Journal of Urban Development and Smart Cities
2025, 2(1), 210-221

UNIFIED KNOWLEDGE GOVERNANCE FOR MULTITENANT SMART CITY PLATFORMS: SEMANTIC MODELING, OPERATIONAL DIAGNOSTICS, AND SCALABLE REUSE

Arza Churchman

Contemporary smart city infrastructures no longer function as isolated vertical systems. They are increasingly organized as shared, multi-organizational digital environments in which heterogeneous data streams, analytics services, dashboards, workflows, and user-facing applications are reused across multiple operational domains and jurisdictions. This architectural shift improves scalability and lowers duplication, but it also makes platform governance substantially more complex: when data, processes, and interfaces are interdependent, operators must be able to identify the origin of service disruptions, trace the downstream impact of changes, and support rapid development without losing control over platform integrity. This paper presents a unified knowledge framework for governing multitenant smart city platforms, centered on a semantic model that explicitly represents relationships among data, processing components, dashboards, users, and organizations. The framework is implemented in the open-source SNAP4CITY *ecosystem and supports visual navigation, linked-data publication, semantic querying, and operational inspection through a dedicated Data Inspector and SPARQL-accessible knowledge graph. The model is validated in real smart-city scenarios, including shared-data smart parking workflows and interactive mobility-and-environment control-room applications for what-if analysis, alert generation, and operator action. The production-scale evidence demonstrates that the framework supports complex deployments at urban and regional scale: the largest production environment runs on 48 virtual machines and manages 20 organizations, about 7,500 users, about 2,500 developers, 1,638 dashboards, 10,940 active widgets, 260,761 distinct data sources, 415 IoT applications, 82 data-analytics processes, and 8 web-scraping processes. Semantic monitoring further shows that 100 dashboards are empty (6.1%), 80% of dashboard access time is concentrated in only 21 dashboards, and 41 data sources are reused in at least 20 distinct contexts. The findings show that explicit semantic governance is not merely a modeling convenience; it is a practical urban-infrastructure capability that improves maintainability, reuse, transparency, and operational resilience in large smart city deployments.*

Index Terms — smart city; multitenancy; knowledge graph; ontology engineering; platform governance; digital twins; urban informatics; semantic interoperability

https://doi.org/10.66033/judsc2025-221

INTRODUCTION

Urban digital transformation increasingly depends on platformized infrastructures rather than isolated applications. Municipal services, regional observatories, control rooms, mobility tools, environmental dashboards, and citizen-facing mobile interfaces are now commonly developed on shared data and service layers. This shift is technically advantageous: it reduces repeated ingestion pipelines, lowers maintenance costs, and makes it possible to deliver "smart city as a service" across multiple cities, departments, and stakeholder groups. Yet it also changes the operational problem. When many applications reuse the same data, analytics, and business-logic components, failures and changes become relational rather than local.

In practical terms, a smart city operator often needs to answer questions that are not well served by conventional monitoring dashboards alone. If a visualization stops updating, is the issue caused by a missing data feed, an upstream transformation process, a failing IoT application, a permissions change, a communication error, or a dependency created by another user? If a new service is being developed, which data and APIs are already available, who owns them, and what existing processes should be reused instead of duplicated? If a shared data source changes or expires, which dashboards, widgets, or downstream applications will be affected? These questions are central to the governance of smart urban infrastructure, particularly in multi-organizational settings.

The core argument of this paper is that such environments require a formal knowledge layer that represents not only city entities and sensor semantics, but also the operational structure of the platform itself. A semantic model that links data, processes, dashboards, users, and organizations can transform platform management from a fragmented, tool-by-tool activity into a traceable and inspectable knowledge-driven process.

The framework presented here addresses three practical objectives:

- **O1:** support early identification of faults and dysfunctions by making dependencies among data, processes, and interfaces explicit;
- **O2:** enable developers to discover reusable data, services, and APIs, thereby reducing duplicated implementation effort;
- **O3:** provide operators with platform-wide visibility over resource usage, application complexity, and cross-organizational coupling.

