BSGLg, 85, 2025, 1-23 *Accepté le ???*

TOWARDS FLEXIBLE AND LIGHTWEIGHT SOLUTIONS FOR URBAN DIGITAL TWINS

VERS DES SOLUTIONS FLEXIBLES ET LÉGÈRES POUR LES JUMEAUX NUMÉRIQUES URBAINS

Imane JEDDOUB, Jean-Paul KASPRZYK, Roland BILLEN

Abstract

Urban Digital Twins (UDTs) are increasingly used to support analysis, planning and decision-making in complex urban environments. However, many existing solutions rely on rigid and monolithic architectures that limit their capacity to evolve with changing data sources, applications and technologies. This paper presents the Urban Digital Twin research activities conducted within the GeoScITY laboratory, focusing on architectural principles and tools that promote flexibility, accessibility and long-term maintainability. These contributions rely on semantic 3D city models as a stable spatial backbone and combine lightweight encodings, semantic validation middleware and automated workflows to integrate static and dynamic urban data. The approach is illustrated through several tools developed at GeoScITY, including City2Twin for managing and visualizing evolving 3D city models, automated workflows for data integration, CERBERE for schema mediation and semantic validation, and GIS4IoRT for integrating Internet of Things and robotic systems into geospatial digital twins. Together, these contributions demonstrate how Urban Digital Twins can be designed as evolving, interoperable and reusable systems rather than fixed digital replicas.

Keywords

urban digital twin, CityGML, CityJSON, integration, IoT, simulation, data management

Résumé

Les jumeaux numériques urbains (JNU) sont de plus en plus mobilisés pour l'analyse, la planification et l'aide à la décision dans des environnements urbains complexes. Toutefois, de nombreuses solutions existantes reposent sur des architectures rigides et monolithiques, limitant leur capacité à évoluer face à la diversité des données, des usages et des technologies. Cet article présente les travaux de recherche menés au sein du laboratoire GeoScITY autour des JNU, en mettant l'accent sur des principes architecturaux favorisant la flexibilité, l'accessibilité et la maintenabilité à long terme. Ces contributions s'appuient sur les modèles de villes 3D sémantiques comme socle spatial stable et combinent des encodages légers, des mécanismes de validation sémantique et des chaînes de traitement automatisées pour intégrer des données urbaines statiques et dynamiques. L'approche est illustrée à travers plusieurs outils développés au sein de GeoScITY, notamment City2Twin pour la gestion et la visualisation de modèles 3D évolutifs, des chaînes de traitements automatisées pour l'intégration des données, CERBERE pour la médiation de schémas et la validation sémantique, et GIS4IoRT pour l'intégration des objets connectés et robotiques dans les jumeaux numériques géospatiaux. Ces travaux montrent comment les JNU peuvent être conçus comme des systèmes évolutifs, interopérables et réutilisables, plutôt que comme des répliques numériques figées.

Mots-clés

jumeau numérique urbain, CityGML, CityJSON, intégration, internet des objets, simulation, gestion de données

INTRODUCTION

Urban governance increasingly relies on digital infrastructures to monitor, analyze, and predict complex interactions between physical, environmental, and socio-technical systems. Yet producing and managing urban data remain divergent and fragmented policies addressing building energy retrofitting, air-quality regulation, and traffic management require the integration of static spatial representations, real-time sensor observations, regulatory constraints, and simulation outputs, while in practice such datasets are distributed across departments, encoded in heterogeneous formats, and updated at different temporal and spatial scales. The smart city paradigm, which emerged over the past decade to promote data-driven urban governance through sensor networks, open data, and digital service platforms, has largely remained a strategic and policy-oriented vision rather than a well-articulated technical architecture. Consequently, many smart city initiatives remain dashboard-oriented, privileging monitoring over semantic integration, and provide limited specification of the computational infrastructures necessary to support integrated modelling, scenario evaluation, and long-term decision-making. In this sense, the smart city vision raises expectations of coherent and integrated digital infrastructures yet offers limited direction on how heterogeneous data ecosystems should be structured and maintained in practice.

Urban Digital Twins (UDTs) have emerged within this context as an attempt to move from strategic vision toward operationalization. Building upon earlier developments in 3D city modelling and City Information Modelling (CIM), UDTs are described as digital representations of urban environments that combine semantic spatial models, dynamic data streams, and simulation capabilities within a shared analytical framework. From approximately 2016 onward, the notion of the "digital twin" gained interdisciplinary traction and began to function as an umbrella concept, bringing together ideas from Geographic Information Systems (GIS), Building Information Modelling (BIM), Internet of Things (IoT), and Spatial Data Infrastructures (SDIs) (Stoter *et al.*, 2021). Yet definitions vary across disciplines, and implementations differ widely in terms of scale, scope, and degree of bidirectional coupling between physical and digital systems. In many cases, the term "digital twin" is applied to platforms that incorporate enhanced visualization environments or domain-specific analytical tools rather than integrated and continuously synchronized urban systems. UDT solutions range from generic geospatial representations capturing a snapshot in time to dynamic and domain-specific systems coupled with sensors, simulations, and operational workflows, with many existing initiatives relying on heavy, monolithic platforms that can lock cities into specific technologies, limit transparency, and hinder reuse, auditing, and adaptation to new data sources or policy questions. This conceptual ambiguity reflects a broader discrepancy between the integrative promises of the digital twin metaphor and the realities of urban data infrastructures, shaped by organizational boundaries, legacy systems, and uneven data availability and quality.

This paper addresses these questions by analyzing the contemporary landscape of UDT concept and implementations to identify key architectural and data-related requirements for sustainable UDT development. Rather than treating UDTs as technological products or isolated tools, we approach them as socio-technical infrastructures whose viability depends on how heterogeneous datasets, standards, and organizational arrangements are articulated. This analysis is grounded in applied research conducted within the GeoScITY laboratory across several urban projects, which provided empirical grounding for the design principles presented here and revealed recurring challenges in data integration, interoperability, and long-term system maintenance. Based on this analytical and experimental foundation, we propose a design approach that emphasizes modular data integration, semantic validation, web-native interoperability, and incremental system evolution, operationalized through a set of interoperable tools and workflows including City2Twin, CERBERE, automated ETL pipelines, and GIS4IoRT. The contribution of this paper is therefore twofold: first, the identification and structuring of functional and non-functional requirements derived from a critical analysis of existing UDT practices; and second, the demonstration of how these requirements can be operationalized in practice.

The remainder of this paper is structured as follows. Section I reviews key conceptual perspectives on Urban Digital Twins and introduces the main design principles adopted in this work, with particular emphasis on data integration and interoperability. Section II derives the functional and non-functional requirements for flexible, lightweight, and maintainable UDT architectures. Section III evaluates existing standards and data models in relation to these requirements, positioning current building blocks and highlighting where they support or limit UDT design objectives. Section IV presents the GeoScITY framework and its concrete implementations, illustrating how the identified requirements can be operationalized in practice. The paper concludes with reflections on long-term maintainability and accessibility, and on the role of modular, system-of-systems approaches in the evolution of Urban Digital Twins.

I. CONCEPTUAL FOUNDATIONS OF URBAN DIGITAL TWINS

A. UDT Definition and Maturity Levels

Urban digital twins (UDTs) can be understood as spatial temporal virtual representations of a real-world city or district that need to be continuously synchronized with their physical counterpart at an appropriate rate of synchronization, and are used to understand, simulate and manage urban systems. Recent studies show that the term "Urban Digital Twins" or "Digital Twins for Cities" are still used inconsistently and often overlaps with earlier paradigms such as GIS-based city models, city information models or spatial data infrastructures. (Jeddoub *et al.*, 2023) show that many current "UDT" initiatives in fact behave more like digital shadows, with limited or no bidirectional coupling or automation, and argue that a full UDT should include four core capabilities, namely, spatiotemporal integration of heterogeneous data, dynamic updating, simulation and prediction, and interaction with various stakeholders. (Kritzinger *et al.*, 2018) distinguishes between digital 'model', shadow' and 'twin', which have been developed in the context of manufacturing industries. In this framework, a digital model refers to a digital representation of a physical entity with no automatic data exchange between the two; a digital shadow denotes a system in which data flows automatically from the physical to the digital counterpart, but not in the reverse direction; and a digital twin implies fully bidirectional and automated data exchange between the physical and digital entities. This typology was extended to the city and urban context to introduce a maturity level according to the level of coupling between the physical and virtual counterparts. Figure 1 synthesizes these maturity distinctions by illustrating the progression from digital models to digital shadows, and finally to fully coupled digital twins. The levels differ in terms of data integration, synchronization frequency, and degree of bidirectional interaction between the physical and virtual systems. This clarification helps distinguish conceptual aspirations from current operational realities.

