy that integrates Edge and Cloud brokers. This hybrid approach enables low-latency local computation for time-critical services, while simultaneously supporting wide-area data dissemination for monitoring, analytics, coordination, and remote service provisioning.

At the core of NSCA lies a distributed event-driven communication fabric that enables asynchronous, decoupled interactions among heterogeneous clients. These include:

- Physical assets, such as smart cameras, roadside units, and mobile devices;
- Software services, including perception modules, risk detection engines, and decision-support logic;
- External observers, such as dashboards, data aggregators, and supervisory control systems.

Although the architecture is designed to be protocol-agnostic and technology-independent, the current prototype instantiates NSCA using the MQTT (Message Queuing Telemetry Transport) protocol due to its lightweight footprint, reliability over unreliable networks, and wide adoption in IoT ecosystems. Communication among these entities follows the publish/subscribe (Pub/Sub) model, which promotes logical decoupling between message producers and consumers. This paradigm supports system modularity, elastic scalability, service composability, and resilience to dynamic network conditions, all of which are necessary for real-time smart city applications.

Hybrid edge–cloud deployment

To meet heterogeneous Quality of Service (QoS) requirements, SCA employs a multi-broker architecture:

- *Edge Broker* – Deployed within local domains (e.g., MEC nodes or micro data centers), it handles latency-critical communication, allowing localized interactions among nearby assets and services. This is essential for safety-critical use cases, such as real-time Vulnerable Road User (VRU) risk alerts or cooperative perception.

- *Cloud Broker* – Deployed in the public cloud or institutional data centers, it enables global service access, long-term storage, and support for cross-domain applications, including fleet monitoring, digital twin synchronization, and historical analytics. It connects the Edge part with NSCA-cloud and data spaces (see next Section).

To ensure transparent interoperability between these two layers, brokers are connected through a bridge mechanism. This allows selective message replication, topic forwarding, and cross-domain routing, enabling data to remain local by default while being escalated to the cloud when necessary for supervision or large-scale aggregation. In practical terms, the bridge operates as a native MQTT inter-broker link that forwards only the topics explicitly configured for propagation. Each broker maintains its own namespace, and the bridge maps selected topic trees from the edge broker to the cloud broker using configurable include/exclude rules. Broker-level selection and topic forwarding within NSCA are configured through standard EMQX configuration files, with the initial setup typically performed via the broker's graphical administration interface. In contrast, the dynamic configuration of assets, services, and observers — across both edge and cloud environments — is managed through REST APIs exposed by NSCA and a dedicated web application that interacts with these interfaces. This separation of concerns allows low-level broker routing and infrastructure policies to remain stable and predictable, while higher-level system behavior can be adjusted at runtime without modifying broker configurations. As a result, NSCA maintains operational robustness at the communication layer while enabling flexible orchestration and lifecycle management through application-level services.

The multi-broker architecture prevents full-mesh synchronization and ensures that only the required subtopics—such as aggregated events, service outputs, or supervisory metadata—are transmitted upstream. The mechanism also supports bidirectional forwarding, allowing cloud-based services to publish commands or configuration updates that are injected into the edge broker with controlled scope. By relying on MQTT's built-in session handling, QoS policies, and retained messages, the bridge guarantees consistent delivery semantics across domains without modifying client applications. This enables scalable, selective routing between brokers while keeping the messaging load tightly bounded at the edge.

A distinguishing feature of NSCA is its semantic topic hierarchy, which structures message flows according to functional roles and data scope. This design enables: (i) Fine-grained topic filtering for efficient message routing; (ii) Service discovery via structured topic introspection; (iii) Action invocation and response routing based on command semantics; (iv) Logical segregation of communication categories (telemetry, events, commands, and service responses). To ensure consistency and traceability in dense deployments, NSCA enforces naming conventions and message metadata standards. Each asset is uniquely referenced using `<asset_type>/<asset_id>`, while services register using standardized descriptors. Action-response interactions are tracked using request identifiers (request_id) to maintain reliability in asynchronous command execution. Together, these design choices enable NSCA to serve as a modular, evolvable, and MQTT-native middleware tailored to the communication demands of smart urban ecosystems, supporting thousands of concurrent entities distributed across heterogeneous networks.

Topic namespace collisions in NSCA are prevented through a tenant-based prefixing strategy applied to the topic hierarchy. Each participating organization — such as a municipality or city district — is assigned a unique tenant identifier that is prepended to all topics forwarded toward the cloud broker (i.e., topics follow the structure `/<tenant>/` …). This approach allows independent edge deployments to maintain overlapping internal topic structures while ensuring global uniqueness and isolation once data is exposed at the cloud level. Currently, tenant prefixes are configured at the broker and application layers, while strict enforcement through the identity and access-control framework is still under consolidation and represents an area of ongoing development. In addition, seamless user migration across edge domains — potentially spanning multiple tenants — is being explored as part of future platform extensions to support more dynamic mobility scenarios and cross-city interoperability.

Client categorization and interaction model

To organize communication and interaction within the NSCA ecosystem, all connected entities are categorized into three functional client roles, based on their operational purpose and interaction semantics:

1. *Assets* – These are NSCA clients representing physical or logical entities that produce data streams or accept control commands. Typical examples include environmental sensors, wearable devices, connected vehicles, mobile applications, and smart cameras. Each asset is uniquely identified using an `<AssetType>/<AssetId>` convention, which ensures consistent naming and traceability across deployments. Assets primarily publish telemetry, status information, and event notifications, and may subscribe to action commands issued by services.

2. *Services* – Services are active computational modules that process incoming data, implement operational logic, and generate outputs for consumption by other system participants. SCA distinguishes between two types of services:

    - Generic Services, which are openly accessible without prior registration (e.g., weather information, map services);

    - Asset Services, which require explicit registration per assets and provide context-aware, client-specific outputs, such as personalized VRU safety alerts or cooperative perception feedback. Services publish their availability and interaction contracts through dedicated topic channels, enabling dynamic discovery and on-demand interaction.

3. *Observers* – Observers are read-only clients that subscribe to specific topic hierarchies for monitoring, auditing, or visualization purposes. Examples include control room dashboards, system supervisors, digital twin interfaces, and logging tools. Though they do not actively influence system behavior, observers play a crucial role in situational awareness and system transparency. More in detail, the observer role acts as a mediating layer that continuously monitors asset behavior and service interactions across the messaging fabric. Rather than directly influencing system execution, observers collect telemetry, state changes, and event notifications from assets and interpret these signals to support coordination and situational awareness. By tracking variations in asset activity and service outputs, observers can propagate relevant updates to downstream components and facilitate the dynamic association between assets and services that expose compatible capabilities. In this sense, the observer functions as an interface between physical entities and logical processing modules, enabling coherent, responsive, and adaptive interactions across the edge–cloud continuum while preserving the decoupled nature of the publish/subscribe model.

NSCA adopts a flexible interaction paradigm in which clients are not bound to a single predefined function but can dynamically participate with multiple roles at the same time. In particular, individual services and applications may concurrently act as assets (publishing telemetry or events), services (processing data and generating outputs), and observers (subscribing to selected topics for monitoring or analytics), depending on the operational context. This multi-role capability reflects real smart-city deployments, where components often combine sensing, computation, and supervision within a single logical entity. By enabling concurrent role participation, NSCA reduces architectural fragmentation, simplifies service composition, and supports more adaptive interaction patterns across the edge–cloud continuum without introducing additional orchestration layers or rigid client separation.

Communication among these entities is orchestrated through the NSCA broker infrastructure, which provides message routing, service coordination, and lifecycle management. Supported interaction patterns include:

- *Service Registration* – Assets register to Asset Services through dedicated registration topics, enabling per-client service sessions.

- *Service Publication* – Service availability and metadata are published as retained information messages, allowing late joiners to discover active services.

- *Service Notification* – Services publish tailored outputs to asset-specific channels (e.g., risk_level for VRU alerts).

- *Action Invocation* – Assets and observers may invoke commands on services, with optional request/response correlation via unique identifiers.

These message flows are illustrated in Figure 1, which depicts the information exchange across the MASA Edge–Cloud Continuum, showing how assets, services, and observers interact via telemetry streams, events, actions, service

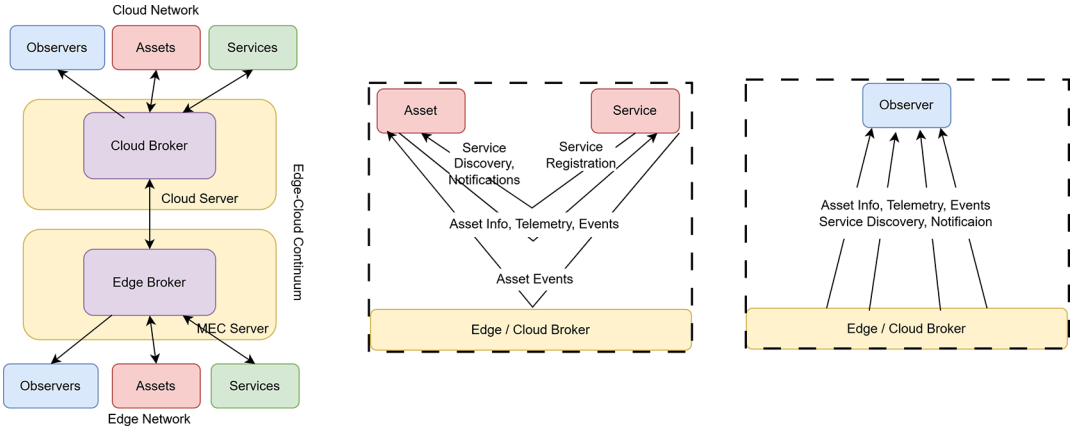


Figure 1. Edge Cloud broker in NOUS. The left side illustrates the hybrid edge–cloud deployment, where an edge broker and a cloud broker are interconnected through an inter-broker bridge that enables selective topic forwarding, bidirectional message propagation, and controlled data escalation from local domains to cloud services. This mechanism allows latency-sensitive interactions to remain at the edge while exposing aggregated events and supervisory data to remote observers and data space components. The right side depicts the logical interaction model among NSCA clients: assets publish telemetry and events, services perform processing and generate notifications, and observers subscribe to selected topic hierarchies to monitor system behavior. The diagram highlights how semantic topics support service discovery, registration workflows, action invocation, and asynchronous notifications, enabling decoupled communication across heterogeneous urban entities without direct dependencies between producers and consumers.

discovery, and notifications across edge and cloud brokers. The left part of the figure shows the edge–cloud bridge mechanism connecting the edge broker and the cloud broker, enabling selected topics to be forwarded across domains while keeping most traffic local to the edge. The right part illustrates the interaction between assets and services—covering service discovery, registration, and event exchange—as well as the observer role, which can subscribe to asset telemetry and service notifications to monitor the system without affecting its operation.

Message topic design and flow model

The NSCA communication fabric is organized around a semantic topic hierarchy that structures message flows according to functional categories such as telemetry, events, actions, and service coordination. Topics follow a hierarchical naming convention that separates concerns while enabling efficient routing and interoperability. Although natively implemented using MQTT, the topic conventions are technology-agnostic and can be mapped to other event-driven messaging solutions—such as ZeroMQ or AMQP—through routing key translation. Additional subtopics enable resource-level filtering, allowing clients to subscribe only to relevant data without processing unnecessary streams. This organization results in bandwidth-efficient communication and supports fine-grained access control, both essential for large-scale smart city deployments.

The NSCA topic model also supports stateful and asynchronous interactions through: (i) Retained messages, which allow new clients to retrieve the latest state immediately upon subscription; (ii) Request/response subtopics, which enable service acknowledgements and correlation of asynchronous workflows. Payloads are encoded using JSON enriched with schema definitions and version tags, ensuring extensibility and backward compatibility. Each message carries structural metadata enabling consistent parsing, system observability, and message tracing across distributed brokers.

Together, the topic-driven architecture and broker-layer deployment strategy allow SCA to support high-frequency, low-latency communication required by mission-critical smart city services, such as real-time safety systems or cooperative perception, while preserving interoperability with external platforms and future integration with data space architectures and event-driven orchestration frameworks.