These objectives place the work directly within the scope of smart-city development and urban digital governance. The contribution is not limited to ontology design in the abstract; it concerns the management of real urban service infrastructures in which data, control, and collaboration must remain coordinated at scale.

The remainder of the paper is organized as follows. Section positions the framework in relation to prior smart-city, IoT, and semantic-platform literature. Section explains the operational requirements that motivate the model. Section summarizes the SNAP4CITY platform context. Section presents the unified knowledge model. Section demonstrates the framework in representative urban scenarios and inspection workflows. Section reports deployment-scale quantitative evidence. Sections and discuss implications and conclude.

RELATED WORK

Smart-city infrastructures have matured through several overlapping streams of research: data ingestion and warehousing, IoT interoperability, semantic modeling, microservice architectures, and digital-twin-oriented analytics. A long-standing challenge across these streams is that most frameworks solve only one layer of the problem.

Large-scale smart-city platforms typically begin with heterogeneous data ingestion, storage, and API access. These infrastructures are effective for collecting, indexing, and serving data, but many remain oriented toward single processes or single vertical domains rather than the management of interdependent multi-application ecosystems. Open-source smart-city architectures, including SNAP4CITY, have shown that city-scale services can be built through modular microservices, shared APIs, and reusable dashboarding components [2, 3]. However, as reuse increases, the operational complexity of the platform also increases.

Semantic approaches have substantially improved interoperability in smart-city and IoT environments. Models such as KM4CITY and related ontology-based approaches support the representation of city entities, spatial-temporal relationships, and heterogeneous sensor data [4]. More broadly, knowledge-graph approaches have reinforced the importance of explicit semantics for cross-domain integration, discoverability, and reasoning [6]. At the IoT infrastructure level, standards and platforms such as FIWARE have further advanced interoperable data exchange [5].

Yet an important gap remains. Most smart-city ontologies focus on *what* exists in the city—sensors, places, observations, devices, transport entities, and urban phenomena. They are less effective at representing *how* smart-city applications themselves are composed: which dashboards depend on which data, which applications generate or consume which outputs, how developers and organizations share resources, and where an operational fault propagates through a chain of reuse.

This distinction matters in large multitenant systems. A city platform may be semantically rich at the data level and still lack adequate support for tracing dependencies among dashboards, widgets, IoT apps, data analytics, and developer-owned resources. Process-aware industrial knowledge graphs move closer to this concern [7], but industrial process contexts differ materially from urban multitenancy, where public-facing interfaces, open-data reuse, shared control-room workflows, and cross-organizational access are routine.

Digital-twin research further sharpens the need for integrated governance. As smart-city platforms support simulation, forecasts, and what-if analysis, they require not only synchronized data but also reliable traceability across the entire application stack [8]. In that context, a platform-level knowledge model becomes a governance instrument: it links urban data infrastructure with operational accountability.

This paper addresses that need by presenting a semantic framework focused on platform entities and their interrelations. Its novelty lies not in replacing established city ontologies, but in complementing them with a governance-oriented layer that makes the structure of multitenant smart-city applications inspectable, reusable, and controllable.

OPERATIONAL REQUIREMENTS FOR URBAN PLATFORM GOVERNANCE

The requirements that motivate the framework arise from the realities of shared smart-city deployments. In regional and metropolitan settings, multiple cities and agencies often depend on the same data resources: weather forecasts, mobility feeds, environmental observations, public-transport data, telecommunications-derived movement indicators, GIS layers, and other contextual services. A multitenant architecture allows these resources to be loaded and maintained once, then reused across many applications. This reduces duplication and enables more sustainable operation, but it also creates dependency chains that are difficult to manage without a unifying model.

Three operational requirements are especially important.