Following this line of work, Abdelrahman *et al.* (2025) employ large-scale Natural Language Processing (NLP) and statistical analysis on a corpus of 15,000 full-text articles to investigate how digital twins are described across domains in the built environment. Their component analysis reveals two distinct clusters: a group of high-performance, real-time (HPRT) components, including cloud computing, high-performance computing, security protocols, real-time data, simulation models and AI/ML models, and a group of long-term decision-support (LTDS) components, including data representation, data validation, visualization and policy-related components. While the HPRT cluster is characteristic of computationally intensive applications such as manufacturing and autonomous vehicles and is rarely observed in building and urban contexts, the LTDS cluster dominates in building, architectural and urban digital twins, where 2D/3D data, semantic data representation, validation and visualization are common compared to real-time data, AI/ML and advanced simulation models. On this basis, (Abdelrahman *et al.*, 2025) conclude that, in their current state of practice, digital twins in the built environment are better characterized as LTDS systems operating over a wide range of temporal scales rather than real-time systems; at the same time, they show that components such as real-time data streaming, AI/ML and high-performance computing are gaining prominence over time, suggesting that future, evolving definitions of digital twins may incorporate these elements as integral rather than optional parts of the concept.

B. Architectural and Governance Perspectives on UDTs

Although consensus on a universal definition remains one of the main challenges, UDTs are already operating in practice and the term is increasingly used as a convergence of urban and geospatial concepts such as GIS, GeoBIM, smart cities, spatial data infrastructures and IoT, which share the objectives of representing the current state of the city, integrating data from different domains and supporting multiple decision-making processes. This broad use of the term raises the risk that long-standing issues such as data silos, fragmented governance and limited interoperability are simply reframed under a new label rather than being concretely addressed, and that many solutions remain limited to isolated projects rather than evolving into open, reusable urban platforms (Lei *et al.*, 2023). In parallel, another challenge that needs to be highlighted is the motivation towards a digital twin as an exact digital mirror of the city;

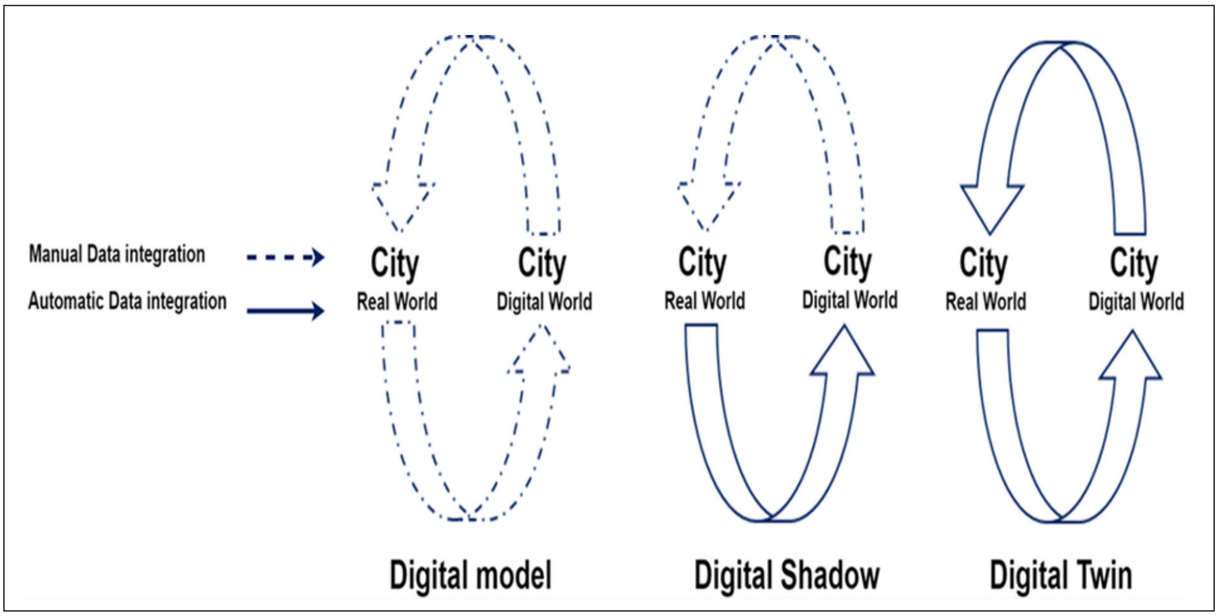

Figure 1. Maturity levels of the urban digital twin concept (Jeddoub *et al*., 2023)

while this may be appropriate for a single entity (for instance, a product in manufacturing), it is difficult to replicate and maintain at city scale, where abstraction and generalization are necessary and different applications require different views of the same underlying reality (Batty, 2025, 2018). Building on this, (Coenen *et al.*, 2025) conceptualize Digital twins (DTs) as data-driven decision-support systems for public administrations rather than as a single, monolithic technology stack. They are framed as central urban integrators that connect data and models from various policy domains with different purposes (e.g. mobility, climate, planning) to support evidence-based decision-making and cross-domain collaboration. Figure 2 illustrates this multi-scale configuration as a system of systems or a twin of twins, where digital twins at different spatial and functional levels are interconnected rather than hierarchically absorbed into a single monolithic system (Lu *et al.*, 2019). The exchanges between these levels concern different categories of data rather than fixed datasets. For example, lower-level twins (e.g., building or infrastructure DTs) may provide geometric representations, operational states, sensor observations, and performance indicators to higher-level city or national twins. Conversely, higher-level twins may transmit aggregated indicators, regulatory parameters, scenario definitions, or planning constraints downward. The exact nature of these exchanges depends on the application context, governance arrangements, and data availability. Therefore, Figure 2 should be interpreted as a structural representation of possible data interconnection rather than as a prescriptive or standardized data flow model.

Using a system-of-systems perspective, they argue that a DT consists of several heterogeneous but independently operable systems that are networked together for a common goal, with the overall functionality emerging from their interactions rather than from any single component. Five types of systems jointly deliver the core meta-features of a DT: an IoT system for sensing and data acquisition, visualization systems for interaction with users, a data-space system for structuring and sharing data, an algorithm and data management system for analytics and model handling, and a scenario-building system for exploring policy options and their impacts. These systems are explicitly linked to meta-requirements derived from the policy cycle of local administrations, such as describing the physical twin in (near-)real time, predicting the effects of interventions and supporting cross-domain assessment. As a result, they conclude that an Local DT should not be understood as a standard software product serving a single objective, but as a configurable concept whose architecture and components evolve with the decision-support needs of the locality it serves, and whose main added value lies in breaking down data silos through the coordinated interplay of multiple underlying systems. This perspective also raises fundamental questions about architecture and governance: whether UDTs should be designed as centralized systems offering a single, authoritative view of the city, or instead as decen-

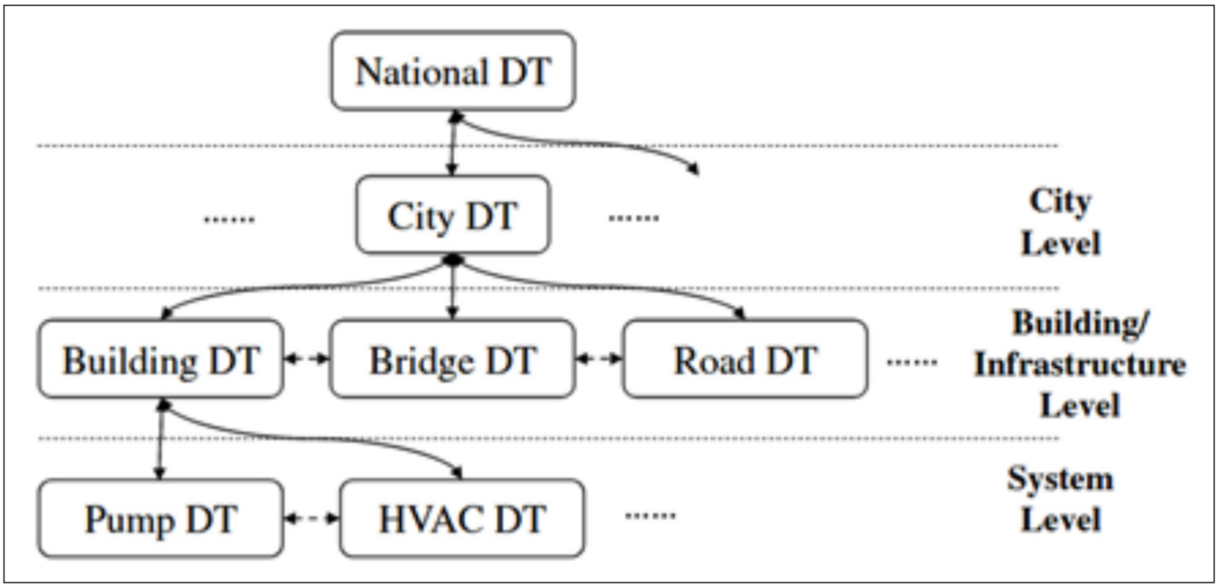

Figure 2. Urban digital twins conceptualized as a connected system of systems operating across multiple spatial scales and infrastructure components (Lu *et al.*, 2019)

tralized, federated approaches of domain-specific twins that interoperate through shared standards, reference models and institutional arrangements.