Dynamic plugin and filtering

In the context of NSCA, edge intelligence refers to a set of distributed capabilities embedded within the edge infrastructure that continuously observe, analyze, and act on data streams close to their sources. These capabilities enable cooperative behaviors across edge nodes while remaining transparent to applications and services operating on top of the communication layer. Rather than embedding intelligence directly into client logic, NSCA introduces coordination mechanisms within the messaging and orchestration infrastructure itself, allowing the system to adapt dynamically

without modifying application code. Functions such as dynamic plugins, message observation, semantic filtering, and inter-broker coordination operate at the broker and infrastructure level, enabling the platform to analyze, enrich, aggregate, or redirect data flows in a transparent and scalable manner across the edge–cloud continuum.

Dynamic plugin and filtering capabilities extend the NSCA messaging infrastructure by enabling broker-level extensibility that can be configured and adapted at runtime rather than being fixed at design time. Through programmable plugins and declarative filtering rules, the broker becomes an active coordination point capable of validating payloads, enriching metadata, and selectively routing messages based on semantic or contextual criteria. This approach increases architectural flexibility by allowing the system to adapt to evolving data flows and heterogeneous application requirements, while also providing tighter operational control through fine-grained policy enforcement directly within the communication layer. As a result, NSCA can support diverse edge–cloud scenarios — from safety-critical event processing to large-scale data aggregation — without requiring changes to client applications or service logic.

To enhance flexibility and control over message handling within SCA, we introduced broker-level extensibility through custom plugins.[1] In the current implementation, an MQTT broker plugin written in C for performance-critical environments is employed. The plugin leverages libyaml to load a declarative YAML configuration, enabling system integrators to specify message transformation, filtering, and routing rules directly at the broker layer. These rules operate over both topic structure semantics and payload content, aligning with NSCA's architectural model. A key advantage of this solution is its runtime configurability: plugin behavior can be modified without recompilation or downtime, allowing deployment environments to adapt dynamically to new service requirements. By executing inside the broker's main event loop, the plugin introduces sub-millisecond processing latency, making it suitable for real-time and safety-critical smart city scenarios. Typical transformations supported by the plugin include:

- *Payload validation and sanitization* (schema compliance, field checking),
- *Metadata augmentation* (timestamp insertion, coordinate tagging, source labeling),
- *Conditional routing* based on content attributes (e.g., on the basis of asset type or geographic tags),
- *Protocol translation and republishing* to internal service topics.

For instance, in mobility use cases, a mobile application may publish GPS telemetry to a generic topic. The plugin can validate message fields, enrich the telemetry with standardized timestamps, classify the spatial region, and reroute the data to a service-specific topic (e.g., VRU risk evaluation) or trigger alert generation if malformed or anomalous patterns are detected.

A central requirement for scalability in large scale smart city systems is efficient filtering, especially in dense urban sensing environments where thousands of entities generate continuous data streams. Broker-side filtering ensures that only relevant events are forwarded to subscribed services, significantly reducing network load and processing overhead. This is particularly effective for spatially constrained services, which only require events within defined geographic regions or risk zones. The combination of hierarchical topic organization and plugin-driven filtering logic enables selective dissemination close to the data source, reducing latency and preventing unnecessary message propagation. In the current prototype, spatial filtering is not implemented inside the broker plugin itself. Instead, spatial selection is performed at the service layer by the GeoPerception component — and subsequently reused by the VRU pipeline — through geospatial queries issued to a dedicated in-memory cache of recent object locations. GeoPerception maintains a short sliding window of the latest positions for each tracked entity, indexed by infrastructure-level identifiers, and updates this cache continuously as new detections arrive. Upon each user telemetry update, the service retrieves only the objects located within a configurable radius around the requester by leveraging Redis geospatial indexing primitives (e.g., GEOADD for position updates and GEOSEARCH or radius queries for proximity selection). This design intentionally keeps the broker plugin stateless, avoiding shared mutable state within the broker execution context, while delegating stateful operations to an external component that supports safe concurrent access through atomic commands. As clarified in the revised manuscript, broker plugins are primarily used for lightweight transformations and routing, whereas spatial filtering follows a stateless worker pattern implemented at the service layer and backed by an external geospatial cache.

This modular and extensible approach empowers NSCA to incorporate cooperative edge intelligence without modifying client applications or services. It also creates a foundation for future extensions, such as edge-level anomaly detection, adaptive prioritization, local caching, and context-aware access control. By shifting part of the decision logic to the broker layer, NSCA enhances system responsiveness and scalability while maintaining interoperability and deployment agility.

Nous Smart City Architecture (NSCA) - Cloud

The transition toward next-generation smart city infrastructures increasingly depends on the ability of multiple organizations—public administrations, mobility operators, utility providers, private companies, and technology vendors—to exchange data in a trusted, controlled, secure, and transparent manner. Traditional cloud architectures, typically built around centralized data lakes or tightly coupled integration pipelines, no longer suffice in ecosystems where data sources are heterogeneous, distributed, and bound by strict governance constraints. To address these limitations, the European Union is promoting a new paradigm for cross-organizational collaboration: **Data Spaces**.[2,3]

A **Data Space** is a federated, interoperable, and governance-driven ecosystem in which independent stakeholders can share data without relinquishing ownership or control. Instead of aggregating data in a single repository, each participant maintains sovereignty over its assets while exposing them through standardized interfaces, usage policies, and contractual frameworks. This approach aligns with the principles of transparency, trust, interoperability, and privacy-by-design that underpin the broader European strategy for digital sovereignty. In particular, Data Spaces are envisioned as foundational components for EU-wide initiatives such as the European Mobility Data Space, the Smart Communities Data Space, and cross-domain environments that connect mobility, energy, logistics, tourism, and public administration services.

Within this strategic landscape, **SIMPL/Open (SIMPL)**—the *Smart Middleware for Trusted Data Spaces*—plays a crucial enabling role.[4] SIMPL is an open, modular, and standards-aligned middleware architecture developed by the European Commission to simplify the creation, deployment, and operation of trustworthy data spaces. It provides a reference set of building blocks that implement the technical foundations required for data governance, interoperability, identity management, and secure data exchange. Rather than prescribing a single monolithic platform, SIMPL promotes a highly composable architecture: each organization can adopt only the components it needs, while maintaining compatibility with other participants through common APIs, vocabularies, policy frameworks, and security services.

At its core, SIMPL supports three complementary goals:

1. *Data sovereignty and trust enforcement.* SIMPL incorporates mechanisms for declarative data usage policies, identity and credential management, auditing, and contract negotiation, ensuring that data providers can specify how their assets may be accessed, processed, or redistributed. These policies follow European values of privacy, security, and accountability, making SIMPL a natural fit for public-sector and multi-stakeholder scenarios typical of smart cities.

2. *Interoperability and semantic consistency.* The framework defines common data models, metadata vocabularies, and semantic reference structures that allow heterogeneous systems to communicate using shared meaning. This is essential in domains like mobility or urban management, where datasets originate from traffic sensors, IoT devices, administrative registries, digital twins, or third-party platforms.

3. *Federation rather than centralization.* SIMPL enables multiple autonomous domains to interconnect through a federated data infrastructure, in which each participant runs its own connectors, catalogues, and policy engines. This model supports scalable collaboration among cities, agencies, and private entities without requiring them to surrender data or depend on a single platform operator.

Integrating SIMPL/Open into the Nous Smart City Architecture (NSCA)

In the **NSCA Cloud layer**, the SIMPL/Open architecture serves as the foundation upon which the NOUS ecosystem builds its Data Space–compliant services. While the NSCA Edge layer handles real-time telemetry, perception, and low-latency coordination through MQTT-based publish/subscribe mechanisms, the NSCA Cloud layer focuses on multi-stakeholder data exchange, long-term storage, policy-driven interoperability, and cross-domain analytics. SIMPL/Open directly supports this by providing:

- *EDC-based data exchange pipelines*, enabling secure contract negotiation, data transfer, and controlled access to event-driven urban data streams.

- *Federated catalogs and metadata registries* that make data assets discoverable across multiple organizations while preserving semantic uniformity.

- *Common services for identity, authorization, and certification,* ensuring consistent trustworthiness and verifiable compliance across the federation.

- *Policy enforcement and auditing mechanisms* that regulate usage conditions, enabling providers to expose only approved subsets of their telemetry, events, or historical datasets.

- *A modular governance layer* that supports scalable involvement of city departments, utility operators, mobility companies, research partners, and private stakeholders.

In this model, the NOUS Data Space acts as the cloud-side extension of NSCA, enabling cities not only to share real-time data but also to create value-added services, cross-city analytics, and interoperable digital twins aligned with EU-wide Data Space initiatives. For example, mobility services running in MASA can be exposed—under clear contractual terms—to other municipalities, industrial partners, or national platforms; environmental data can be combined with traffic patterns; and multi-city intelligence can be developed without violating data protection constraints or duplicating storage. Whereas NSCA manages low-latency interactions close to data sources through event-driven communication, the Data Space layer enables continuous ingestion of selected data streams and provides near–real-time access through standardized, policy-controlled interfaces. This separation allows operational processes to remain responsive at the edge while supporting secure multi-stakeholder data sharing and long-term analytics in the cloud. Building on this foundation, value-added services can be developed by combining live and historical datasets with analytics and AI pipelines, transforming edge observations into cross-domain insights, interoperable digital twins, and decision-support applications aligned with Data Space governance principles.

For urban environments, Data Spaces and SIMPL/Open unlock several strategic capabilities:

- *Multi-stakeholder cooperation by design.* Cities can coordinate with transport operators, emergency responders, energy providers, and technology partners using unified, policy-governed data channels.

- *Cross-city federation.* Independent NSCA deployments across different municipalities can interoperate, aligning with EU digital policies.

- *Data sovereignty and GDPR compliance.* Sensitive data—including mobility traces, sensor telemetry, and video-derived metadata—remains under the control of its originator.

- *Scalable, interoperable digital twins.* Federated data exchange enables the creation of shared, cross-domain digital twins for mobility, environment, logistics, and public safety.

- *Reduced vendor lock-in.* SIMPL's open and modular architecture enables plug-and-play adoption of connectors, catalogs, and governance components compatible with the European Data Space ecosystem.

By integrating SIMPL/Open as the backbone of its Cloud layer, NSCA positions itself not only as an efficient smart city middleware but also as a **future-proof, Data Space–aligned architecture**, capable of supporting advanced urban services across cities, regions, and national infrastructures. Building on these principles, the NSCA Cloud layer adopts SIMPL/Open as the foundation for implementing trusted, sovereign, and interoperable data exchange among smart-city stakeholders. While the introductory discussion highlights the strategic relevance of Data Spaces and the role of SIMPL in enabling secure federation, the following sections describe how these concepts are concretely instantiated within the NOUS architecture. In particular, NSCA integrates SIMPL/Open through a set of operational components that manage event-driven data, publish it as standardized data assets, negotiate access policies, and coordinate the transfer of information across organizational boundaries.

To make these mechanisms actionable in real deployments, NSCA structures its cloud-side workflow around two complementary perspectives: the **data provider**, responsible for exposing event-based data streams to the Data Space, and the **data consumer**, responsible for discovering, accessing, and integrating shared data assets. This separation reflects the federated nature of Data Spaces, where each participant maintains control over its resources while interoperating through common catalogs, connectors, and policy enforcement services (see Figure 2).

The next subsections describe these two roles in detail. They illustrate how raw events produced at the edge are synchronized, filtered, registered, and ultimately shared through the SIMPL/Open-compliant infrastructure; and conversely, how authorized consumers discover, contract, and receive event-based assets through secure and policy-controlled pipelines. Together, these workflows show how NSCA operationalizes Data Space principles within a scalable cloud architecture that supports cross-city collaboration, multi-stakeholder services, and long-term integration of smart-city intelligence.