Conceptual operating chain of the platform

Heterogeneous urban data sources and IoT devices → real-time ingestion and storage → knowledge-base indexing (KM4CITY + UKM) → SCAPI and microservices → IoT apps and data-analytics processes → dashboards, control-room tools, mobile apps, and what-if interfaces.

The UKM overlays this chain by making relationships among data, processes, interfaces, users, and organizations explicit, queryable, and visually inspectable.

Figure 1: High-level logic of semantic governance in a multitenant smart-city platform.

Traceability of Problems and Service Disruptions

When a dashboard or user-facing application fails, operators need to trace the cause quickly. In a smart-city environment, the source of failure may reside in a data provider, an ingestion connector, an IoT broker, an analytics process, a business-logic flow, a storage layer, an API endpoint, or a change introduced by another developer. Without a unified representation of relationships, even simple failures can become time-consuming to diagnose.

Reusable Development Across Shared Infrastructures

Multitenant smart-city platforms are most effective when developers can discover what already exists and build on it. A new scenario should not require rebuilding a data connector, re-ingesting a dataset, or recreating a service that is already available elsewhere on the platform. Operators therefore need a model that exposes references to data, processes, and APIs, while developers need guided access to reusable assets.

Platform-Wide Monitoring and Resource Governance

As the number of users, organizations, and applications grows, platform operators require a governance layer that reveals usage concentration, underused assets, dependency hotspots, and cross-organizational coupling. This is particularly important in environments that support free trials, third-party development, and multiple tenants, where the underlying platform must remain robust despite heterogeneous patterns of use.

These requirements are not confined to technical convenience. They affect cost control, service continuity, transparency, and the ability of public-sector and civic actors to rely on shared digital infrastructure as part of urban governance.

PLATFORM CONTEXT: THE SNAP4CITY ARCHITECTURE

The framework is implemented in the open-source SNAP4CITY ecosystem, which provides the operational context for the unified knowledge model. SNAP4CITY supports multiple geographic areas, cities, topics, and tenants (organizations) on the same platform. Each organization may include developers, decision-makers, and end users. The platform manages heterogeneous data sources from external services, open-data portals, data providers, IoT networks, and multiple push/pull protocols.

Data are processed in real time and indexed in OpenSearch, while semantic indexing is handled through a knowledge base implemented as an RDF triple store. The underlying city-data layer is grounded in KM4CITY, which models spatial, temporal, and relational aspects of urban entities [4]. On top of that city-data layer, the present framework adds a semantic representation of platform entities and their relationships.

Business logic and workflow orchestration are implemented through containerized services and Node-RED-based IoT applications, supported by more than 180 platform microservices. These components may run on premises, at the edge, or in the cloud. Historical and real-time data, as well as derived knowledge, are made available to front-end services, analytics, and applications through SCAPI and microservices. This architecture enables advanced urban functions such as prediction, simulation, anomaly detection, early warning, dynamic routing, and what-if analysis [2, 10, 9].

In such a platform, the technical challenge is not merely interoperability. It is the ability to maintain visibility across a dense network of reused components. The UKM is designed to meet that challenge.

UNIFIED KNOWLEDGE MODEL

Design Principles

The unified knowledge model is a semantic layer for representing platform entities and the relationships among them. It is designed as linked data: classes, properties, and instances are published in a form that can be inspected both by humans and by machines. The model is therefore both a formal ontology and a practical governance resource.

The design follows three principles:

1. **Platform-centered representation**: the model captures not only urban data, but also the structure of applications, dashboards, widgets, services, and administrative entities.
2. **Navigable relationships**: dependencies among entities are explicit, enabling visual browsing, Data Inspector traversal, and SPARQL querying.
3. **Operational usefulness**: the model is designed to support fault analysis, reuse discovery, monitoring, and platform-wide assessment.

Core Class Structure

At the highest level, the model is organized around a root class, `s4cThing`, which acts as the parent for the three main semantic branches: processing entities, data entities, and administrative entities.