Beyond technical architecture, some authors have proposed analytical and governance-oriented perspectives that analyze how UDTs should be structured from organizational and institutional point of view. These conceptual contributions frame UDTs as federated, multi-view systems that are tailored to specific uses but remain connected through shared data models, interoperability standards, and governance arrangements (Yan *et al.*, 2025). These perspectives therefore move the debate beyond definitions and technical components, and focus on how UDTs are organized, governed, and aligned with institutional decision-making processes.

C. UDT applications and use cases

Despite these conceptual frameworks, which remain largely analytical and not yet systematically tested in practice, empirical work on existing initiatives shows that UDTs are already being deployed across a wide range of urban applications. (Alva *et al.*, 2022) identified district and city-scale UDTs used for forecasting, emergency planning, operational optimization, participatory planning, policy development and scenario modelling. This growing body of use highlights that, even without a common definition, UDTs initiatives are increasingly implemented as integrated platforms for data processing, urban analysis, design, simulation and modelling. They aim to connect the different stakeholders and actors involved in urban development and management, and to support the smart city vision of a sustainable, inclusive, and participatory city. In practice, they are built on open standards and on multi-scale, multi-temporal databases that integrate a wide variety of data sources, representing a broad range of urban features, systems and processes.

These implementations aim to coordinate data and models across multiple stakeholders and to provide shared analytical environments rather than isolated domain-specific tools. From technical perspectives, they rely on multi-scale spatial databases and interoperable standards to combine heterogeneous datasets, including 3D city models, sensor observations, simulation outputs, and thematic layers. However, the degree of actual integration and interoperability varies significantly across initiatives, and many platforms remain domain- or project-specific.

D. Data Integration and Interoperability as Core UDT Design Principles

Several key concepts underpin the analysis of Urban Digital Twin (UDT) architectures and are often used interchangeably in the literature. Clarifying these terms is necessary to structure the discussion.

- interoperability refers to the ability of datasets, systems, or components to exchange information and to interpret and reuse it consistently across platforms. It extends beyond technical compatibility and includes semantic, organizational, and legal dimensions, ensuring that exchanged information retains its meaning and operational validity across contexts (Acharya *et al.*, 2024);

- data integration, by contrast, refers to the process of combining heterogeneous datasets into a coherent and unified representation aligned with specific application requirements. While interoperability enables exchange and interpretation, integration concerns harmonization and consolidation (Noardo, 2022);
- discoverability denotes the ability of datasets, services, or components to be identified, located, and accessed through standardized metadata, catalogues, or APIs. In UDT contexts, discoverability is essential to enable reuse and cross-domain integration;
- openness refers to the use of non-proprietary standards, transparent data models, and accessible interfaces that allow independent actors to inspect, reuse, and extend system components;
- flexibility describes the capacity of a UDT architecture to accommodate new datasets, services, or analytical models over time without requiring substantial redesign. This typically relies on modular structures and loosely coupled interfaces;
- robustness denotes the ability of a UDT to operate reliably under imperfect conditions, such as incomplete data, heterogeneous update frequencies, or evolving schemas, while maintaining traceability and consistency.

These concepts are interrelated but not equivalent. Interoperability enables integration; openness facilitates interoperability; discoverability supports reuse; and flexibility and robustness characterize architectural performance over time.

In UDT implementations, interoperability operates across multiple interconnected levels, commonly distinguished as technical, semantic, organizational, and legal interoperability. These levels extend beyond ICT systems and data formats to include institutional practices, governance structures, and regulatory frameworks.

Integration in UDTs should not be understood as a single operation but as a layered process occurring at different architectural levels (Jeddoub *et al.*, 2024, Yan *et al.*, 2025). It can be addressed:

- at the conceptual data model, through shared data models, ontologies, and semantic mappings that ensure conceptual alignment;
- at the database level, through schema alignment, identifier reconciliation, coordinate reference and temporal harmonization, and format transformations;
- at the front-end level, through interoperable APIs, simulation models, and processing workflows allow datasets and analytical components to interact within operational platforms.

Importantly, interoperability in UDTs concerns not only datasets but also system components such as simulation engines, analytics modules, visualization tools, and IoT services. Ensuring that these components can exchange inputs, outputs, and parameters in standardized ways is essential for cross-domain scenario modelling and decision support.

In practice, UDT implementations face persistent barriers at both integration and interoperability levels, including heterogeneous sources, mismatched coordinate systems and levels of detail, semantic inconsistencies, limited metadata, and incompatible processing environments. When addressed through standardization, semantic alignment, modular architecture, and API-based infrastructures, these mechanisms enable scalable and evolutive UDT implementations capable of cross-domain reuse.

In this work, we use these complementary design principles to address the challenges of data integration and interoperability in UDT implementations. Rather than treating them as isolated requirements, we consider them as a basis for evaluating how UDT architectures support the integration, exchange, reuse, and long-term evolution of heterogeneous urban data and services. Building on lessons learned from several UDT implementations in which we were involved, the next section translates this perspective into a set of functional and non-functional requirements for flexible and lightweight UDT design.

II. REQUIREMENTS FOR FLEXIBLE AND LIGHTWEIGHT UDT

Drawing on our involvement in several UDT implementations, as well as an in-depth analysis of a range of UDT projects, we propose a framework to assess technologies and architectural choices for flexible and lightweight UDTs. The requirements are organized into two complementary categories: functional requirements, which specify the capabilities the UDT must provide, and non-functional requirements, which guide architectural decisions and long-term operation.

Functional requirements (What capabilities must a flexible and lightweight UDT provide to integrate, update, and serve urban data for multiple applications?)

- federated Integration of Heterogeneous Data: A UDT must support the integration of heterogeneous datasets, including 3D city models, BIM models, infrastructure networks, IoT streams, and simulation outputs. Because these datasets are often distributed across organizations and updated independently, integration should follow a federated approach based on shared identifiers, metadata, and spatial or semantic relationships;
- multi-Scale and Multi-Application Support: A UDT must support multiple spatial scales and application contexts through appropriate Levels of Detail (LoDs), application-specific data delivery, and reusable service interfaces;
- dynamic and Streaming Data Integration: Beyond static spatial representations, a UDT must ingest, synchronize, and serve time-varying data from sensors, IoT devices, and mobile systems through temporal querying and incremental updates, without requiring full regeneration of the 3D city model;
- structured Semantic Integration: Consistent cross-domain analysis requires structured semantic alignment. The UDT must rely on shared conceptual models, schema mappings, or controlled vocabularies that allow datasets and components to be interpreted coherently. In this paper, semantic integration refers primarily to alignment at the conceptual schema level (e.g., CityGML classes, IFC mappings), rather than to full Linked Data implementations based on RDF/OWL, although such approaches may complement the architecture;
- simulation Coupling and (Re) Integration into the UDT: A UDT must support the coupling of simulation processes and the reintegration of simulation outputs into the same spatial and semantic framework used for other urban data. Whether simulations are executed within the UDT environment or externally, the architecture should ensure traceability and consistency.

Non-functional requirements (What requirements need to be meet for the UDT to remain usable, scalable, maintainable, and secure in long-term operation?)