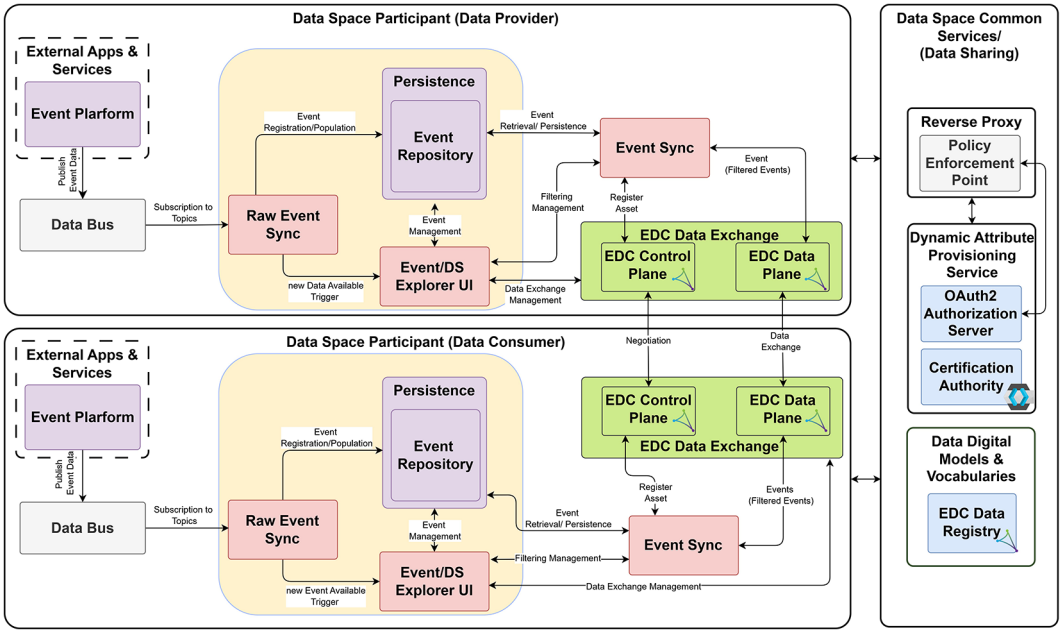


Figure 2. NCSA Data space architecture. The figure illustrates the high-level components that enable governance-aware data sharing across the NSCA cloud layer. Event-based data generated at the edge are ingested through the Event Platform and synchronized by the Event Sync component toward the Persistence layer, where datasets are stored and prepared for publication. The Event/DS Explorer provides a unified interface to configure filtering rules, manage metadata, and expose selected assets to the Data Space ecosystem. Secure exchange is handled through EDC Control and Data Planes, which support asset discovery, contract negotiation, and policy-compliant transfer between providers and consumers. On the right side, Data Space Common Services — including identity management, certification, and policy enforcement — ensure trusted interoperability, semantic consistency, and governance across federated participants.

Data provider perspective

On the provider side, an Event Platform emits new event data into the Data Bus – which implements a topic-based publish/subscribe mechanism. The Raw Event Sync component handles these events and triggers the synchronization operations required for updating the Event Repository in the Persistence layer, ensuring long-term storage of the event-based data. The Event/DS Explorer UI is a key component in the architecture that bridges the gap between the event data management and data space operations. It provides a user-friendly web interface that allows users to explore, configure, and manage the event-based data and their data sharing configurations. The Explorer acts as the main point of interaction with the Event Sync component, offering visibility into which data events are available, how they are structured and how they are exposed to the EDC Data Exchange component. The main function of the Event/DS Explorer UI is to enable filtering before event-based data assets are published to the data space. Users can select which subset of the events should be included in the shared data plane, ensuring only relevant and approved information is made accessible to external parties. The Event Sync module registers the event-based data assets within the EDC Control Plane, where the corresponding metadata are published and made discoverable to data space participants. The EDC Data Plane then supports the secure and policy-compliant data transfer, delivering the filtered event-based data to authorized consumers.

The Event Sync component enables the publication of data assets following the standardized workflow defined by the Data Space protocol. When an existing dataset is stored in the Event Repository, the data provider uses the Event Sync module to define and attach descriptive metadata that capture the dataset's structure, semantics, and intended usage context. These metadata ensure that assets become discoverable, interpretable, and interoperable within the Data Space ecosystem. Once prepared, the dataset is exposed through a certified connector, where usage policies, access conditions, and contractual agreements are specified and enforced prior to data exchange. This process guarantees that data publication remains aligned with governance requirements while preserving data sovereignty and controlled access across participating organizations.

The Raw Event Sync component acts as the bridge between MQTT-based event streams generated within NSCA and the SIMPL/Open Data Space infrastructure. It subscribes to selected topics on the cloud broker and persistently stores incoming events in the Event Repository, progressively building a historical dataset for each monitored stream. These

accumulated datasets can then be processed through the Event Sync component, which prepares the corresponding metadata, applies governance rules, and registers them as data assets within the EDC Control Plane. Once published, authorized consumers can access the data through the EDC Data Plane, typically receiving historical batches that comply with the defined sharing policies and contractual conditions. This workflow ensures continuity between real-time event ingestion and governed data publication, enabling NSCA to transform live MQTT streams into interoperable Data Space assets without disrupting edge-level operations.

MQTT brokers preserve per-topic message ordering and provide reliable delivery through configurable Quality of Service (QoS) levels, ensuring that event streams maintain temporal consistency as they traverse the NSCA messaging fabric. The Raw Event Sync component persistently stores incoming messages in the Event Repository as an ordered log immediately upon reception, providing durable storage and preventing data loss even under transient failures. The Event Sync component subsequently reads from this repository rather than directly from live streams, thereby maintaining ordering guarantees while decoupling ingestion from Data Space publication workflows. During distribution, delivery through the EDC Data Plane follows an at-least-once transfer model, combining persistent buffering and contract-driven data exchange to ensure reliable end-to-end propagation without compromising data integrity.

Data consumer perspective

On the consumer side, a similar mechanism ensures seamless synchronization and consumption of event-based data assets. Through the Event/DS Explorer UI, consumers can discover available offerings, initiate data transfer workflows, and leverage the Event Sync component to receive and store event-based data assets to the Persistence layer based on specific configurations. The EDC Control Plane facilitates the discovery and negotiation of event-based data assets, while the EDC Data Plane executes the secure and policy-compliant transfer of data in accordance with the negotiated contracts and sharing policies established during the data exchange process.

Data space common services

Interactions between Data Space participants are secured through the Data Space Common Services layer, which enforces standardized identity management, authentication, authorization, and usage-control policies to ensure that data exchanges occur only between trusted entities and strictly according to agreed governance rules. This security model guarantees traceability, controlled access, and compliance with Data Space principles while supporting federated collaboration across organizations. From an implementation perspective, these mechanisms rely on widely adopted industrial standards — including OAuth2.0 for authorization, and SSO frameworks based on SAML and LDAP for identity federation — enabling interoperability with existing enterprise infrastructures and ensuring a consistent trust framework across the NSCA cloud ecosystem. These components ensure that all data transactions comply with established identity, authentication, and trust policies among trusted participants. Furthermore, the EDC Data Registry, as part of the Data Digital Models and Vocabularies, maintains standardized metadata, digital asset descriptions, and interoperability vocabularies, thereby ensuring semantic consistency and discoverability of the exchanged information across the data space.

Experimental evaluation

To demonstrate the expressiveness, flexibility, and generality of the Nous Smart City Architecture (NSCA), we present two representative services that showcase different interaction models, communication patterns, and deployment scenarios within the MASA testbed. These services -- differing in interaction models (event-driven vs. aggregative), communication patterns (low-latency push vs. hybrid streaming/query), and deployment strategies (edge-centric vs. edge–cloud) -- collectively illustrate how SCA natively supports heterogeneous roles (assets, services, observers), semantic topic coordination, and distributed message flows across the edge–cloud continuum.

Vulnerable Road User (VRU) Service (Safety-Critical Asset Service)

The VRU Service is a real-time safety application designed to mitigate collision risks for pedestrians, cyclists, and other vulnerable road users (VRUs). It delivers personalized, low-latency safety alerts by fusing telemetry data from heterogeneous assets distributed across the environment. Interaction with the service is based on a registration model in which each participating asset subscribes to safety notifications relevant to its context.

Smart cameras and roadside units continuously publish object detections, including bounding boxes, object classes, and tracking identifiers, while mobile applications provide periodic GPS telemetry from user devices. For each registered asset, the VRU Service correlates visual perception from cameras with GNSS data from mobile devices to identify and track nearby entities in real time. Based on this correlated information, it evaluates proximity risk by considering relative distance, speed, and predicted trajectory, assigning each situation a severity level such as low, medium, or high. The VRU service evaluates proximity risk by combining relative distance, speed, and predicted trajectories of interacting entities to

estimate potential conflicts and assign a corresponding severity level. The manuscript clarifies this algorithmic approach while emphasizing that its parameters are fully customizable, allowing adaptation to different operational contexts, deployment scenarios, and safety requirements without affecting the underlying NSCA architecture. When a potential hazard is detected, the service generates a targeted alert and publishes it to dedicated notification channels so that only the relevant users and systems receive it. By enabling mission-critical, spatially aware reasoning across distributed and heterogeneous components, the VRU Service exemplifies the capabilities of the SCA architecture in real-world safety scenarios.

Beyond real-time alerting, the VRU Service also produces a continuous stream of structured, privacy-preserving metadata—such as aggregated detection events, risk-level distributions, near-miss indicators, and spatio-temporal mobility patterns—that can be exported to the NSCA Cloud layer and exposed through the Data Space. Once synchronized through the Event/DS Explorer and registered as data assets in the SIMPL/Open-compliant infrastructure, these datasets become securely discoverable and shareable with authorized third parties, including other municipalities, mobility agencies, road-safety authorities, and insurance operators. Because data sovereignty and usage policies are enforced at the connector level, providers can expose only high-level or anonymized aggregates while retaining full control over how the information is accessed, reused, or combined with external datasets. This enables cross-city benchmarking of safety hotspots, integration with regional or national mobility platforms, and the development of advanced risk-modelling services, all while ensuring compliance with GDPR and Data Space governance rules.

GeoPerception service (Edge Cooperative Awareness)

The GeoPerception Service enables localized cooperative perception by aggregating and correlating environmental detections from nearby edge nodes around a mobile user to enhance situational awareness. In this sense, it can be interpreted as the first stage of the VRU application pipeline, providing contextual awareness that can later be leveraged by higher-level safety services. Unlike the alert-oriented VRU Service, its goal is not to infer risk but to reconstruct contextual understanding of the surrounding environment. It receives object detections from edge-deployed cameras, which continuously publish semantic labels, and GPS telemetry from mobile applications that may also specify query preferences, such as requesting only certain types of objects like vehicles. For every telemetry update received from a mobile asset, the service dynamically computes a spatial vicinity of 50 meters around the requester and filters incoming detections based on relevance, combining criteria such as distance, object class, and potential visibility constraints. The resulting localized perception view is then sent directly to the requesting asset through a dedicated notification channel. In doing so, the GeoPerception Service demonstrates how SCA supports dynamic edge-powered service composition, fine-grained spatial filtering, and low-latency analytics, all while significantly reducing bandwidth usage by transmitting only context-relevant information.

In addition to providing real-time, localized situational awareness to individual users, the GeoPerception Service produces a rich stream of structured environmental metadata—such as aggregated object counts, traffic density indicators, class distributions, and spatial perception snapshots—that can be synchronized to the cloud and exposed as Data Space assets. Once processed through the Event/DS Explorer and registered via SIMPL/Open connectors, these perception-based datasets become discoverable and consumable by authorized third parties, including mobility planners, transportation companies, digital-twin operators, and neighboring municipalities. Because the Data Space enforces strict usage policies and supports granular filtering, providers can share only anonymized, aggregated, or domain-specific subsets of the data, ensuring full compliance with GDPR and local governance requirements. This enables advanced applications such as inter-city mobility analytics, infrastructure planning, crowd-flow monitoring, and enhanced digital twins that combine perception data across multiple regions—ultimately fostering cross-city collaboration and more informed decision-making.

In the current implementation, a purely topic-based subscription model does not provide a viable baseline for the spatial selection required by GeoPerception. Relying exclusively on hierarchical topic filtering would not scale in dense sensing environments and would complicate access control and safe exposure of fine-grained sensor streams across tenants. Instead, GeoPerception adopts a query-oriented interaction pattern in which the service dynamically selects only the objects located in the vicinity of a requesting user while simultaneously normalizing heterogeneous payloads originating from different infrastructure sources (e.g., cameras and RSUs) into a unified data representation. This additional processing step introduces a modest overhead compared to delivering pre-filtered local data; in our deployment, we observed an additional latency of approximately 10–20 ms per request. The exact impact depends on factors such as detection density, request frequency, and available worker capacity, reflecting a deliberate trade-off between minimal latency and improved scalability, controlled data exposure, and standardized downstream consumption within the NSCA architecture.