This structure is important because it places application-layer artefacts inside the same semantic space as data and administration. Dashboards, widgets, IoT apps, analytics processes, devices, storage, and organizations can therefore be traversed through a common graph rather than through separate tools.

Relationship Semantics

The value of the model lies in its relationship vocabulary. Properties such as `hasWidget`, `useData`, `useIotApp`, and `generates` make dependencies explicit. A dashboard can be linked to its widgets; a widget can be linked to the data it consumes; an IoT app can be linked both to the data it uses and to the outputs it generates. Administrative relations expose ownership, grouping, and organizational structure.

This explicit relation layer allows operators and developers to move in either direction through the graph:

Table 1: Core semantic structure of the unified knowledge model.

Class	**Role in the platform**
`s4cThing`	Root class representing any platform entity managed in the semantic layer.
`ProcessingThing`	Parent class for processing elements that ingest, transform, compute, render, or expose information.
`ContainerizedApp`	Dynamically deployed application class, specialized into executable services.
`DataAnalytic`	Containerized analytics process, typically implemented via Python or R scripts.
`IotApp`	Node-RED-based IoT application for business logic, data flows, and event-driven processing.
`PortiaCrawler`	Web-crawling process for extracting external web data into the platform.
`DataSource`	Entity that provides data into the platform.
`DataProviding`	Entity that exposes data outside the platform.
`Storage`	Dual-role class functioning as both a data source and a data-providing element.
`IotDevice` / `IotBroker`	Device and broker entities involved in real-time sensing and management.
`Dashboard`, `Widget`, `Synoptic`, `ExternalService`	User-interface and service-facing entities that consume, display, or expose data and results.
`Data`	Semantic representation of managed information objects, including sensors, KPIs, heatmaps, POIs, traffic flows, and related data artefacts.
`AdministrativeThing`	Parent class for entities used to administer data and processing elements.
`User`, `Organization`, `UserGroup`, `Group`	Administrative entities representing users, tenants, user groups, and grouped data resources.

- from a data source to the dashboards, widgets, and applications that depend on it;
- from a dashboard or application back to the data it uses or produces;
- from a service failure to all potentially affected downstream artefacts;
- from a resource to the users or organizations associated with it.

In operational terms, the model replaces fragmented lookup with traversable knowledge.

Publishing and Access

The ontology and its instances are published as linked data, and the resulting graph can be accessed through multiple interfaces: a linked open graph visualization, the platform Data Inspector, and a public SPARQL endpoint. This multi-modal access is essential for usability. Developers may prefer direct semantic querying; operators often benefit from guided visual or faceted navigation.

VALIDATION IN REPRESENTATIVE URBAN SCENARIOS

Scenario 1: Shared-Data Smart Parking and Traffic Services

The first validation scenario concerns smart parking and traffic-information services. Two solutions coexist on the same platform. The first collects contextual, historical, and real-time information on parking status and traffic flow, along with external weather forecasts, stores them, semantically indexes them, and exposes them through SCAPI for dashboard and mobile-app consumption. The second solution reuses those same data and combines them with analytics processes to produce traffic-flow and parking predictions [11].

This scenario demonstrates a core property of the framework: multiple applications can share data and processing dependencies, and those shared relations are semantically represented rather than implicitly buried in code and configuration. As a result, operators can identify which downstream services depend on a specific data flow, and developers can discover reusable inputs instead of rebuilding parallel pipelines.

Scenario 2: Interactive Mobility, Environment, and What-If Control Room

The second scenario is more complex and better reflects urban governance needs. It supports a control-room application for mobility and environment management that integrates historical traffic, weather, and air-quality data; prediction services; critical events reported by field operators; alert generation; dynamic routing; and operator-driven what-if analysis.

In this setting, authorized users can define road constraints, create and save alternative traffic scenarios, run simulations, and eventually trigger real-world actions such as issuing messages to mobile apps, connected-drive devices, and variable message panels. The scenario therefore connects data ingestion, analytics, dashboards, operator input, simulation, and actuation inside a single platform workflow.