- accessibility and developer usability: The UDT is required to expose data and functionality through web APIs and standard query mechanisms, using developer-friendly formats (e.g., JSON-based encodings such as CityJSON where appropriate) and documentation that enables rapid integration by diverse clients, including modern web browsers;
- maintainability and evolvability: The UDT should be modular and loosely coupled with clear separation between storage, API layers, and clients; components (e.g., ingestion, storage, processing, and visualization services) need to be independently upgradable and reusable to support evolving requirements and emerging technologies;
- performance and scalability: The UDT should maintain acceptable response times for interactive and analytical use while scaling to city-level data volumes and high-velocity data streams; this typically requires scalable strategies such as caching, 3D tiling/levels of detail (LOD), and asynchronous processing for computationally intensive tasks;
- robustness and reliability: The UDT needs to handle incomplete datasets, irregular update cycles, and sensor outages through monitoring and explicit provenance/quality indicators that preserve user trust in outputs;
- security, privacy, and governance support: The UDT must secure access, provide auditing, and policy-driven data sharing; governance mechanisms should be adaptable to changing urban needs and emerging technologies, and may include participatory structures (e.g., advisory boards and citizen representation) to align DT development with public values;
- cost and operational feasibility: The UDT should minimize integration and operational overhead (e.g., by avoiding monolithic dependencies and supporting standards-based interoperability) to improve long-term sustainability under public-sector budget and skills constraints.

These requirements form the framework used in the following section to assess existing data models, standards, and architectural approaches in relation to flexible and lightweight UDT design.

III. POSITIONING EXISTING STANDARDS AND MODELS WITHIN UDT REQUIREMENTS

Building on the requirements introduced above, this section evaluates the principal data models and interoperability standards commonly leveraged in Urban Digital Twins (UDTs). The objective is not to provide an exhaustive list of standards, but to examine how existing standards contribute to or limit flexibility, lightweight implementation, semantic integration, dynamic data handling, and simulation coupling within UDT architectures.

In practice, UDTs are implemented as interoperable ecosystems that combine static 3D representations of the built environment with dynamic observations (e.g., sensor data) and expose these resources to multiple applications. Achieving such integration at city scale requires shared standards for data modelling, exchange, discoverability, and web-based access, since UDT components are typically distributed across organizations, software platforms, and update cycles. The Open Geospatial Consortium (OGC) provides a mature portfolio of standards spanning both spatial data infrastructure web services (e.g., cataloguing, feature and map delivery, processing) and the latest OGC API family, which adopts lightweight, modular, web-native interfaces aligned with API-first architecture. In parallel, domain-specific specifications such as CityGML and CityJSON for semantic 3D city models, BIM/IFC for detailed building information, SensorThings for observations, Moving Features for trajectories, and GeoPose for orientation enable structured representation and linking of urban entities across domains. These standards act as complementary building blocks within UDT ecosystems; however, their ability to support flexible and lightweight architecture depends on how they are combined, profiled, and implemented. The following subsections therefore analyze the most relevant standards in terms of their alignment with the functional and non-functional requirements identified in Section II, focusing on 3D city models, BIM integration, IoT and streaming data, and simulation coupling.

A. Semantic 3D City Models (CityGML and CityJSON)

Semantic 3D city models play a central role in fulfilling the requirement for structured semantic integration and multi-scale support. Unlike purely graphical or mesh-based models, semantic city models represent urban entities as object-based structures in which geometry, attributes, and relationships are explicitly defined. This enables consistent referencing of urban objects such as buildings, roads, or city furniture, allowing additional information including sensor observations, simulation outputs, and thematic attributes to be attached to the same spatial entities.

Among existing standards, CityGML is widely recognized as the most established open standard for semantic 3D city modelling. It defines a comprehensive conceptual model and an XML/GML-based encoding for representing virtual 3D city and landscape models. The model is organized into thematic modules and supports multiple Levels of Detail (LoD), allowing the same urban object to be represented at different spatial resolutions while maintaining consistent semantics. The latest version, CityGML 3.0, further extends the modelling framework by revising the LoD concept, introducing improved support for time-dependent properties, and enabling more flexible domain extensions (Kutzner *et al.*, 2020). These characteristics make CityGML particularly suitable for UDT contexts where a stable spatial reference is required to integrate heterogeneous datasets and analytical outputs, including simulation results such as energy demand, noise levels, or environmental indicators.

Despite these advantages, the XML/GML encoding traditionally used in CityGML can become verbose and computationally heavy, particularly for web-based applications and large-scale city models. This limitation has motivated the development of alternative encodings designed to support more lightweight and developer-friendly implementations.

CityJSON provides such an alternative by offering a JSON-based encoding aligned with the CityGML conceptual model. While preserving the core concepts of city objects, semantics, and levels of detail, CityJSON represents the data in a compact JSON structure that is easier to parse, manipulate, and exchange in modern web environments. This makes CityJSON particularly attractive for Urban Digital Twin implementations relying on web APIs, browser-based visualization, and microservice architectures. However, CityJSON currently implements a subset of the full CityGML conceptual model and may require extensions or complementary mechanisms to support more advanced modelling capabilities.

Beyond data models and encodings, database implementations play an important role in operational UDT architectures, particularly when dealing with large datasets and frequent updates. File-based representations such as CityGML or CityJSON are often complemented by relational or document-oriented database systems that support efficient querying, indexing, and integration with external services.

One widely used solution is 3DCityDB, which maps the CityGML conceptual schema to a relational database structure in systems such as PostgreSQL/PostGIS or Oracle. This approach enables efficient storage, querying, and management of large semantic 3D city models while maintaining consistency with the CityGML data model. Database-based implementations are particularly relevant for UDTs because they support frequent updates, integration with analytical workflows, and interaction with external services.

More recently, lightweight alternatives such as the CityJSON Database (CJDB) have been proposed. CJDB stores CityJSON objects directly within PostgreSQL/PostGIS using JSON-based structures (e.g., JSONB columns), providing a more compact and flexible schema compared to fully relational CityGML implementations. This design reduces schema complexity and facilitates integration with web-based applications. However, it also introduces trade-offs. For example, storing geometries within JSON structures may complicate advanced spatial queries, and schema constraints cannot be enforced as strictly as in relational representations of the CityGML model.

These differences illustrate a broader architectural trade-off within UDT design (i.e., rich semantic models such as CityGML offer strong conceptual consistency and analytical capabilities, whereas lightweight encodings and database schemas such as CityJSON and CJDB improve usability and web interoperability). In practice, many UDT implementations combine these approaches by using CityGML as a comprehensive semantic reference model while adopting CityJSON or database-based implementations for operational workflows.

Overall, semantic 3D city models provide a stable spatial backbone that supports the integration of heterogeneous urban datasets, simulation outputs, and sensor observations. Their ability to represent urban objects consistently across scales makes them a fundamental building block of Urban Digital Twins, even though additional mechanisms are required to support dynamic data streams, real-time interactions, and cross-domain interoperability.

B. BIM and IFC for Multi-Scale Integration

Beyond semantic 3D city models, Building Information Modelling (BIM) provides another important source of 3D information for UDTs. BIM models describe buildings and infrastructures as collections of parametric, volume-based objects representing physical components such as walls, slabs, pipes, or equipment. This object-centric modelling paradigm allows stakeholders in the Architecture, Engineering, and Construction (AEC) domains to work within a shared digital representation during design, construction, and operation phases, supporting the full lifecycle management of built assets.

BIM models are typically exchanged using the Industry Foundation Classes (IFC) standard, an open data model maintained by buildingSMART. IFC describes detailed geometry, construction components, materials, and technical systems, often including internal building structures and engineering details that are not represented in city-scale models. As a result, BIM provides a much higher level of geometric and semantic detail than semantic 3D city models such as CityGML or CityJSON. Within UDT architecture, BIM is therefore primarily used at the building or asset scale, complementing the broader, city-scale coverage provided by semantic 3D city models. Many UDT implementations consequently adopt multi-scale data architecture, in which city models provide the spatial backbone while IFC models supply detailed information for selected assets or infrastructures.

Despite these complementarities, BIM-GIS integration remains technically challenging. Conceptually, BIM and geospatial models follow different modelling paradigms. IFC models often rely on constructive solid geometry and local coordinate systems optimized for design workflows, whereas geospatial models typically use boundary representations referenced in global coordinate systems. This difference leads to both geometric mismatches, when converting volumetric BIM geometry into valid geospatial surfaces, and semantic mismatches, since IFC typically defines more detailed building classes and component hierarchies than city models. Consequently, most BIM-GIS conversion workflows require explicit mapping rules and generalization strategies to determine which attributes, objects, and geometric details should be preserved or simplified.

In addition, the form of BIM-GIS integration strongly depends on the intended application. Some workflows focus on one-way transformations, such as generating 3D city models from IFC files, while others rely on loose coupling approaches, where BIM elements are linked to city objects through identifiers or external references. Recent GeoBIM research initiatives have highlighted that these integration strategies often produce heterogeneous results, partly due to the lack of standardized requirements for specific urban workflows such as digital building permitting or construction monitoring. Robust georeferencing of BIM models has also been identified as a critical prerequisite before semantic or geometric interoperability can be achieved.