Experiments

To assess the latency and responsiveness of the NSCA communication infrastructure, we evaluated the end-to-end (E2E) delivery time of messages across the architecture. We first examined baseline communication latency under three representative deployment configurations, independent of any service logic, to isolate the overhead of NSCA's messaging fabric. We then analyzed data from two operational services deployed in MASA—VRU and GeoPerception—to evaluate NSCA under real-world producer–consumer interaction patterns. These measurements quantify the communication delays introduced by broker traversal, inter-broker forwarding, and optional message transformation, excluding any internal processing performed by services themselves. While VRU and GeoPerception are representative of safety-critical and perception services, the SCA architecture supports a broader set of urban applications with similar requirements.

The following architectural assumptions guide our evaluation:

1. Services are deployed at the Edge to model realistic latency-sensitive deployments.

2. Assets may be deployed either at the Edge or in the Cloud, depending on connectivity and mobility constraints.

3. Routing adapts dynamically to the location of message producers and consumers, using inter-broker forwarding only when necessary.

We evaluated E2E communication latency across three deployment pathways:

- *Cloud-to-Edge (CE):* a cloud-based asset publishes data to a cloud broker, which forwards it through the inter-broker bridge to an edge broker where the consuming service is located.

- *Edge-to-Cloud (EC):* an edge asset communicates with a cloud-based consumer through the edge broker and inter-broker bridge.

- *Edge-to-Edge (EE):* both producer and consumer operate within the same local edge domain, using only the edge broker.

These three paths correspond to the routing flows illustrated in Figure 1 (left). In the CE and EC cases, messages must traverse both brokers through the inter-broker bridge, adding network delay and security-layer overhead before reaching the destination. In contrast, EE communication remains confined to the local edge broker, resulting in the lowest and most predictable latency. This distinction highlights how NSCA's hybrid broker deployment enables flexible routing: local interactions are kept within the edge domain for real-time processing, while supervision, cloud-level consumers, and cross-domain services can still access the necessary data through controlled inter-broker forwarding.

Figure 3 decomposes these latencies into three propagation segments: (i) source asset → source broker, (ii) inter-broker forwarding, and (iii) destination broker → destination asset. To obtain these measurements, we deployed lightweight publisher and subscriber clients that exchanged timestamped messages through the NSCA communication fabric under controlled configurations. Each experiment consisted of repeated telemetry transmissions at a fixed rate, where timestamps were embedded at the source asset and collected at the destination to compute end-to-end delivery time. Services were intentionally kept minimal to isolate communication overhead from application processing, and measurements were repeated across multiple runs to mitigate transient network fluctuations.

As expected, the Edge-to-Edge (EE) path delivers the lowest latency due to local routing within the same edge broker, avoiding inter-broker traversal and WAN propagation; this makes it particularly suitable for time-critical applications such as VRU safety alerts. In both Cloud-to-Edge (CE) and Edge-to-Cloud (EC) scenarios, additional delay arises from wide-area network transmission, TLS/security handling at cloud ingress, and broker bridging operations. The EC path is slightly faster than CE because messages originating at the edge traverse outbound network policies more efficiently, whereas incoming cloud traffic is subject to stricter ingress filtering and NAT/firewall inspection at the edge perimeter. These results highlight how NSCA's hybrid broker architecture dynamically adapts routing paths while preserving predictable latency characteristics across heterogeneous deployment configurations.

In a second experiment, we evaluate the impact of broker-level message and object filtering on communication efficiency. The GeoPerception service applies spatial filtering at the edge by selecting only detections that fall within a configurable

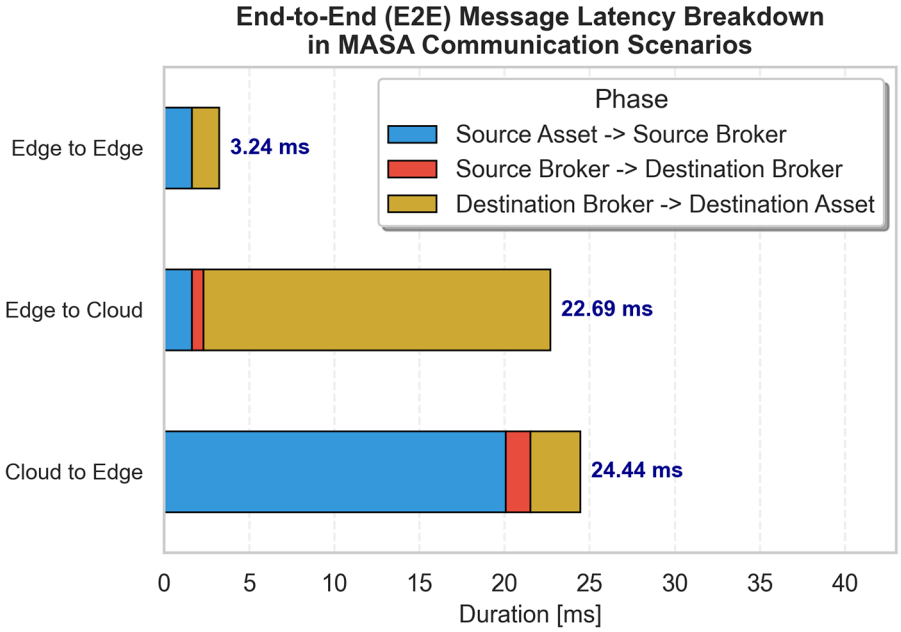


Figure 3. The figure presents the decomposition of message delivery latency into three propagation segments — source asset to source broker, inter-broker forwarding, and destination broker to destination asset — under three representative deployment paths: Edge-to-Edge (EE), Edge-to-Cloud (EC), and Cloud-to-Edge (CE). The comparison highlights how local edge routing minimizes delay by avoiding inter-broker traversal, while cloud-involved paths introduce additional overhead due to WAN transmission, security processing, and broker bridging. The visualization emphasizes the role of the hybrid broker architecture in dynamically adapting routing depending on the location of producers and consumers, showing how NSCA preserves low-latency communication for safety-critical services while still enabling cloud-level supervision and data sharing. The breakdown also illustrates the relative impact of network ingress policies and asymmetric routing effects observed in real deployments.

vicinity around a requesting mobile asset, thereby reducing both bandwidth consumption and downstream processing load. To quantify this effect, we deployed a publisher that continuously generated perception events from edge cameras and a subscriber client that received the resulting data stream through the NSCA broker infrastructure. Two configurations were executed under identical conditions: one with filtering disabled, where all detections were forwarded, and one with filtering enabled at the broker level. During each run, the subscriber recorded the cumulative payload volume and the number of detected objects received over time. As shown in Figure 4, we fitted a linear regression to both payload size and object count curves to estimate the effective growth rate of transmitted data, allowing a direct comparison of slopes between filtered and unfiltered scenarios and providing a clear measure of bandwidth and processing savings achieved through edge-side filtering.

- Payload volume increases with a slope of 12.7 KB/s without filtering and with 2.7 KB/s with filtering.
- The number of detected objects delivered to the consumer increases with a slope of 37.7 objects/s without filtering and with 4.9 objects/s with filtering.

The slightly lower byte reduction compared to object reduction is due to message headers and metadata that must be preserved, even when no objects are present. These results show that moving filtering and aggregation to the Edge significantly reduces network and processing overhead, preventing unnecessary traffic from entering the Cloud or burdening downstream applications. This confirms the effectiveness of SCA's broker-level filtering strategy for scalable perception services.

Finally, to evaluate interaction responsiveness within NSCA, we analyzed the complete message exchange timeline of the VRU service, measuring the latency of each communication phase from initial broker connection to service unsubscription. The experiment was conducted using a controlled client application that executed the full interaction workflow through the NSCA messaging layer while timestamps were recorded at each step to isolate communication delays from internal service processing. The interaction sequence included: (i) connection to the broker, (ii) service discovery through

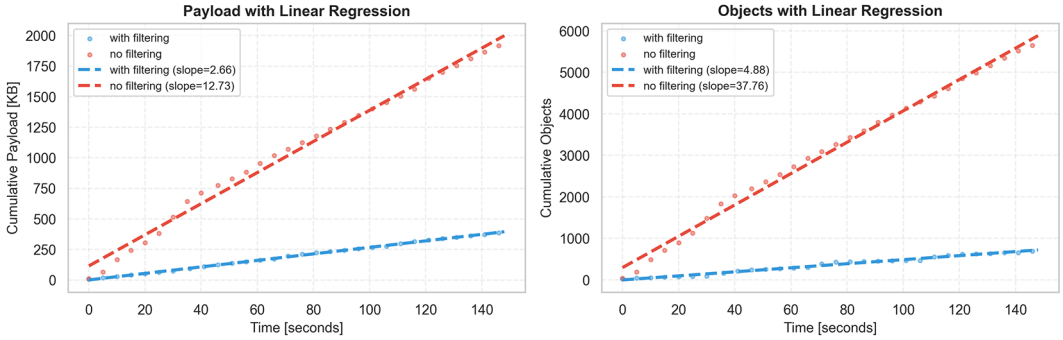


Figure 4. The figure compares the cumulative data received by a consumer with and without spatial filtering performed at the edge. Blue curves represent the filtered output delivered to downstream clients, while green curves correspond to the unfiltered data stream processed by the GeoPerception service. The left plot shows the evolution of total payload volume (KB) over time, whereas the right plot reports the cumulative number of detected objects transmitted. Linear regression lines highlight the different growth rates, illustrating how broker-level filtering significantly reduces the number of propagated detections and limits bandwidth consumption while preserving relevant contextual information. The results demonstrate how semantic topic routing and edge-side aggregation in NSCA minimize unnecessary data dissemination, improving scalability for perception-driven smart city applications and reducing the load on cloud ingestion pipelines and remote observers.

retained topic metadata, (iii) service registration via dedicated registration channels, (iv) the first telemetry request–response cycle, (v) a second telemetry exchange performed under an already established session, and (vi) service unsubscription. The first telemetry response incorporates initialization overhead related to connection setup, topic subscription, and service binding, whereas subsequent interactions benefit from session reuse and stabilized routing paths, resulting in lower and more consistent latency.

Experiments were executed across four deployment configurations to assess the impact of network placement and connectivity technology: edge-based assets communicating over Ethernet, edge deployments over 5G NSA MEC connectivity, and cloud-based assets interacting through the inter-broker bridge. Each configuration was repeated multiple times to account for transient network variability, allowing a comparative analysis of interaction phases under heterogeneous edge–cloud conditions. Results (Figure 5) show that:

- Ethernet at the Edge delivers the best performance, achieving minimal and stable latency.
- 5G NSA via MEC introduces an extra 10–15 ms per interaction phase, but remains suitable for mobile VRU deployments thanks to consistent latency.
- Cloud-based assets show the highest and most unstable latency, due to WAN traversal, inter-broker routing, and public internet congestion.
- Unsubscription is consistently lightweight, demonstrating the efficiency of topic-based interaction coordination.

These findings emphasize that edge deployment is essential for time-critical safety services, while 5G MEC offers a practical compromise between mobility and latency. Cloud-only deployments are functional but unsuitable for strict real-time guarantees.

Although these experiments primarily assess the performance of the NSCA messaging layer for real-time edge services, their implications extend directly to the Data Space integration in the cloud. Efficient edge-to-cloud routing, predictable latency, and aggressive broker-level filtering directly influence how event-based datasets are synchronized, catalogued, and exposed as data assets in the SIMPL/Open ecosystem. Lower propagation delays ensure that aggregated mobility and perception data reach the cloud layer in near–real time, enabling timely updates of shared assets, digital twins, and analytics pipelines. Similarly, reducing unnecessary payloads at the edge minimizes cloud ingestion overhead, improves the scalability of the Event Sync component, and ensures that only meaningful, policy-compliant information enters the Data Space. As a result, the performance characteristics demonstrated in these experiments are foundational not only for

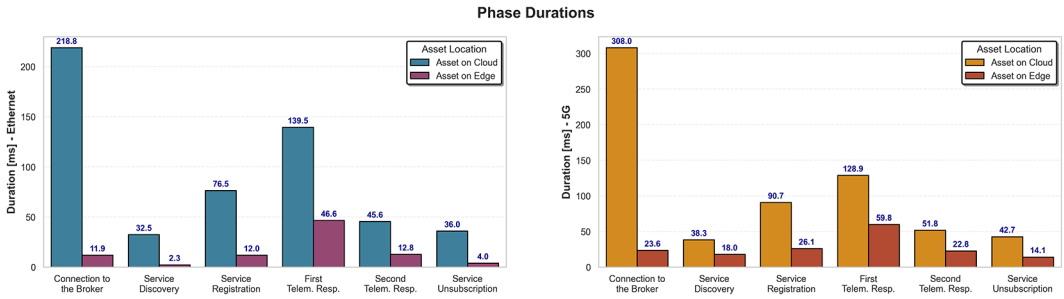


Figure 5. The figure illustrates the sequence of interaction phases between assets and services — including broker connection, service discovery, registration, telemetry exchange, and unsubscription — measured across edge and cloud deployments using Ethernet and 5G NSA connectivity. Each timeline highlights the temporal contribution of individual communication steps, showing how initialization introduces a slightly higher delay during the first telemetry cycle, while subsequent interactions benefit from session reuse and stabilized routing paths. Edge deployments over Ethernet achieve the lowest and most predictable latency, whereas 5G MEC connectivity introduces a consistent overhead while maintaining responsiveness suitable for mobile safety applications. Cloud-based interactions exhibit higher variability due to WAN traversal and inter-broker forwarding. Overall, the visualization emphasizes how NSCA's event-driven architecture maintains efficient interaction workflows across heterogeneous network conditions while supporting real-time safety services. Left: Asset located at the Edge with Ethernet. Right: Same deployment over 5G NSA, introducing consistent latency overhead.

real-time applications like VRU and GeoPerception, but also for the reliability, cost-efficiency, and responsiveness of multi-stakeholder data exchange within the NOUS Data Space.