This is precisely the type of urban-management context in which semantic governance becomes critical. A what-if dashboard, an alerting dashboard, and the IoT apps they trigger may share common data objects and upstream services. Without explicit semantic traceability, understanding the operational structure of such a system quickly becomes difficult.

Visual Graph Inspection

The platform provides graph rendering of the UKM through an integrated linked-data browser. In the mobility-and-environment control-room scenario, the graph reveals how dashboards, selector widgets, IoT apps, and shared data objects are interlinked. This is valuable not only for documentation but also for live operational inspection: shared data dependencies become visually obvious, and developers can identify where changes in one application may affect others.

Data Inspector for Operational Navigation

The Data Inspector serves as the primary guided interface for navigating the UKM. Through faceted search, full-text search, and map-based filtering, users can inspect a data object and immediately retrieve its attributes, values, ownership, licensing status, and health-related metadata. They can then move to connected artefacts, such as the KPI editor or the IoT applications that consume or produce the selected resource.

This capability is operationally significant. A developer can verify whether a resource is stable and reusable

Table 2: Production-scale profile of the largest SNAP4CITY deployment.

Measure	Value
Cloud virtual machines	48
Organizations (tenants)	20
Users	$\sim$ 7,500
Developers	$\sim$ 2,500
Dashboards	1,638
Event-driven dashboards linked to IoT apps	525
Active widgets (including synoptics)	10,940
Average new incoming data messages per day	$\sim$ 1.8 million
Distinct data sources	260,761
IoT-device-originated data sources	23,134
Registered sensors and actuators	125,086
IoT brokers	19
Heatmaps with time series	361
Traffic flows with time series	23
Registered external services	180
IoT apps	415
Data-analytics processes	82
Web-scraping processes	8

before integrating it. An operator can inspect whether a resource interruption is local or downstream from another process. The same semantic graph therefore supports both development and governance.

Semantic Querying for Dependency Analysis

The model also supports direct SPARQL queries. For example, a traffic sensor in Florence with ID `METRO11` can be used as an anchor point to identify all dashboards, widgets, and IoT apps that depend on it. A typical query traverses from dashboards to widgets through `hasWidget`, and from widgets or IoT apps to the sensor through `useData`. Similar queries can be used to inspect the impact of an IoT app fault, enumerate affected services, or detect heavily reused resources.

This querying capability turns the semantic layer into a platform-analysis engine. It supports fault tracing, impact assessment, and portfolio-level monitoring without requiring manual reconstruction of dependencies.

PLATFORM-SCALE QUANTITATIVE EVIDENCE

Production Deployment Profile

The production deployment demonstrates that the framework is not a laboratory prototype. At the time of assessment, the largest production environment of the SNAP4CITY framework ran on 48 virtual machines in the cloud and supported a genuinely large multitenant ecosystem.

These figures make the governance challenge concrete. The problem is not simply to model a few devices or a single application; it is to maintain visibility and control across a highly heterogeneous urban digital ecosystem in which thousands of users, hundreds of executable components, and hundreds of thousands of

Table 3: Examples of platform-level operational statistics derived through semantic querying.

Query Type	Observed Result
Empty dashboards	100 of 1,638 dashboards are empty (6.1%).
Most-used dashboards	80% of total dashboard access time is concentrated in the 21 most-used dashboards.
Least-used dashboards	Only 50 of 1,638 dashboards have an access time of 0 minutes (3.0%); 525 dashboards are event-driven from IoT apps.
Most-reused data	41 of 260,761 distinct data sources are used in at least 20 different contexts or use cases.
Cross-organizational IoT cohesion	137 IoT apps produce data for two different organizations.

data sources coexist.

Semantic Monitoring Statistics

Beyond deployment scale, the semantic layer enables platform-wide operational statistics that would be difficult to obtain consistently without explicit graph relations.

These numbers are analytically meaningful in several ways.