Beyond modelling challenges, integration also raises issues related to data management and performance. BIM models are often large and complex, and their native viewers and workflows are optimized for design coordination rather than geospatial analysis. As a result, BIM datasets typically require transformation, simplification, or restructuring before they can be integrated into web-based UDT platforms. This includes resolving local coordinate systems, aligning building models with national mapping frameworks, and reducing geometric complexity to support interactive visualization.

For these reasons, BIM is generally not treated as a standalone UDT platform but rather as a complementary component within a broader data integration architecture. In most UDT implementations, semantic 3D city models provide the geospatial backbone for urban-scale analysis, while IFC models contribute detailed building-level information when required for specific applications such as digital permitting, facility management, or energy simulation. Ongoing GeoBIM initiatives aim to further standardize these integration workflows and reduce interoperability barriers, reinforcing the role of BIM as an important but partial building block in UDT ecosystems.

C. IoT Standards and Context Management

In addition to static spatial representations, UDTs must integrate dynamic observations from sensors and connected devices in order to maintain a continuous link between the physical city and its digital representation. These data streams include measurements related to mobility, environmental conditions, energy consumption, or infrastructure monitoring, often collected at high temporal frequencies. Integrating such dynamic information is necessary for supporting real-time monitoring, predictive modelling, and scenario analysis within UDT platforms.

To support these requirements, several interoperability standards have emerged for managing sensor data and contextual information. From a geospatial perspective, the OGC SensorThings API provides a widely adopted RESTful framework for publishing, querying, and managing observations and sensor metadata. The standard builds upon the Observations and Measurements (O&M) model and enables interoperable access to time-series data through web APIs. SensorThings aligns well with several UDT requirements, including streaming data integration, discoverability, and interoperable access to observations. In many implementations, SensorThings API is used as a dynamic data layer connected to semantic 3D city models, allowing observations to be associated with urban objects such as buildings, streets, or environmental monitoring stations. Concepts such as the CityGML Dynamizer further enable time-varying properties to be attached directly to city objects, linking static spatial representations with real-time measurements.

In parallel, NGSI-LD, standardized by ETSI, has gained interest in smart city ecosystems as a context-management framework. Unlike SensorThings API, which focuses primarily on observation management, NGSI-LD adopts an entity-centric model in which urban entities and their relationships are represented using linked-data principles. This approach allows contextual information about objects such as attributes, relationships, and temporal states to be managed dynamically through context brokers. As a result, NGSI-LD is widely used in smart city platforms to integrate heterogeneous datasets and support event-driven applications (Bauer *et al.*, 2021).

Despite their complementary roles, integrating IoT standards with geospatial urban models remains challenging. Sensor observations are often managed within dedicated data platforms, while semantic 3D city models operate within geospatial infrastructures. Linking these components requires consistent identifiers, spatial references, and semantic mappings between sensor entities and urban objects. Without such alignment mechanisms, IoT data streams risk remaining isolated within domain-specific platforms rather than contributing to an integrated digital twin environment.

In practice, many UDT implementations combine these standards within multi-layer architectures, where semantic 3D city models provide the spatial backbone and IoT platforms manage dynamic observations. This approach allows sensor measurements to be associated with urban objects and supports integrated workflows ranging from monitoring to simulation and decision support. However, achieving seamless interoperability across these layers requires proper schema alignment, metadata management, and governance mechanisms to ensure consistent interpretation of dynamic data across domains.

Overall, IoT and context-management standards constitute a key enabling layer for UDTs by providing mechanisms for managing time-varying data and linking digital models to real-world measurements. Their effective integration with semantic city models is therefore essential for transforming static digital representations into operational, data-driven urban platforms.

D. Simulation Coupling and Interoperable Services

Simulation capabilities are a key component of many UDTs implementations, as they allow the digital representation of the city to represent not only its current state but also potential future states under alternative policies, designs, or environmental conditions. By integrating simulation outputs into the same spatial framework used for dynamic data, UDTs can support predictive analysis and scenario exploration across domains such as energy systems, mobility, environmental processes, or infrastructure planning.

In many implementations, semantic 3D city models such as CityGML or CityJSON act as the geometric and semantic input layer for simulation workflows. These models provide the spatial structure required to define simulation inputs and to attach resulting indicators to specific urban objects. A common architectural pattern consists of exporting the city model to domain-specific simulation tools, executing simulations externally, and reintegrating the results into the UDT as attributes, indicators, or spatial layers associated with buildings, streets, or infrastructure networks.

This approach is widely used in the urban energy domain, where simulation platforms such as SimStadt rely on CityGML-based models to compute district-scale indicators including heating demand, solar potential, or energy supply scenarios. In such workflows, semantic city models are enriched with building attributes and geometric information, processed within specialized simulation engines, and the resulting indicators are then linked back to the same city objects within the digital twin environment. Similar integration patterns have been demonstrated in other domains including noise propagation, urban wind modelling, traffic simulation, and environmental risk assessment.

A critical enabling mechanism for this integration is the use of web-based APIs and interoperable service interfaces. Rather than embedding simulation engines directly inside UDT platforms, most architectures rely on loosely coupled workflows in which simulation results are exposed through interoperable services. Standards such as OGC API-Features, OGC API-Processes, and OGC API-Tiles, as well as SensorThings for time-series outputs, enable simulation indicators to be stored, queried, and visualized across distributed components of the UDT infrastructure. Through these APIs, simulation outputs can be delivered as attributes attached to city objects, as tiled raster or 3D layers, or as time-series indicators that can be explored interactively in web-based environments such as Cesium-based 3D viewers. This service-oriented architecture allows simulation workflows to remain modular while enabling results to be integrated with sensor data, geospatial datasets, and visualization interfaces.

Despite these promising developments, several limitations remain. Simulation workflows often require substantial preprocessing of city models, including geometry cleaning, level-of-detail harmonization, and parameter enrichment before they can be used as simulation inputs. Many simulation engines also operate in offline or batch-processing modes, producing results that are reintegrated into the digital twin only after computation rather than through continuous feedback loops. In addition, computational cost, model calibration requirements, and uncertainty propagation remain significant challenges when attempting to integrate simulation processes into operational UDT infrastructures.

Consequently, while current standards and modelling approaches enable the reintegration and visualization of simulation outputs within UDTs, the execution of fully synchronized simulations coupled with real-time urban data streams remains an evolving research challenge. Future UDT architectures will likely require coupling between simulation engines, urban data infrastructures, and API-based service layers to support dynamic scenario exploration and decision-support workflows.

E. Discussion

The analysis of data models and interoperability standards for Urban Digital Twins (UDTs) highlights a consistent architectural pattern: no single standard is sufficient to address the full range of functional and non-functional requirements. Instead, UDTs emerge as composite ecosystems in which multiple standards are combined to balance semantic richness, scalability, interoperability, and performance.

Semantic 3D city models such as CityGML and CityJSON provide the foundational spatial backbone of UDTs, enabling consistent representation of urban objects across multiple scales. Their strength lies in structured semantics and the ability to integrate heterogeneous datasets, including simulation outputs and sensor observations. However, a clear trade-off exists between the richness and rigor of models like CityGML and the lightweight, web-friendly characteristics of CityJSON and related database approaches, reflecting broader tensions between expressiveness and implementation efficiency.

At finer scales, BIM and IFC extend this foundation by contributing detailed building-level information. While this enhances multi-scale modelling capabilities, the integration of BIM and geospatial models remains complex due to differences in geometry, semantics, and coordinate systems. As a result, BIM is typically incorporated selectively, reinforcing the importance of hybrid architectures where different data models serve complementary roles rather than being fully unified.

The integration of dynamic data through IoT standards such as SensorThings API and NGSI-LD introduces an additional layer of complexity. These standards enable real-time data management and contextual information exchange, which are essential for transforming static models into operational digital twins. However, effective integration depends on consistent identifiers, semantic alignment, and governance mechanisms to avoid fragmentation across platforms.

Finally, simulation coupling illustrates both the potential and current limitations of UDTs. Existing standards and service-based architectures support the integration and visualization of simulation outputs, but tightly coupled, real-time simulation remains challenging due to preprocessing requirements, computational constraints, and limited synchronization with live data streams.

Overall, the discussion underscores that interoperability in UDTs is not achieved solely through standard adoption, but through careful orchestration of complementary technologies, data models, and service interfaces. Future progress will depend on improving alignment between these components, reducing integration overhead, and advancing towards more dynamic, loosely coupled, and scalable architectures capable of supporting real-time urban analytics and decision-making.