Related work

Recent advances in smart city infrastructures have underscored the need for scalable and adaptive communication architectures capable of handling heterogeneous data streams, latency-sensitive services, and distributed intelligence across the IoT–Edge–Cloud continuum. Conventional centralized platforms—typically built around large-scale data warehouses or distributed stream-processing frameworks such as Apache Kafka and Apache Flink—provide strong support for high-throughput analytics.[5,6] However, their rigid topologies, tightly coupled data pipelines, and complex deployment lifecycles make them less suitable for dynamic, event-driven environments where responsiveness and modularity at the network edge are essential.

To address these architectural constraints, the Data Mesh paradigm has emerged as a decentralized alternative, advocating domain-oriented data ownership, self-serve data infrastructure, and federated governance to promote interoperability at scale.[7] While promising, Data Mesh approaches often overlook the real-time communication and synchronization requirements typical of cyber-physical urban systems.

This paper builds upon our previous research in edge computing, cooperative safety services for Vulnerable Road Users (VRUs), and distributed service orchestration, introducing a Smart City Architecture (SCA) that combines broker-level extensibility, topic-driven interaction semantics, and publish/subscribe service coordination. The design is inspired by earlier contributions that explored 5G MEC-enabled low-latency edge-to-cloud communication and adaptive orchestration mechanisms for distributed services. In particular, VRU collision risk mitigation using MEC infrastructures was evaluated in 8, while broker-side computation and semantic message routing were enabled through a plugin-based MQTT extension introduced in 1. Complementary work demonstrated urban sensor fusion for VRU risk prediction and investigated dynamic orchestration across heterogeneous computing tiers.[9–13] Additionally, real-world deployments of smart city platforms and data mesh-inspired designs in urban contexts were reported in 1, 14. The SCA framework consolidates these research directions by providing a unified, MQTT-native middleware with modular client registration and multi-role entity interaction.

Within the mobility and safety domain, related efforts include V2I/V2P services for proactive VRU protection,[15,16] machine learning-based pedestrian intent prediction,[17] and augmented reality interfaces for cooperative awareness.[18] Further advancements in ADAS systems leveraging 5G and edge computing were discussed in 19, 20. SCA extends this line of work by enabling fine-grained messaging at the client level instead of relying on generic broadcast-based patterns, improving privacy, scalability, and selective information dissemination through topic-based registration and edge-based message routing.

With regard to MQTT-based distributed systems, several contributions have explored federated broker overlays[21,22] and hierarchical routing strategies for publish/subscribe networks,[23,24] though often at the expense of architectural simplicity or maintainability. Alternative efforts have proposed MQTT gateways with message filtering and AI-based traffic prioritization,[25,26] but these approaches typically introduce additional operational layers and deployment overhead. In contrast, SCA adopts a broker-native approach that preserves the simplicity of MQTT while enabling runtime extensibility via plugins, supporting YAML-configurable message transformations, semantic routing rules, and minimal processing latency across distributed edge deployments.

Recent MEC offloading research increasingly targets uncertainty-aware and robust optimization, especially in multi-server and vehicular edge settings where both wireless conditions and compute availability fluctuate rapidly.[27,28] Alongside robust formulations, learning-based approaches — often based on deep reinforcement learning — and digital twin–assisted pipelines are emerging to capture spatio-temporal dynamics and support adaptive offloading and caching decisions in highly mobile environments.[29,30] These works highlight a shift from static resource orchestration toward adaptive, risk-sensitive edge intelligence, complementing the communication-centric perspective adopted by NSCA. In this context, our architecture focuses on providing a flexible event-driven middleware capable of supporting such adaptive strategies through broker-level extensibility and semantic routing, rather than proposing a specific offloading optimization framework.

Foundational contributions on mobile edge computing and vehicular task offloading established the architectural basis for low-latency distributed intelligence, introducing adaptive learning mechanisms and communication-aware resource management strategies.[31–33] While these earlier works primarily addressed performance optimization and computation placement, more recent studies emphasize uncertainty modeling, dynamic service placement, and robustness in safety-critical mobility scenarios.[27–30] This progression motivates the design of NSCA as an infrastructure capable of supporting evolving edge intelligence paradigms within real-world smart city deployments.

Recent research on European Data Spaces and federated data-sharing infrastructures has increasingly emphasized the need for trusted, interoperable, and sovereignty-preserving mechanisms for cross-organizational data exchange. The International Data Spaces (IDS) architecture has been one of the earliest and most influential frameworks in this area, introducing certified connectors, usage-control policies, and standardized contract negotiation to ensure secure, sovereign data sharing among independent organizations.[34] Building on IDS principles, the GAIA-X initiative proposes a federated cloud and data ecosystem where participants interconnect through common services for identity, trust, and catalog federation, with strong alignment to European regulations on privacy, security, and transparency.[35] These foundations have directly informed the design of sector-specific European Data Spaces, such as the European Mobility Data Space (EMDS) and the Smart Communities Data Space, which promote cross-city interoperability and semantically harmonized data models across public administrations and mobility service providers.

Recent contributions on data governance and European Data Spaces further emphasize the need to integrate real-time edge communication with sovereign, policy-driven data sharing frameworks.[36] In smart city ecosystems, governance models increasingly address interoperability requirements, cross-departmental collaboration, and standardized data access mechanisms aligned with European regulatory principles. These studies highlight that technical edge-cloud solutions alone are insufficient without governance-aware architectures capable of enforcing usage policies, trust, and organizational interoperability. Within this perspective, NSCA complements existing edge-cloud frameworks by introducing a governance-oriented cloud layer based on Data Space principles, bridging real-time edge intelligence with regulated multi-stakeholder data exchange.

Within this broader ecosystem, the European Commission recently introduced SIMPL/Open (Smart Middleware for Trusted Data Spaces) as a unifying middleware architecture that operationalizes Data Space concepts with modular, open-source building blocks for connectors, policy enforcement, semantic registries, and identity services.[4,37] SIMPL/Open aims to reduce fragmentation by providing a reference implementation aligned with the upcoming Data Act, Data Governance Act, and High-Value Data frameworks, ensuring that public administrations and private stakeholders can deploy interoperable Data Space solutions with minimal integration overhead. Recent technical reports highlight the importance of modularity, extensibility, and common APIs in enabling cities, industrial clusters, and research institutions to participate in federated data ecosystems without losing control over their assets.

Further academic work has examined the challenges of semantic interoperability and cross-domain data governance in Data Spaces, with several studies exploring ontology alignment, metadata standardization, and policy-compliant data processing pipelines for multi-stakeholder environments.[2,38] These contributions complement SIMPL/Open by emphasizing the need for shared vocabularies, machine-readable governance rules, and automated compliance

checking—capabilities that are crucial for smart city deployments where datasets originate from heterogeneous cyber-physical systems. Together, these initiatives position Data Spaces as a core enabler for next-generation smart city platforms and provide the conceptual and operational building blocks that NOUS integrates within its cloud architecture.

Conclusion

This paper presented the Nous Smart City Architecture (NSCA), a modular and extensible middleware that unifies real-time edge intelligence with sovereign and interoperable data sharing in the cloud. Validated within the Modena Automotive Smart Area (MASA), NSCA demonstrates how lightweight publish/subscribe communication, semantic topic structuring, and federated brokers can support low-latency coordination among heterogeneous urban assets at the network edge. At the same time, its cloud-side integration with SIMPL/Open enables event synchronization, metadata curation, and policy-controlled publication of urban data assets within a Data Space ecosystem. Together, these two layers establish a seamless edge–cloud continuum, combining millisecond-level responsiveness for safety-critical services with governance-aware data interoperability across organizational boundaries.

Experimental results provide concrete evidence of the effectiveness of this design. The latency analysis showed that Edge-to-Edge routing consistently achieves the lowest and most stable end-to-end delivery times by avoiding inter-broker traversal, confirming the suitability of NSCA for time-critical applications such as VRU safety alerts. Conversely, Cloud-to-Edge and Edge-to-Cloud paths introduce predictable overhead due to WAN propagation and security processing, yet remain viable for supervisory and data-sharing workflows. The GeoPerception filtering experiment demonstrated that broker-level spatial filtering significantly reduces communication load, decreasing the growth rate of transmitted payload and detected objects while preserving relevant contextual information, thus improving scalability for perception-driven applications. Furthermore, the VRU interaction timeline analysis highlighted how session reuse stabilizes communication latency after initialization, with Ethernet-based edge deployments achieving the most consistent performance and 5G MEC deployments introducing only moderate additional delay, confirming the feasibility of mobile safety services within the NSCA framework.

Beyond performance metrics, the deployment of the VRU and GeoPerception services illustrated how NSCA accommodates heterogeneous service archetypes — ranging from event-driven safety pipelines to aggregative perception workflows — while maintaining architectural transparency for applications. These services demonstrate that real-time edge intelligence can coexist with governed cloud interoperability, enabling data products generated at the edge to be securely exposed within a Data Space and reused by municipalities, mobility operators, and third-party analytics platforms. The results collectively validate NSCA not only as a communication middleware but also as an enabling infrastructure for cooperative edge intelligence and multi-stakeholder urban data ecosystems.

Future work will expand NSCA along both architectural dimensions. On the edge side, we plan to enhance multi-protocol interoperability (AMQP, DDS, NGSI-LD), strengthen QoS guarantees, and extend plugin-based processing toward distributed edge analytics and anomaly detection. On the Data Space side, we aim to broaden semantic interoperability, automate usage-policy enforcement, and improve cross-city federation through standardized connectors and governance models aligned with emerging European initiatives for mobility and smart communities. By advancing both the edge middleware and the cloud Data Space infrastructure, NSCA aspires to become a comprehensive foundation for next-generation smart city platforms — one that supports real-time intelligence, secure data exchange, and scalable collaboration across diverse urban stakeholders.

Ethics and consent

Ethical approval and consent were not required.

Data availability

This paper describes software architecture, so there is no data associated with this publication.

Software availability

Source code available from: https://gitlab.eclipse.org/eclipse-research-labs/nous-project

Archived software available from: https://doi.org/10.5281/zenodo.17972799

License: Apache 2.0 license.