First, usage concentration reveals operational criticality. If 80% of dashboard access time is concentrated in only 21 dashboards, then maintaining those dashboards and their dependencies is disproportionately important for service continuity.

Second, semantic reuse becomes measurable. The fact that 41 data sources are reused in at least 20 contexts confirms that cross-scenario dependency is not incidental; it is a structural property of the platform. This is exactly why a model of relationships is necessary.

Third, cross-organizational sharing can be quantified. When 137 IoT applications produce data for two organizations, operational governance is no longer merely a technical concern within a single team. It becomes an issue of multi-actor coordination inside urban digital infrastructure.

Operational Query Portfolio

The framework supports a broad portfolio of platform analyses, including:

- cohesion among solutions and organizations;
- identification of data dependencies needed to create a new service;
- impact assessment when a dataset changes or becomes unavailable;
- complexity assessment of dashboards and multi-dashboard solutions;
- identification of dashboards with faults in specific connections;
- detection of empty, unused, public, or private dashboards over defined time windows;

- discovery of the most-used data, most-used dashboards, and most crucial IoT apps;
- counting dashboards that consume a given dataset, use outputs of a given IoT app, or act through business logic.

Taken together, these capabilities show that the UKM functions as a governance layer for urban digital infrastructure rather than as a static metadata registry.

DISCUSSION

Why Platform Semantics Matter in Urban Development

Urban development is increasingly inseparable from digital infrastructure. Smart mobility, environmental monitoring, control-room applications, and citizen-facing services depend on data pipelines, reusable APIs, and shared analytics. In such contexts, governance capacity depends not only on technical performance, but also on the ability to understand interdependence.

The unified knowledge model addresses this need by making the *operational architecture* of the smart-city platform visible. Its importance lies in three interconnected effects.

First, it improves **diagnostic speed**. When platform artefacts are linked semantically, operators can move from a symptom to its likely upstream causes without reconstructing dependencies manually.

Second, it improves **development efficiency**. Reuse becomes discoverable rather than accidental. Existing data, APIs, and services can be found and integrated more easily, reducing duplicated engineering effort.

Third, it improves **governance transparency**. The platform can quantify concentration of use, cross-organizational sharing, underused assets, and dependency hotspots. These are governance metrics, not just technical ones.

Implications for Smart-City Control Rooms and Digital Twins

The framework is especially relevant for control-room environments and digital-twin-oriented systems. What-if analysis, simulations, routing adjustments, and dynamic public communications all require confidence in the lineage of the data and processes involved. If the semantic structure of the platform is opaque, decision support becomes harder to trust.

By contrast, when data, applications, and dashboards are semantically connected, digital-twin functions become more governable. Operators can inspect not only city data, but the application stack that transforms those data into forecasts, alerts, and user-facing views. This strengthens the reliability of digitally mediated urban decisions.

Multi-Organizational Collaboration as a First-Class Design Constraint

A central practical lesson from the deployment is that multitenancy is not a secondary administrative detail. It is a foundational design condition. The platform simultaneously supports multiple organizations, different user roles, and shared resources across cities and use cases. In such settings, the ability to represent organization-level relations, ownership, and user groups in the same semantic model as data and processes is essential.

This is a major reason why the framework is relevant to urban development rather than only to software engineering. Smart-city platforms increasingly mediate collaboration across public agencies, regional operators, and external stakeholders. Governance therefore requires models that are simultaneously technical and organizational.

Limitations

Several limitations should be stated explicitly.

First, the framework is designed for platform governance and inspection rather than automated prediction. It improves traceability and monitoring, but it does not by itself prioritize interventions or forecast failure probabilities. That next step would require additional operational modeling.

Second, the present work focuses on semantic structure and platform observability. It does not attempt to replace runtime performance engineering, cybersecurity controls, or contractual service management. These remain essential companion layers.

Third, although the deployment is large and real, it is rooted in the SNAP4CITY ecosystem. The model is conceptually portable, but implementation details will depend on the surrounding platform architecture.