IV. PROPOSED DESIGN FOR FLEXIBLE AND LIGHTWEIGHT URBAN DIGITAL TWIN ARCHITECTURES

Building on the requirements and standards analysis presented in the previous sections, this section presents a design approach for implementing flexible and lightweight Urban Digital Twin architectures (refer to Figure 3). The architecture separates data acquisition, urban data modelling, storage and management, interoperability services, analytical processes, and application layers. The design principles of flexibility, lightweight implementation, and robustness guide the design of the core infrastructure layers through a modular system-of-systems design: flexibility is achieved via loosely coupled components and stable APIs; lightweight implementation is supported by web-native encodings and service interfaces; and robustness is ensured through explicit validation, provenance handling, and controlled integration workflows. The following subsections illustrate how these principles are implemented through complementary architectural components developed within the GeoScITY laboratory. Each subsection highlights a set of standards or models in relation to the functional and non-functional requirements identified in Section II, with particular attention to semantic integration, multi-scale support, dynamic data handling, simulation coupling, and implementation constraints.

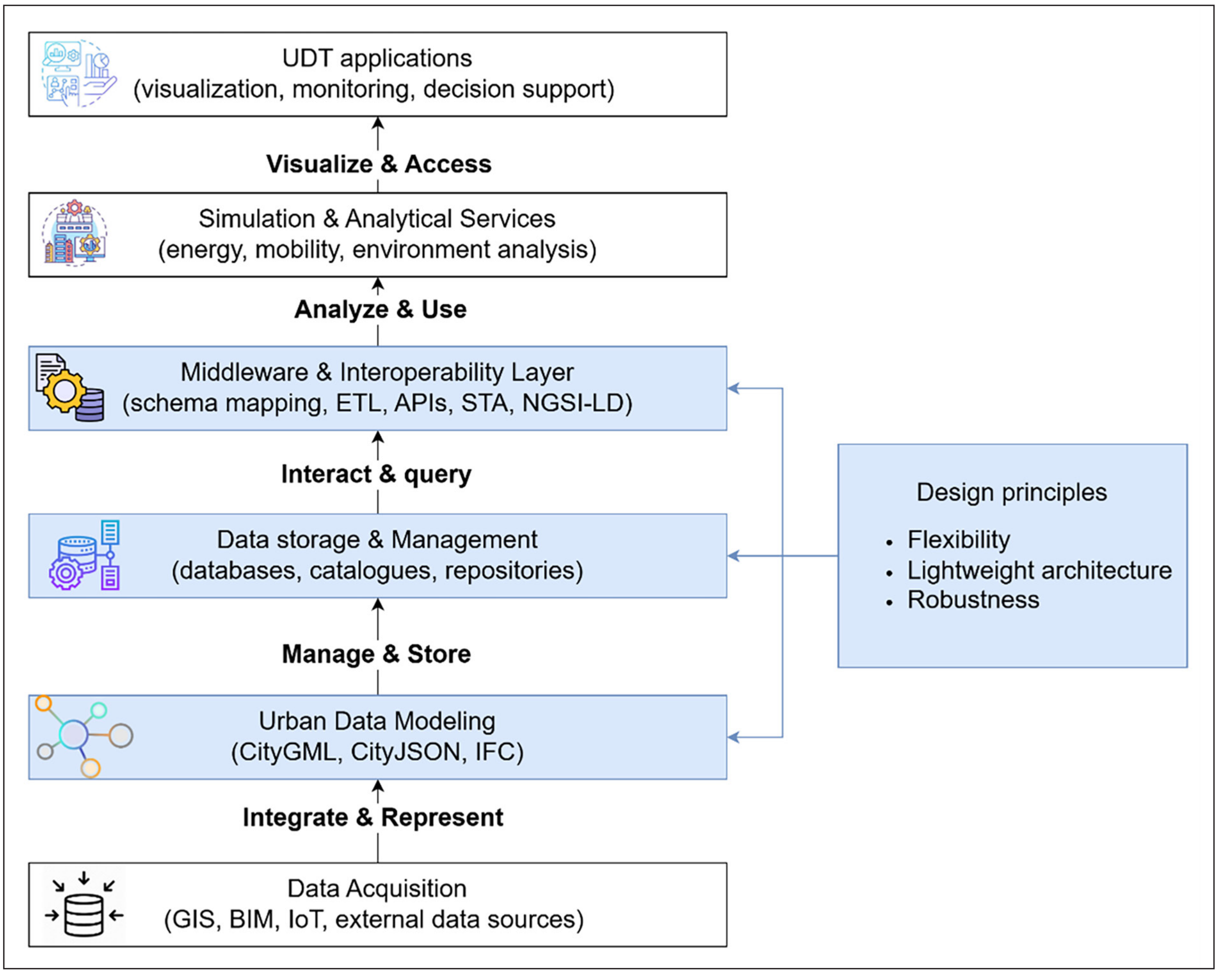

Figure 3. General layered architecture of a flexible and lightweight Urban Digital Twin (UDT), illustrating the main components from data acquisition to applications and the design principles guiding their integration

A. Semantic Data Management Layer for Flexible UDT Architectures

In this section, the main requirements addressed are structured semantic integration, multi-scale support, accessibility, and maintainability. Urban Digital Twins (UDT) increasingly integrate large volumes of heterogeneous and dynamic thematic data, notably originating from Internet of Things (IoT) devices such as environmental sensors, traffic monitoring systems, and building energy management infrastructures. These data streams are characterized by high frequency, evolving schemas, and significant volume, making them poorly suited to traditional relational database management systems.

To address these challenges, (Nys and Billen, 2021) proposed Measur3D, a CityGML-oriented implementation based on a document-oriented NoSQL database, namely MongoDB. Leveraging the flexible and scalable nature of document databases, Measur3D enables the efficient management of semi-structured and high-volume data typically associated with UDT use cases. In this approach, CityGML concepts are mapped to a limited number of collections, including CityModel, AbstractCityObject, Geometry, Texture, and Material. While the abstract city object follows a predefined structure, thematic objects are stored as BSON documents inspired by the CityJSON schema, allowing for flexible extensions and evolution over time.

However, this flexibility comes at the cost of schema enforcement and semantic consistency. MongoDB does not natively support complex constraints comparable to those of relational databases, and therefore cannot guarantee compliance with the CityGML conceptual model. To overcome this limitation, (Nys, 2023) introduced CERBERE, a middleware positioned between client applications and the database.

CERBERE embeds the full CityGML schema and acts as a bidirectional validation and filtering layer. All write and read operations are validated against application-specific schema definitions before being transmitted to or retrieved from the database. This ensures that only semantically valid and correctly structured data are stored, while allowing different applications to interact with the same underlying dataset using distinct, possibly partial, schema views. In this architecture, the database remains a flexible storage backend, while consistency and semantic integrity are enforced at the middleware level.

This design provides a high degree of architectural flexibility: schema evolutions can be accommodated without modifying the database schema or existing data, and multiple applications can coexist while sharing common identifiers for city objects. CERBERE thus reconciles the scalability and adaptability of NoSQL databases with the strict semantic requirements of CityGML-based UDTs.

Nevertheless, despite its advantages for big data and semantic flexibility, the Measur3D–CERBERE solution remains limited in terms of spatial analysis capabilities. MongoDB offers only basic 2D spatial indexing and querying, and lacks native support for advanced 3D spatial operations. As a result, while Measur3D and CERBERE are well suited for managing large-scale thematic and IoT data in UDTs, they must be complemented by relational GIS databases to fully support complex 3D geoprocessing requirements. Therefore, beyond its current integration with Measur3D, the CERBERE API could be extended to interface with other CityGML-oriented database implementations, such as 3DCityDB and CJDB (Kasprzyk *et al.*, 2024). By abstracting data access through a unified API layer, CERBERE could act as a common semantic and transactional gateway, dispatching queries to the most appropriate backend depending on the nature of the requested operations (Figure 4).

This architecture contributes primarily to the requirements of flexibility and semantic robustness identified in Section II by allowing schema evolution without database restructuring and by enforcing semantic validation through the middleware layer.