References

1. Pietri M, Taccini L, Bedogni L, *et al.*: **Efficient and flexibile IoT communication through a plugin-based MQTT processing architecture. In: International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT).** 2025.
Publisher Full Text
2. Otto B: **A federated infrastructure for European data spaces.** *Commun ACM.* 2022; **65**(4): 44–45.
Publisher Full Text
3. Scerri S, Tuikka T, de Vallejo IL, *et al.*: **Common European data spaces: challenges and opportunities.** *Data Spaces: Design, Deployment and Future Directions.* 2022: 337–357.
Publisher Full Text
4. European Commission: Smart middleware for trusted data spaces (simpl). 2024.
5. Machado IA, Costa C, Santos MY: **Data mesh: concepts and principles of a paradigm shift in data architectures.** *Procedia Comput Sci.* 2022; **196**: 263–271.
Publisher Full Text
6. Žaja M, Čavrak I, Lipić T: **Benchmarking apache beam for IoT applications.** *International Convention on Information, Communication and Electronic Technology.* 2021.
Publisher Full Text
7. Goedegebuure A, Kumara I, Driessen S, *et al.*: **Data mesh: a systematic gray literature review.** *ACM Comput Surv.* 2024; **57**(1): 1–36, 11.
Publisher Full Text
8. Rossini E, Pietri M, Picone M, *et al.*: Vulnerable Road Users accident prevention via smart city data fusion: experimental evaluation of a 5G MEC architecture. In: International Symposium on Network Computing and Applications. 2024.
Publisher Full Text
9. Bedogni L, Picone M, Pietri M, *et al.*: Fluid computing in the internet of things: a digital twin approach. In: IEEE Consumer Communications & Networking Conference (CCNC). 2024.
Publisher Full Text
10. Bedogni L, Mamei M, Picone M, *et al.*: **Fluid computing & digital twins for intelligent interoperability in the IOT ecosystem.** *Future Gener Comput Syst.* 2025; **171**: 107855.
Publisher Full Text
11. Picone M, Bedogni L, Pietri M, *et al.*: Digital twins & fluid computing in the edge-to-cloud compute continuum. In: IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). 2024.
Publisher Full Text
12. Picone M, Bedogni L, Pietri M, *et al.*: Dynamic function validation and simulation in fluid digital twins. In: International Symposium on Distributed Simulation and Real Time Applications (DS-RT). 2024.
Publisher Full Text
13. Picone M, Barbone A, Morandi R, *et al.*: From physical to digital: exploring digital twins within the Modena Automotive Smart Area. In: IEEE Consumer Communications Networking Conference. 2025.
Publisher Full Text
14. Rossini E, Bicocchi N, Hadjidimitriou NS, *et al.*: Towards a distributed data mesh model for the IoT-edge-cloud continuum in smart cities. In: IEEE/ACM Symposium on Edge Computing. 2024.
Publisher Full Text
15. Lusvarghi L, Grazia CA, Klapez M, *et al.*: **Awareness messages by Vulnerable Road Users and vehicles: field tests via lte-v2x.** *IEEE Trans Intell Veh.* 2023; **8**(10): 4418–4433.
Publisher Full Text
16. Papadimitratos P, De La Fortelle A, Evenssen K, *et al.*: **Vehicular communication systems: enabling technologies, applications, and future outlook on intelligent transportation.** *IEEE Commun Mag.* 2009; **47**(11): 84–95.
Publisher Full Text
17. Rasouli A, Kotseruba I, Tsotsos JK: **Understanding pedestrian behavior in complex traffic scenes.** *IEEE Trans Intell Veh.* 2018; **3**(1): 61–70.
Publisher Full Text
18. Berge SH, de Winter J, Hagenzieker M: **Support systems for cyclists in automated traffic: a review and future outlook.** *Appl Ergon.* 2023; **111**
Publisher Full Text, | **Publisher Full Text**
19. Feng L, Li W, Lin Y, *et al.*: **Joint computation offloading and URLLC resource allocation for collaborative MEC assisted cellular-V2X networks.** *IEEE Access.* 2020; **8**: 24914–24926.
Publisher Full Text
20. Yang YS, Lee SH, Chen GS, *et al.*: **An implementation of high efficient smart street light management system for smart city.** *IEEE Access.* 2020; **8**: 38568–38585.
Publisher Full Text
21. Longo E, Redondi AEC, Cesana M, *et al.*: **MQTT-ST: a spanning tree protocol for distributed MQTT brokers.** In: *IEEE International Conference on Communications (ICC).* 2020.
Publisher Full Text
22. Staglianò L, Longo E, Redondi AEC: **D-MQTT: design and implementation of a pub/sub broker for distributed environments.** In: *IEEE Int Conf Omnilayer Intell Syst.* 2021.
Publisher Full Text
23. Chockler G, Melamed R, Tock Y, *et al.*: **Constructing scalable overlays for pub-sub with many topics.** In: *ACM Symposium on Principles of Distributed Computing.* 2007; 109–118.
Publisher Full Text
24. Fidler E, Jacobsen HA, Li G, *et al.*: **The PADRES distributed publish/subscribe system.** In: *Principles and applications of distributed event-based systems.* 2010.
Publisher Full Text
25. Beniwal G, Singhrova A: **A systematic literature review on IOT gateways.** *J King Saud Univ-Comput Inf Sci.* 2022; **34**(10): 9541–9563.
Publisher Full Text
26. Toskov B, Toskova A, Bogdanov S, *et al.*: **Intelligent IoT gateway.** In: *International Conference Automatics and Informatics.* 2021.
Publisher Full Text
27. Zhu L, Li B, Tan L: **A vehicular edge computing offloading and task caching solution based on spatiotemporal prediction.** *Future Gener Comput Syst.* 2025; **166**: 107679.
Publisher Full Text
28. Qian P, Wang L, Fan J, *et al.*: **Robust offloading strategy in VEC with computation uncertainty and imperfect CSI.** *Comput Netw.* 2025; **271**: 111597.
Publisher Full Text
29. Dong S, Tang J, Abbas K, *et al.*: **Task offloading strategies for mobile edge computing: A survey.** *Comput Netw.* 2024; **254**: 110791.
Publisher Full Text
30. Jin X, Zhang S, Ding Y, *et al.*: **Task offloading for multi-server edge computing in industrial Internet with joint load balance and fuzzy security.** *Sci Rep.* 2024; **14**: 27813.
31. Shi W, Cao J, Zhang Q, *et al.*: **Edge Computing: Vision and Challenges.** *IEEE Internet Things J.* 2016; **3**(5): 637–646.
Publisher Full Text
32. Mao Y, You C, Zhang J, *et al.*: **A Survey on Mobile Edge Computing: The Communication Perspective.** *IEEE T Veh Technol.* 2017; **19**(4): 2322–2358.
Publisher Full Text
33. Sun Y, Guo X, Song J, *et al.*: **Adaptive Learning-Based Task Offloading for Vehicular Edge Computing Systems.** *IEEE Commun Surv Tutor.* 2019; **68**(4): 3061–3074.
Publisher Full Text
34. Usländer T, Teuscher A: **Industrial data spaces.** *Designing data spaces.* 2022.
Publisher Full Text
35. GAIA-X AISBL: GAIA-X architecture document.
36. Bozkurt Y, Rossmann A, Pervez Z, *et al.*: **Assessing data governance models for smart cities: Benchmarking data governance models on the basis of European urban requirements.** *Sustain Cities Soc.* 2025; **130**: 106528.
37. European Commission: European mobility data space - vision paper. 2023.
Publisher Full Text
38. Fensel D, Şimşek U, Angele K, *et al.*: *Knowledge graphs: methodology, tools, and selected use cases.* Springer; 2021.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✗ ? ? ?

Version 2

Reviewer Report 04 September 2026

https://doi.org/10.21956/openreseurope.25079.r77289

© **2026 Camargo V.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Valter Vieira Camargo
Computing Department, UFSCar - Universidade Federal de São Carlos, São Carlos, Brazil

Summary
The paper presents **NOUS**, an extensible middleware designed to enable the development of interoperable and intelligent urban services through an edge-cloud architecture. According to the authors, NSCA addresses a central challenge in urban digitalization: supporting scalable and efficient communication among heterogeneous actors while integrating with emerging data space infrastructures. The proposed solution works as a platform on which smart city applications can be developed.
The usefulness of the infrastructure is demonstrated through the development of two applications. The first is a service whose goal is to provide notifications about possible collisions between pedestrians and cyclists. The second is also a service that enables localized perception between vehicles and intelligent infrastructures, with the goal of improving mobility intelligence and environmental understanding.
The paper is technically sound. For a paper whose main goal is to present an infrastructure for developing smart city applications, I believe that the level of architectural detail is sufficient. However, I have some concerns.
Organization of the Paper
One issue, in my opinion, is the absence of section numbering. It is quite difficult to navigate through the paper without numbered sections.
Sections 3 and 4 — Edge and Cloud Architecture
Sections 3 and 4 present a well-aligned concept for both the Edge and Cloud layers, effectively leveraging event-driven MQTT for local interactions and SIMPL-based Data Spaces for cloud-level governance.
To further improve these sections, it would be beneficial to provide a little more technical detail about some operational aspects. On the Edge side, a brief discussion of how the system handles transient network disconnections or resource orchestration would help clarify its resilience in dynamic urban environments.
Similarly, providing some additional details about the inter-broker bridge, particularly regarding semantic data mapping, filtering, and traffic management between the Edge and Cloud layers, would give readers a clearer understanding of how the continuum operates seamlessly in

practice.

Related Work

Section 2 (Related Work) provides a useful background on data space initiatives (e.g., SIMPL, FIWARE, and IDSA) and edge-cloud mobility frameworks. However, it currently lacks a systematic, side-by-side comparative analysis between the proposed NSCA architecture and existing state-of-the-art platforms.

To better highlight the main novelty and research contributions of this work, the authors could include a comparative summary table. This comparison could evaluate NSCA against existing edge-cloud middleware according to key technical dimensions, such as data sovereignty and governance, edge publish/subscribe latency, dynamic cross-layer connector mechanisms, and multi-stakeholder interoperability.

Explicitly articulating the specific architectural limitations of existing solutions that NSCA addresses would also help clarify why a new architecture is necessary and would further strengthen the rationale for the proposed approach.

Evaluation Section

Regarding the validation of the proposal, Section 5 currently reads more like a qualitative demonstration of the concept through two specific use cases (VRU Safety and GeoPerception) than a rigorous scientific evaluation of the architecture.

While demonstrating a functional deployment in the MASA living laboratory is a valuable first step, the paper lacks a formal evaluation framework, such as explicitly stated research questions, well-defined baseline comparisons, and systematic experimental protocols, to rigorously assess the architecture's performance under scalable and adverse urban conditions.

To increase the scientific maturity of this section, the authors should strengthen it by incorporating more comprehensive quantitative metrics. Specifically, evaluating critical system aspects such as end-to-end latency, MQTT message throughput, inter-broker connector overhead, network jitter, and resource utilization (CPU/memory at Edge nodes) under varying traffic loads would provide important evidence of NSCA's efficiency and scalability.

Transforming this section from a usage demonstration into a data-driven empirical evaluation would significantly improve the overall technical quality of the paper and strengthen its contribution.

Scientific Contribution

In my opinion, the paper also lacks a more significant scientific contribution. In its current form, the work seems primarily to be an implementation that combines several interesting technologies and concepts, but it does not present a sufficiently clear scientific contribution.

I believe that this aspect could be further strengthened and made more explicit in future versions of the paper.

Is the work clearly and accurately presented and does it cite the current literature?

Partly

Is the study design appropriate and does the work have academic merit?

Partly

Are sufficient details of methods and analysis provided to allow replication by others?

Partly

If applicable, is the statistical analysis and its interpretation appropriate?

No

Are all the source data underlying the results available to ensure full reproducibility?
No source data required

Are the conclusions drawn adequately supported by the results?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: - smart cities

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 07 May 2026

https://doi.org/10.21956/openreseurope.25079.r73076

© **2026 Puliafito A.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Antonio Puliafito
University of Messina, Messina, Italy

This paper presents the NOUS Smart City Architecture (NSCA), a modular middleware designed to support intelligent urban services across the edge–cloud continuum. The architecture combines an MQTT-based publish/subscribe messaging layer at the edge with a SIMPL/Open-compliant Data Space ecosystem in the cloud. Two operational services — the Vulnerable Road User (VRU) Safety Service and the GeoPerception Service — are deployed in the Modena Automotive Smart Area (MASA) testbed to validate the approach experimentally. The paper addresses a relevant and timely problem in urban digitalization, and the real-world deployment provides meaningful practical context. Overall, the manuscript is reasonably well structured and the architecture is clearly motivated. However, several aspects require further clarification or strengthening before the work can be considered fully acceptable.

Overall Recommendation
Approved with Reservations

Major Comments
1. Lack of quantitative comparison with baseline systems
The paper claims advantages in latency, scalability, and modularity over existing platforms such as Apache Kafka-based pipelines and commercial MEC middleware, but provides no quantitative benchmarking against these systems. The related work section discusses alternatives at a high level but does not situate NSCA within a structured comparative framework. At minimum, a

qualitative comparison table outlining the key differences in design choices, capabilities, and trade-offs with respect to representative existing systems would substantially strengthen the contribution and its positioning.