Future Directions

A clear next direction is the semantic organization of artificial-intelligence processes, including training and execution workflows under an MLOps paradigm. As urban platforms incorporate more advanced analytics and increasingly sophisticated digital twins, the semantic governance layer should expand accordingly. This includes richer representation of model lineage, retraining pipelines, experiment tracking, and higher-order simulation assets.

CONCLUSION

This paper has presented a unified semantic framework for governing multitenant smart-city platforms. The central contribution is a knowledge model that explicitly represents the relationships among data, processing components, dashboards, users, and organizations, making a large urban digital infrastructure traceable, inspectable, and reusable.

The results show that this approach is not merely conceptual. It is implemented in a production-scale smart-city platform operating on 48 virtual machines and supporting 20 organizations, about 7,500 users, about 2,500 developers, hundreds of processing components, and hundreds of thousands of data sources. Through linked-data publication, graph browsing, Data Inspector navigation, and SPARQL querying, the framework enables practical platform governance at scale. It supports problem diagnosis, development reuse, resource monitoring, and quantitative assessment of cross-application and cross-organizational coupling.

From an urban-development perspective, the key implication is straightforward: as cities increasingly rely on shared digital infrastructures, the governability of those infrastructures becomes a strategic capability. A smart city platform must do more than ingest and visualize data; it must also reveal how its own applications are composed and how its resources are shared. The unified knowledge model provides that visibility and, in doing so, strengthens the reliability, maintainability, and scalability of smart-city operations.

REFERENCES

[1] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter Cities and Their Innovation Challenges," *Computer*, vol. 44, no. 6, pp. 32–39, 2011.

[2] C. Badii, E. G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, and S. Valtolina, "Snap4City: A Scalable IoT/IoE Platform for Developing Smart City Applications," in *Proceedings of the IEEE Smart City Innovation Conference*, 2018.

[3] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, and M. Paolucci, "Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs," *Future Generation Computer Systems*, vol. 75, pp. 14–29, 2017.

[4] P. Bellini, M. Benigni, R. Billero, P. Nesi, and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-City Service," *International Journal of Visual Languages and Computing*, vol. 25, pp. 827–839, 2014.

[5] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A Standard-Based Open Source IoT Platform: FIWARE," *IEEE Internet of Things Magazine*, vol. 2, no. 3, pp. 12–18, 2019.

[6] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, and S. Neumaier, "Knowledge Graphs," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37, 2021.

[7] C. Diamantini, A. Mircoli, D. Potena, and E. Storti, "Process-Aware IIoT Knowledge Graph: A Semantic Model for Industrial IoT Integration and Analytics," *Future Generation Computer Systems*, vol. 139, pp. 224–238, 2023.

[8] T. Ruohomäki, E. Airaksinen, P. Huuska, O. Kesäniemi, M. Martikka, and J. Suomisto, "Smart City Platform Enabling Digital Twin," in *Proceedings of the International Conference on Intelligent Systems*, 2018.

[9] M. Fanfani, L. A. I. Palesi, and P. Nesi, "Microservices' Libraries Enabling Server-Side Business Logic Visual Programming for Digital Twins," *SoftwareX*, vol. 27, p. 101805, 2024.

[10] P. Bellini, M. Fanfani, P. Nesi, and G. Pantaleo, "Snap4City Dashboard Manager: A Tool for Creating and Distributing Complex and Interactive Dashboards with No or Low Coding," *SoftwareX*, vol. 26, p. 101729, 2024.

[11] S. Bilotta, L. A. I. Palesi, and P. Nesi, "Predicting Free Parking Slots via Deep Learning in Short-Mid Terms Explaining Temporal Impact of Features," *IEEE Access*, vol. 11, pp. 101678–101693, 2023.

Arza Churchman, Professor Emerita, Faculty of Architecture and Town Planning, Technion-Israel Institute of Technology

Manuscript Published; 15 October 2025.