B. Lightweight Web-Based UDT Architecture

In this part, we illustrate the main requirements related to multi-scale support, federated data integration, structured semantic integration, and operational feasibility. Motivated by these requirements, we designed City2Twin, an open-source Urban Digital Twin (UDT) framework to

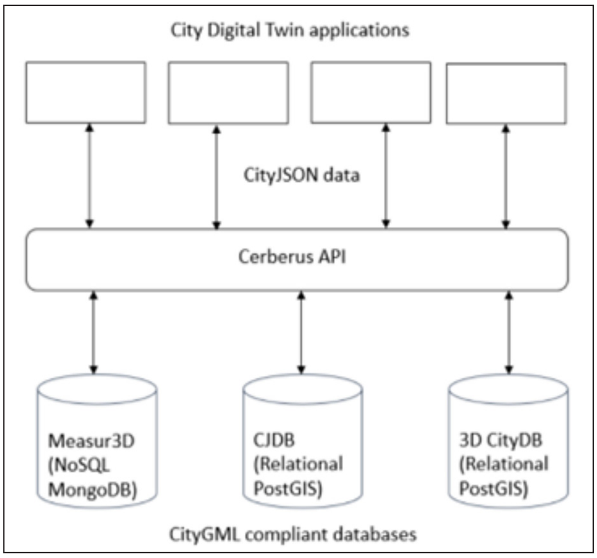

Figure 4. CERBERE architecture illustrates a flexible semantic validation layer supporting scalable data storage while preserving CityGML semantic consistency. (Kasprzyk *et al.*, 2024)

integrate, manage, and visualize heterogeneous urban datasets in a lightweight and scalable environment (Benirina Parfait *et al.*, 2024). Its originality lies in adopting CityJSON and the CityJSON Database (CJDB), a simpler, lean and developer-friendly alternative to traditional CityGML/3DCityDB solutions. Figure 5 illustrates how the proposed architectural principles, modular data integration, lightweight storage, and web-based visualization are implemented within the City2Twin framework.

1. Datasets

The datasets used in this study are summarized in Figure 6. Static data includes two 3D city models based on CityGML and encoded in CityJSON. one file containing buildings, roads, and vegetation, and a second file containing city furniture objects (i.e., lampposts). These models were generated from PICC vector data and airborne LiDAR over the Outremeuse district and its surroundings and are complemented by a LAS point cloud and a 2D GeoJSON layer. CityJSON is adopted because of its compact, flattened structure and its native compatibility with CJDB, which stores the models as CityJSONLines. Dynamic data concern air-quality measurements in Liège, provided by ISSEP (Public Service Scientific Institute) via APIs and measured at fixed monitoring stations distributed across the city (refer to Figure 6). The dataset includes pollutant concentrations (e.g. $PM_{2.5}$, PM_{10}, NO_2, O_3), temperature, and sensor metadata, which are later processed to derive the BelAQI air-quality index and associated time series for the UDT.

As shown in Figure 5, the platform is structured around three principal components: data preparation and integration, data storage, and data visualization and analysis.

2. Data Preparation and Integration (Figure 5a)

The framework starts with preprocessing of the 3D city model, including ground reprojection, geometry type conversion, and transformation into CityJSON sequences (CjSeq). Sensor positions and dynamic data such as air quality measurements are retrieved through external APIs and configured using a REST interface built with Flask. External datasets, including point clouds or GeoJSON, are reprojected or tiled when needed. All these elements are integrated through a unified data integration workflow that prepares static and dynamic sources for storage.

3. Data Storage (Figure 5b)

City2Twin stores the 3D model in CJDB, chosen for its lightness, simplicity, and natural compatibility with CityJSON encoding. Advanced spatial operations are supported through PostgreSQL/PostGIS. Dynamic or time-varying data is stored separately in a SensorThings API compliant database (OGC STA DB), enabling robust management of time series (Figure 7). Static and dynamic datasets are linked through unique object identifiers following the CityThings concept (Santhanavanich and Coors, 2021).

4. Data Visualization and Analysis (Figure 5c)

Visualisation and analysis in City2Twin are based on a direct connection between the client and the database, which contrasts with conventional workflows that rely on manual export to web-optimised formats such as 3D Tiles or glTF. In our approach, static 3D objects are retrieved on demand from CJDB using SQL queries and exported as CityJSON through the CJDB exporters and server-side conversions (CjSeq). These data are then streamed to an interactive

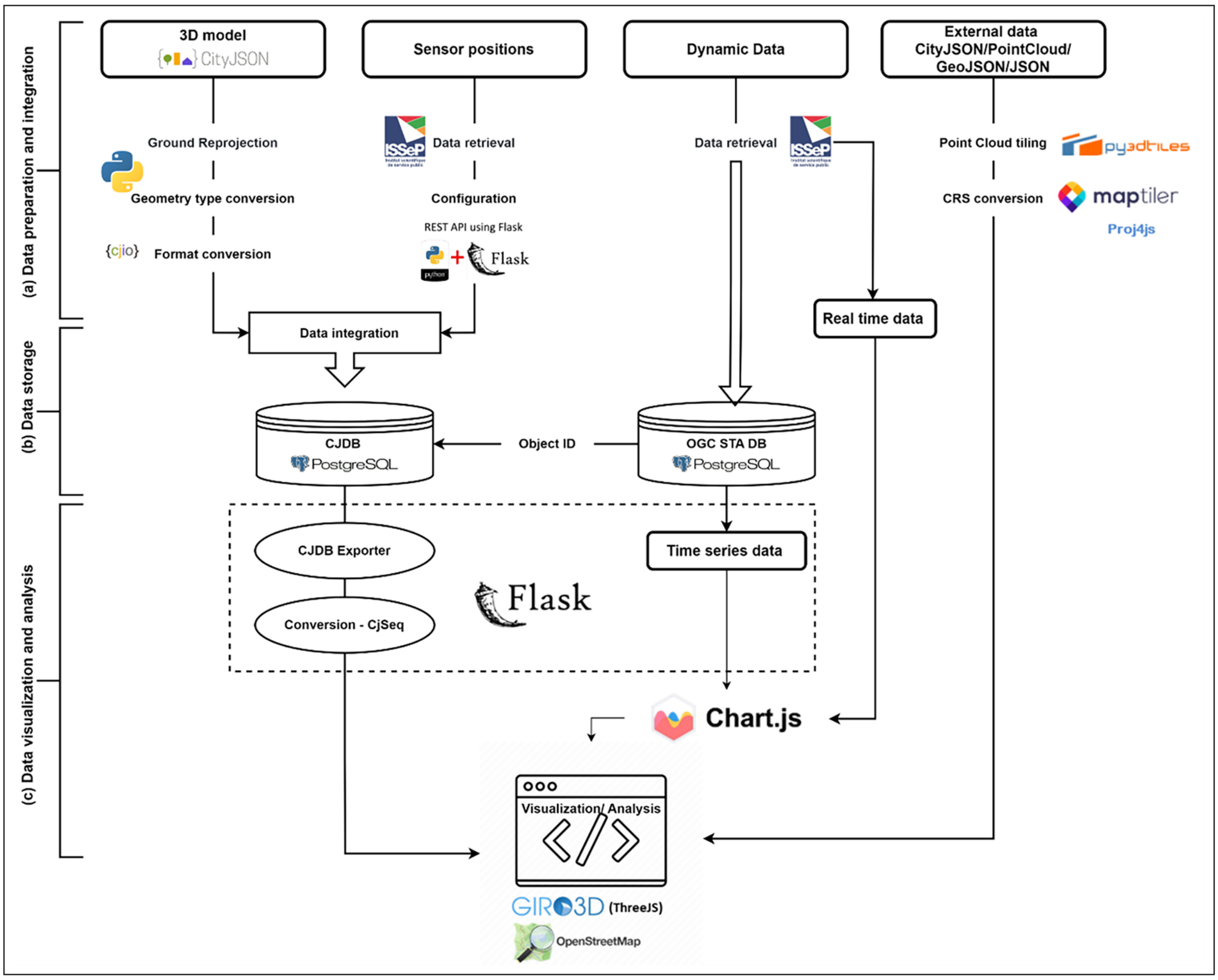

Figure 5. City2Twin architecture illustrating the implementation of a lightweight UDT workflow: (a) data integration, (b) storage layer, (c) web-based visualization and analysis (Benirina Parfait *et al.*, 2024)

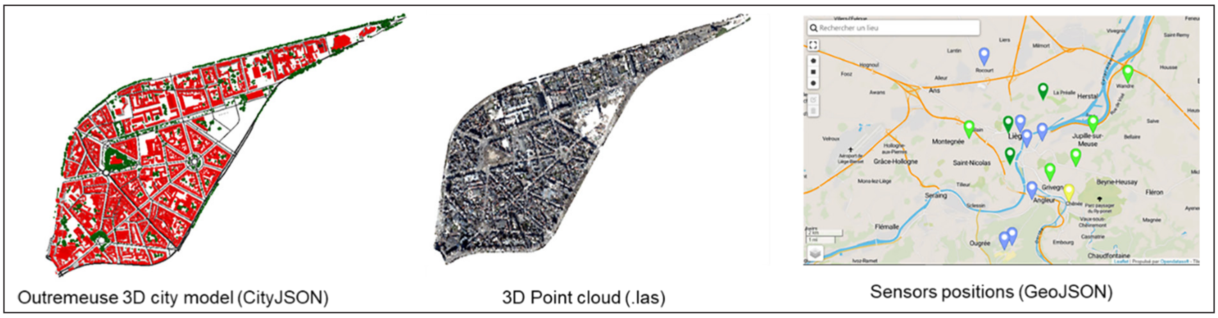
Outremeuse 3D city model (CityJSON)
3D Point cloud (.las)
Sensors positions (GeoJSON)

Figure 6. Datasets overview

web interface implemented with Giro3D/Three.js. This direct approach avoids an intermediate tiling or cloud-hosting step and ensures that any change in the database (e.g. editing geometries or attributes) is immediately reflected in the viewer. Users can explore 3D objects, semantic surfaces and urban attributes, import and modify 2D/3D data, and apply attribute- or spatial-based filters (e.g. buffering, intersections), while remaining linked to the authoritative database.