2. Scalability analysis is absent

The experimental evaluation measures end-to-end latency under a limited number of controlled conditions but does not address scalability. The paper targets dense urban deployments with potentially thousands of concurrent assets, topics, and services — yet no results or analysis are provided on how the system behaves as load increases. The authors should provide either experimental measurements of throughput and latency degradation under increasing concurrency, or at minimum a rigorous analytical discussion of expected scalability bounds and known bottlenecks in the current implementation.

3. Security and access control are incompletely specified

The paper mentions tenant-based topic prefixing to avoid namespace collisions and references OAuth2.0/SAML/LDAP for authentication, but explicitly acknowledges that strict enforcement through the identity and access-control framework is still under consolidation. This is a significant limitation for a system intended for multi-stakeholder urban deployments handling potentially sensitive data, including mobility traces and video-derived metadata. The authors should clearly delineate what is currently implemented and tested versus what is planned for future work, and should discuss the security implications of operating the system in its current state.

4. The VRU risk assessment algorithm is underspecified

The VRU service is presented as a key demonstration of the architecture, yet the risk evaluation algorithm — which combines relative distance, speed, and predicted trajectory to assign severity levels — is described only superficially. While the authors acknowledge this is not the paper's primary focus, sufficient algorithmic detail should be provided to allow scientific scrutiny and to distinguish the contribution from prior VRU collision avoidance work cited in the related work section. If the algorithm is described in detail elsewhere, an explicit reference to the corresponding publication should be provided.

5. Application descriptions lack technical depth and concrete real-world grounding

Both the VRU Safety Service and the GeoPerception Service are described at a level of generality that does not adequately reflect the complexity and specificity of real-world urban deployments. The descriptions read as illustrative sketches rather than rigorous accounts of operational systems. In their current form, it is difficult to assess whether the services address realistic traffic scenarios or represent idealised conditions.

For the VRU Safety Service, the paper does not specify what types of road scenarios were actually encountered during the MASA testbed deployment (e.g., uncontrolled intersections, mixed pedestrian-cyclist flows, occlusion conditions, night-time or adverse weather situations). No information is given on the number and variety of VRU interactions observed, the false positive and false negative rates of the detection pipeline, or the end-to-end latency budget required to make a collision warning actionable. Without this, it is impossible to judge whether the system operates within safety-critical time constraints in realistic conditions, or whether it has merely been demonstrated in a controlled, low-density environment.

For the GeoPerception Service, the paper describes a 50-metre vicinity radius as the spatial selection criterion but provides no justification for this choice, nor any discussion of how it was calibrated against actual sensor coverage in MASA. The service is described as aggregating detections from heterogeneous edge cameras and RSUs, but no details are given on how sensor heterogeneity (different fields of view, resolutions, update rates, coordinate reference frames) is handled in practice. It would also be valuable to report how perception quality degrades as a function of distance, object density, or infrastructure availability in the real testbed.

The authors are encouraged to complement the architectural description with concrete, data-driven accounts of the scenarios tested in MASA, including quantitative indicators of service quality (detection accuracy, alert latency distribution, coverage gaps) and a discussion of edge cases or failure modes observed. This would substantially increase the technical credibility of the contribution and demonstrate that the services have been validated against real operational challenges rather than idealised assumptions.

Minor Comments

6. Broker plugin overhead is not directly measured

The paper describes the MQTT broker plugin written in C and claims sub-millisecond processing latency, but no direct micro-benchmark of this overhead is presented. The GeoPerception filtering experiment (Figure 4) measures bandwidth savings but not the computational cost of the plugin itself. A dedicated measurement comparing message processing latency with and without the plugin would make this claim verifiable.

6. GeoPerception overhead lacks measurement conditions

The paper reports an additional latency of approximately 10–20 ms introduced by the query-oriented interaction pattern of GeoPerception compared to a simpler delivery approach. However, this figure is stated to depend on detection density, request frequency, and worker capacity, without specifying the conditions under which it was measured. A more rigorous characterisation of this overhead, including the experimental parameters, would improve the reliability of this claim.

7. Figure 5 readability

Figure 5 contains two side-by-side bar charts with multiple overlapping categories that are difficult to parse at a glance. Clearer axis labels, a legend that more explicitly distinguishes edge vs. cloud asset placement, and a brief explanatory sentence in the caption linking each bar grouping to the deployment configuration would improve readability considerably.

8. Inconsistent use of acronyms NSCA / NCSA

The acronyms NSCA and NCSA appear to be used interchangeably in several places throughout the manuscript (e.g., in the cloud section heading and related text). This appears to be a typographical inconsistency. The authors should perform a final proofreading pass to ensure the acronym is used consistently throughout.

9. Future multi-protocol extensions deserve brief architectural discussion

The conclusion mentions planned extensions to AMQP, DDS, and NGSI-LD but does not discuss the architectural implications of adding these protocols to the current design. Even a short paragraph outlining how the current topic abstraction and broker plugin model would accommodate additional protocols would be informative for readers considering adoption.

Specific Questions for the Authors

1. How many concurrent assets and services were active during the latency experiments? What is the expected degradation as these numbers approach a realistic urban deployment scale?
2. Is the MQTT broker plugin currently deployed and running in production within MASA, or only in a controlled experimental setting?
3. How is the sliding window size in the GeoPerception Redis cache calibrated, and how does the system behave when an asset moves faster than the cache update rate?

Conclusion

The paper addresses an important and well-motivated problem and presents a technically sound architecture with real-world validation in a meaningful testbed. The combination of edge-side MQTT middleware with a Data Space cloud layer is a coherent and timely design. However, the

absence of comparative benchmarking, the incomplete specification of security enforcement, the missing scalability analysis, and the underspecified VRU algorithm prevent full approval at this stage. Addressing the major comments above would substantially strengthen the scientific contribution and the manuscript's suitability for indexing.

Is the work clearly and accurately presented and does it cite the current literature?
Partly

Is the study design appropriate and does the work have academic merit?
Yes

Are sufficient details of methods and analysis provided to allow replication by others?
Partly

If applicable, is the statistical analysis and its interpretation appropriate?
Not applicable

Are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions drawn adequately supported by the results?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Distributed Systems, IoT, Cloud, Smart cities

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 17 March 2026

https://doi.org/10.21956/openreseurope.25079.r71146

© 2026 Reddy S. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Shiva Shankar Reddy
Sagi Rama Krishnam Raju Engineering College (A), Bhimavaram, Andhra Pradesh, India

1. The abstract section needs to be precise and consider only relevant data.
2. The content of the introduction section should be cited from the latest articles.
3. Most of the data was written in a general way. There is no benchmark for the content.
4. Need a proper explanation of how the SIMPL incorporates mechanisms for declarative data

usage policies.
5. Explain clearly about how the SIMPL/Open architecture serves as the foundation upon which the NOUS ecosystem builds its Data Space-compliant services.
6. Explain how the Raw Event Sync component handles events and triggers the synchronisation operations required for updating the Event Repository in the Persistence layer.
7. Need to describe clearly how the MQTT brokers preserve per-topic message ordering and provide reliable delivery through configurable Quality of Service (QoS) levels,
8. Explain how the VRU Service is a real-time safety application designed to mitigate collision risks for pedestrians, cyclists, and other vulnerable road users (VRUs).
9. Need to cite properly how the GeoPerception Service enables localised cooperative perception by aggregating and correlating environmental detection.
10. There is no proper explanation. Need to cross-check the entire article as per reviewer suggestions made.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Data Mining, Medical Mining, Machine Learning, Deep Learning, Edge Compuitng, Cloud computing, Image Processing and IoT

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Version 1

Reviewer Report 22 January 2026

https://doi.org/10.21956/openreseurope.24022.r67714

© 2026 Tsiropoulou E. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Eirini Eleni Tsiropoulou
Arizona State University, Tempe, Arizona, USA

The paper introduces the NOUS Smart City Architecture (NSCA), a comprehensive middleware framework designed to bridge the edge-cloud continuum for intelligent urban services.

Though the manuscript is overall well written and easy to follow there are several aspects that they need improvement and further clarification the revised version of the manuscript.

What is the specific mechanism for configuring this selection (e.g., a YAML file, a dynamic API)?

How is topic namespace collision avoided when multiple independent edge sites (e.g., different

city districts) forward topics to the same cloud broker?

Your broker plugin is a powerful feature for edge-side filtering. Can you describe its internal architecture in more detail? Specifically, does it maintain any state (e.g., for spatial filtering across messages), and if so, how is this state managed in a multi-threaded broker environment?

A more thorough discussion of the current state of the art is needed especially dealing with the uncertainty within the computing environment. Please elaborate more on the risk-aware data offloading in multi-server multi-access edge computing environment.

What is the performance overhead you measured for a plugin performing a spatial filter compared to a simple topic-based subscription?

The Data Space integration relies on an Event Sync component. Could you detail its internal workflow?

Does it consume messages from the cloud broker, buffer them, batch them, and then push them to the SIMPL/Open EDC?

How does it handle message ordering guarantees and delivery assurances (at-least-once, exactly-once) between the MQTT world and the Data Space world?

It would be great if the authors can address those comments in the revised version of the manuscript.

Is the work clearly and accurately presented and does it cite the current literature?
Yes

Is the study design appropriate and does the work have academic merit?
Yes

Are sufficient details of methods and analysis provided to allow replication by others?
Yes

If applicable, is the statistical analysis and its interpretation appropriate?
Yes

Are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions drawn adequately supported by the results?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: reinforcement learning

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 12 Feb 2026

Marco MAMEI

We thank the reviewers for their constructive and insightful comments on our manuscript. Below, we report the reviewers' comments (**in bold**) followed by our corresponding responses. We submitted a revised version of the paper to address reviewer's comments. **R2.1 What is the specific mechanism for configuring this selection (e.g., a YAML file, a dynamic API)?** Thank you for the clarification request. Broker-level selection and topic forwarding are configured through standard EMQX configuration files, with the initial setup typically performed via the broker's graphical administration interface. Dynamic configuration of assets, services, and observers—both at the edge and cloud—is instead handled through REST APIs exposed by NSCA and a web application that interacts with these APIs. This separation enables stable infrastructure configuration at the broker level while allowing runtime management through higher-level services. We clarified this point in the manuscript. **R2.2 How is topic namespace collision avoided when multiple independent edge sites (e.g., different city districts) forward topics to the same cloud broker?** Topic namespace collisions are avoided by enforcing a tenant-based prefix in the topic hierarchy. In practice, each organization (e.g., a city or city district) is assigned a unique tenant identifier, which is prepended to all topics forwarded to the cloud broker (i.e., all topic trees are namespaced as //...). This ensures that independent edge sites can forward overlapping internal topic structures without collisions at the cloud level. We note that strict enforcement of tenant prefixes by the user/identity layer is currently under consolidation and is part of our planned future work. Similarly, seamless user migration across edge areas (potentially involving different tenants) is an ongoing extension we plan to address in future versions of the platform. **R2.3 Your broker plugin is a powerful feature for edge-side filtering. Can you describe its internal architecture in more detail? Specifically, does it maintain any state (e.g., for spatial filtering across messages), and if so, how is this state managed in a multi-threaded broker environment?** Thank you for the question. In our prototype, spatial filtering is not implemented inside the broker plugin. Instead, spatial selection is handled by the GeoPerception service (and reused by VRU), which issues geospatial queries to a dedicated in-memory cache of recent object locations. Concretely, GeoPerception maintains a short sliding window (last few seconds) of the latest positions for each tracked object (keyed by its infrastructure-level identifier) and, upon each user update, retrieves only the objects within a radius around the user via Redis geospatial indexing (e.g., GEOADD to update positions and GEOSEARCH/radius queries to select nearby objects). This design keeps the broker plugin *stateless* (avoiding shared mutable state inside the broker execution context) and moves state management to an external component (Redis), which is naturally safe to access from multiple worker threads/processes via atomic commands. We clarified in the revised text that the plugin is used for lightweight transformations/routing, while spatial filtering is realized as a stateless worker pattern in the services, backed by an external geospatial cache. **R2.4 A more thorough discussion of the current state of the art is**

needed, especially dealing with the uncertainty within the computing environment. Please elaborate more on the risk-aware data offloading in multi-server multi-access edge computing environment. Thank you for the suggestion. We agree that uncertainty (e.g., time-varying wireless conditions, mobility, and fluctuating MEC server load/queueing) is central in multi-server MEC offloading. In the revised Related Work, we expanded the discussion to cover (i) risk-aware offloading formulations that explicitly model uncertainty and service-outage risk (e.g., multi-server/multi-access risk-aware offloading via game/optimization perspectives), and (ii) broader task-offloading surveys that summarize uncertainty sources and mitigation strategies (robust/stochastic optimization, queue-aware scheduling, and risk-sensitive objectives). At the same time, we clarified that NSCA does not yet implement an automatic risk-aware offloading optimizer: this remains future work. Our planned extension is to exploit NSCA's monitoring of MEC load/pressure to adaptively decide what runs at the edge/MEC versus in the cloud—migrating *non-real-time, resource-demanding* services to the cloud during congestion to preserve MEC resources for latency-critical pipelines (e.g., safety services). In the current implementation, a "simple topic-based subscription" is not a viable baseline for the spatial selection we target: filtering purely via topic hierarchies would not scale in dense deployments and would complicate access control (ACL) and safe exposure of fine-grained sensor streams. Instead, GeoPerception exposes a query-style API that (i) selects only objects in the user's vicinity and (ii) standardizes heterogeneous payloads (e.g., different formats from cameras vs. RSUs) into a uniform output, simplifying downstream consumption. This additional service step introduces a modest overhead: in our deployment we observed an extra ~10–20 ms per request compared to delivering already-prepared data locally. The exact overhead varies with the amount of incoming detections, request rate, and available worker capacity (i.e., service load/queueing). We clarified this trade-off (slightly higher latency in exchange for scalability, controlled exposure, and data normalization) in the revised text. **R2.5 The Data Space integration relies on an Event Sync component. Could you detail its internal workflow?** The Event Sync component supports the publication of data assets according to the standardized workflow defined by the Data Space protocol. For an existing dataset stored in the Event Repository, the provider uses the Event Sync component to define and attach the required metadata describing the dataset's content and structure. These metadata make the dataset understandable and discoverable to potential consumers. The dataset is then exposed through a certified connector, where usage policies, access conditions, and contractual terms are defined.

R2.6 Does it consume messages from the cloud broker, buffer them, batch them, and then push them to the SIMPL/Open EDC? The Raw Event Sync component bridges MQTT event streams and the SIMPL/Open Data Space. It subscribes to selected MQTT topics on the cloud broker and persistently stores incoming events in the Event Repository, building a historical view for each topic. Using the Event Sync component, as described above, these datasets are then registered as data assets in the EDC Control Plane with appropriate data-sharing policies and delivered to authorized consumers through the EDC Data Plane as historical batches.

R2.7 How does it handle message ordering guarantees and delivery assurances (at-least-once, exactly-once) between the MQTT world and the Data Space world? MQTT brokers preserve per-topic message ordering and provide reliable delivery via configurable QoS. The Raw Event Sync component immediately persists received events in the repository as an ordered log, ensuring durable storage. The Event Sync component

reads from this repository, so ordering is preserved, and delivery through the EDC Data Plane provides at-least-once end-to-end transfer while preventing data loss through persistent buffering.

Competing Interests: No competing interests were disclosed.

Reviewer Report 16 January 2026

https://doi.org/10.21956/openreseurope.24022.r67717

© 2026 Reddy S. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Shiva Shankar Reddy
Sagi Rama Krishnam Raju Engineering College (A), Bhimavaram, Andhra Pradesh, India

1. See that the abstract section contains with the introduction, data used, objectives, models used, results, and conclusion.
2. The content was written in a general way. Need to cite the data with the latest articles. See that the paragraphs should have the connectivity in between them.
3. The subheadings of Figure 1 and 2 need a clear explanation of each.
4. Explain how far the NSCA supports multi-role participation by allowing a single application to assume multiple roles concurrently.
5. How the interaction between assets and services is the observer role.
6. How far the Dynamic plugin and filtering enhance flexibility and control over message handling within SCA.
7. Explain how NSCA empowers to incorporate cooperative edge intelligence without modifying client applications or services.
8. Explain how the NOUS Data Space acts as the cloud-side extension of NSCA, how it enables real-time data, and how to create value-added services.
9. Explain how the interactions between data space participants are secured through the Data Space Common Services layer.
10. A brief explanation is needed for representative services and how they differ in interaction models, communication patterns, and deployment scenarios within the MASA testbed.
11. How the writer evaluates proximity risk by considering relative distance, speed, and predicted trajectory, assigning each situation a severity level.
12. Explain how the GeoPerception Service enables the localized cooperative perception by aggregating relevant environmental detections around a mobile user to enhance situational awareness.
13. Explain how figures 3, 4, and 5 were generated. How the values were obtained for the tables.
14. The conclusion section should be precise and should be covered with the results obtained.
15. The discussion section should be added by citing the latest articles. Need to prove that the proposed approaches were better compared with the existing ones.

Is the work clearly and accurately presented and does it cite the current literature?
No

Is the study design appropriate and does the work have academic merit?
Partly

Are sufficient details of methods and analysis provided to allow replication by others?
Partly

If applicable, is the statistical analysis and its interpretation appropriate?
Partly

Are all the source data underlying the results available to ensure full reproducibility?
Partly

Are the conclusions drawn adequately supported by the results?
No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Data Mining, Medical Mining, Machine Learning, Deep Learning, Edge Compuitng, Cloud computing, Image Processing and IoT

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 12 Feb 2026

Marco MAMEI

Review Response Letter We thank the reviewers for their constructive and insightful comments on our manuscript. Below, we report the reviewers' comments (**in bold**) followed by our corresponding responses. We submitted a revised version of the paper to address reviewer's comments. **R1.1. See that the abstract section contains with the introduction, data used, objectives, models used, results, and conclusion.** The abstract actually follows this template, the only missing element is "data used" since this is an architecture/system description we do not rely on a specific dataset. *Intro*: "This paper introduces the NOUS Smart City Architecture (NSCA), an extensible middleware designed to enable intelligent, interoperable urban services across the edge–cloud continuum. Developed within the NOUS project" *Objectives*: "NSCA addresses a central challenge in urban digitalization: supporting scalable, efficient communication among heterogeneous actors while integrating with emerging data space infrastructures." *Models used* (i.e. System Description in our context): "On the edge side, NSCA builds on the lightweight …. On the cloud side, it integrates a Data Space Ecosystem … This makes the architecture suitable for a wide range of smart city verticals, including road safety, mobility management, *environmental monitoring, and citizen services." Results/conclusion*: "The platform's practical

value is demonstrated through two operational services deployed in a real-world smart city testbed. Together, these services showcase NSCA's ability to integrate IoT data flows, edge analytics, and data-space–enabled cloud services, supporting next-generation, interoperable urban applications" Hope that this clarifies the abstract structure.

R1.2. The content was written in a general way. Need to cite the data with the latest articles. See that the paragraphs should have the connectivity in between them. Thank you for the suggestion. We substantially revised the **Related Work** section to improve technical depth, citation coverage, and narrative continuity between paragraphs. In particular, we expanded the bibliography with recent and authoritative studies on uncertainty-aware MEC offloading, learning-based adaptive offloading strategies, and data-governance frameworks for European Data Spaces. The revised text now connects foundational edge–cloud architectures with emerging challenges such as risk-aware service placement, dynamic edge/cloud partitioning, and governance-driven data sharing. We also restructured the section to strengthen logical transitions between topics—starting from edge/MEC communication architectures, moving to safety-critical mobility services, and finally positioning NSCA within Data Space–enabled interoperability. These additions provide clearer contextualization of our contribution with respect to the latest research trends while improving readability and coherence across paragraphs.

R1.3. The subheadings of Figure 1 and 2 need a clear explanation of each. We notably extended and improved Figures' caption

R1.4. Explain how far the NSCA supports multi-role participation by allowing a single application to assume multiple roles concurrently. This is an important aspect. We clarify in Section 2 and Section 3 that NSCA supports multi-role and individual serves can participate with multiple-roles concurrently.

R1.5. How the interaction between assets and services is the observer role. The observer role in this architecture functions as a mediating layer that continuously monitors asset behavior and service invocations, collecting state and event information from assets and interpreting it to coordinate system behavior. By observing changes in asset state or data production, the observer propagates relevant updates to appropriate services, while also supporting the dynamic binding of services to assets with the required capabilities. In this way, the observer acts as an interface between physical assets and logical services, enabling coherent, responsive, and adaptive interactions within the edge–cloud environment. We clarify these aspects in the paper

R1.6. How far the Dynamic plugin and filtering enhance flexibility and control over message handling within SCA. Dynamic plugin and filtering capabilities extend the NSCA messaging infrastructure by allowing broker-level extensibility that can be modified at runtime, rather than being fixed at design time. These mechanisms provide greater flexibility—because the system can adapt to varying data flows and application needs—and greater control—because message handling policies can be fine-tuned and enforced at the broker level to meet specific functional or performance requirements within the edge–cloud NSCA environment. We clarify these aspects in the paper

R1.7. Explain how NSCA empowers to incorporate cooperative edge intelligence without modifying client applications or services. We thank the reviewer for pointing out this important point. We clarified in the paper that we refer to *edge intelligence* as a set of distributed capabilities embedded in the edge infrastructure that observe, analyze, and act on data streams in proximity to their sources, enabling cooperative behaviors across edge nodes while remaining transparent to applications and services. NSCA enables the

incorporation of cooperative edge intelligence by introducing coordination mechanisms within the communication and orchestration layer, rather than inside client applications or services themselves. By embedding functions such as dynamic plugins, message observation, filtering, and inter-broker coordination at the edge infrastructure level, NSCA can analyze, enrich, aggregate, or redirect data flows transparently.

R1.8. Explain how the NOUS Data Space acts as the cloud-side extension of NSCA, how it enables real-time data, and how to create value-added services. The NOUS Data Space acts as the cloud-side extension of NSCA by aggregating and governing data produced at the edge, while NSCA handles real-time coordination close to data sources. Continuous ingestion of NSCA data streams enables near–real-time access through standardized and controlled interfaces. On this basis, value-added services can be built by combining live and historical data with analytics or AI, transforming edge observations into cross-domain insights and decision-support services. We tried to better highlight the role of NOUS data space in the paper

R1.9. Explain how the interactions between data space participants are secured through the Data Space Common Services layer. Interactions between data space participants are secured through the Data Space Common Services layer by enforcing standardized identity management, authentication, authorization, and usage-control policies, ensuring that data is exchanged only between trusted parties and strictly according to agreed access and governance rules. From an implementation point of view the system is built on industrial standards: OAuth2.0, SSO/SAML/LDAP. We expanded this part in the paper.

R1.10. A brief explanation is needed for representative services and how they differ in interaction models, communication patterns, and deployment scenarios within the MASA testbed. We expanded the VRU and GeoPerception descriptions to clarify how they represent distinct service types within the MASA testbed, differing in interaction models (event-driven vs. aggregative), communication patterns (low-latency push vs. hybrid streaming/query), and deployment scenarios (edge-centric vs. edge–cloud).

R1.11. How the writer evaluates proximity risk by considering relative distance, speed, and predicted trajectory, assigning each situation a severity level. The VRU service evaluates proximity risk by combining relative distance, speed, and predicted trajectories of interacting entities to estimate potential conflicts and assign a corresponding severity level; we clarify this algorithmic approach in the manuscript and emphasize that its parameters are fully customizable to adapt to different operational contexts and safety requirements. However, we also stress that the specific VRU risk evaluation algorithm is not the focus of this paper and is presented only as an example, while the core contribution lies in how NSCA supports and integrates such services at the system level.

R1.12. Explain how the GeoPerception Service enables the localized cooperative perception by aggregating relevant environmental detections around a mobile user to enhance situational awareness. The GeoPerception Service enables localized cooperative perception by aggregating and correlating environmental detections from nearby edge nodes around a mobile user to enhance situational awareness, effectively acting as the first stage of the VRU application pipeline. We tried to better explain its workin in the paper.

R1.13. Explain how figures 3, 4, and 5 were generated. How the values were obtained for the tables. We notably extended the description of the experiments and how figures have been generated.

R1.14. The conclusion section should be precise and should be covered with the results

obtained. We added more precise references to the obtained results
R1.15. The discussion section should be added by citing the latest articles. Need to prove that the proposed approaches were better compared with the existing ones. We extended the analysis of the state of art with recent references. See also answer n.2

Competing Interests: No competing interests were disclosed.