For dynamic data, observations stored in the STA database are visualized either in near-real-time or as time series using Chart.JS, enabling the exploration of pollutant concentration trends, temporal aggregation and temporal filtering. In this way, City2Twin handles both static and dynamic content within a consistent web environment, without requiring separate export pipelines for visualization.

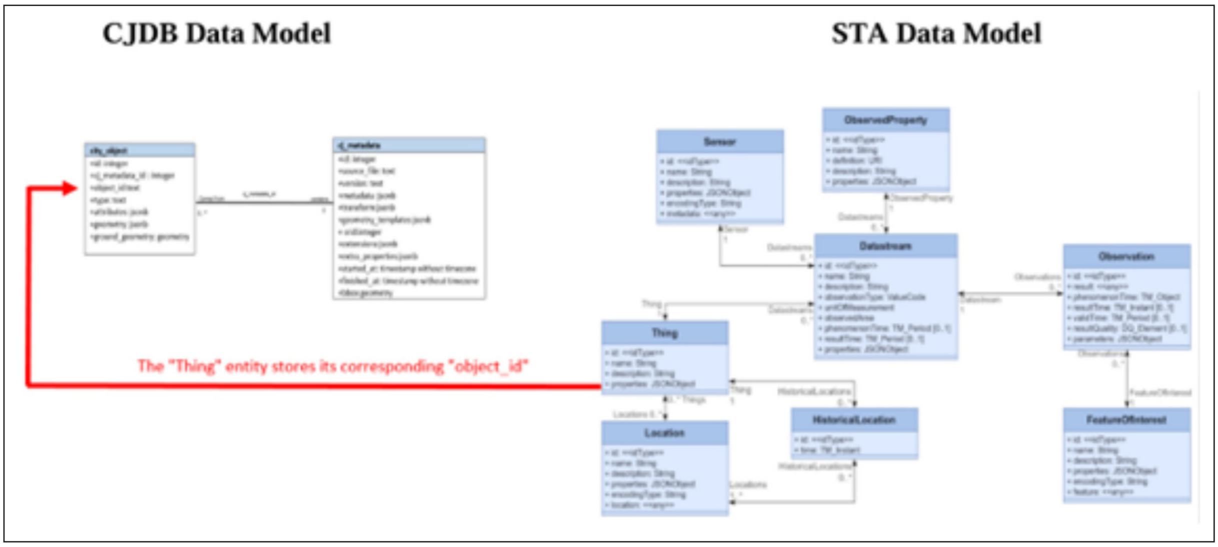

Figure 7. CityThings concept: illustrating the link of data models of the two databases (static and dynamic)

City2Twin is, to our knowledge, one of the first Urban Digital Twins built entirely on CityJSON and CJDB, addressing a gap in the literature regarding open, lightweight UDT architectures. The platform offers several advantages compared to typical UDT implementations: a fully open-source workflow that supports reproducibility and extensibility; a direct viewer database connection that enables continuous updates without manual exports; and integrated analysis functionalities that go beyond those of existing CityJSON tools such as Ninja. It also provides a flexible mechanism for incorporating simulation results (e.g. energy demand, heating loads) and other thematic datasets. City2Twin is designed to support multiple user profiles. GIS professionals gain an intuitive interface for visualizing and editing 2D/3D spatial data; citizens can monitor air quality in near real time and explore temporal patterns (i.e., average hourly values over a selected period); decision-makers and urban planners can perform temporal and spatial analyses to support environmental and urban planning; and energy managers can inspect building-level simulation data directly within the 3D model. Overall, City2Twin illustrates how Urban Digital Twins can be implemented in an accessible and maintainable way, while serving multi-domain purposes and contributing to more open, sustainable and resilient urban management tools.

The architecture primarily supports the lightweight design principle, relying on web-native encodings (CityJSON) and direct database-viewer connection.

C. Automated Data Integration and Processing Pipelines

From a technical point of view, we developed different automatic and semi-automatic workflows whenever needed, both to compensate for the limitations of individual tools and methods and to address specific UDT objectives. Depending on the task, these workflows rely on FME as a visual ETL environment, on custom Python scripts for more domain specific processing, and on other ad hoc tools that support data preparation, integration, and visualization within the Urban Digital Twin.

For instance, FME[1] workflows follow data-pipeline logic of processing to transformation into modelling (Figure 8). Data preparation uses FME readers to load PICC topographic data, airborne LiDAR and other thematic layers for the area of interest. Transformation pipeline then enriches the data and reconstructs 3D city objects: building footprints are combined with LiDAR-derived heights to create block models (LoD1), a TIN is generated for the terrain, roads are simplified and adjusted to this surface, and trees are instantiated from templates in accordance with the CityJSON specification (geometry_templates). At the same time, attributes such as unique IDs, object types, LoDs and semantic surfaces are mapped so that the resulting features conform to the target CityJSON schema and can later be linked to observations or simulation outputs. In the storage step, these features are written to compact CityJSON files and, where required, loaded into PostgreSQL/PostGIS and 3DCityDB, enabling both file-based exchange and database queries. Finally, for visualization purposes, CityJSON can be visualized in the client side such as Ninja for inspection and CesiumJS or Giro3D for advanced web-based 3D visualization.

Alongside FME, custom Python scripts complement this workflow by handling specialized preprocessing and database interaction tasks, ensuring greater flexibility in how data are transformed and managed. To tackle common visualization issues encountered in the 3D web viewers, we developed two custom Python scripts. The first script, based on the “pyproj” library, converts the 3D models from EPSG:3812 to EPSG:4326 so that they can be displayed correctly on the CesiumJS globe. Performing this operation on the server side avoids expensive on-the-fly reprojection in the browser and helps to reduce loading times. The second script addresses the problem of 3D objects “floating” above the terrain (Figure 9). Using cjio, the CityJSON file is read, the minimum Z value is computed for each object, and this value is subtracted from all its vertices so that the object is flattened onto Z = 0. To preserve altitude information for future work, rather than serving visualization purposes only, the minimum Z is stored as a Zmin attribute and can be added back to the geometry if true elevations are needed again. This solution works well for buildings, trees, and most other city objects, even though it intentionally flattens vertical variation in roads and buildings for the purposes of this study. The same procedure is, for instance, applied to vegetation on the Sart-Tilman campus, where trees are draped onto the ground surface to ensure consistent visual integration with the base terrain in the 3D viewers.

Python scripts are also developed to integrate the observation and simulation outputs by implementing data pipelines that feed the Urban Digital Twin. For air-quality IoT data, a Flask-based service retrieves measurements from an external API every five minutes, checks whether values have changed, and records only new observations in the PostgreSQL database (STA DB). After an initial experiment with a FROST SensorThings server, a direct approach was adopted. Data are fetched with requests.get(), parsed as JSON, grouped to match the database schema (e.g. $PM_{2.5}$ and PM_{10} lists), and inserted efficiently using psycopg2 and cursor.executemany() in a dedicated background thread. A similar pattern is used to ingest energy simulation outputs. Results exported as CSV files by external tools (e.g. SimStadt) are processed in Python to align with the identifiers and structure of 3DCityDB (including the Energy ADE tables) and then written into the database so that simulated indicators and time series can be attached directly to the corresponding buildings.

In conclusion, FME workspaces and Python scripts form a coherent and extensible methodology in which FME provides an interactive, widely used ETL alternative for building LoD1 city models, while Python offers automation for coordinate conversion, ground projection and high-frequency ETL between external APIs, simulation tools and the 3D city database ultimately enabling both students and researchers to maintain dynamic, up-to-date 3D views within the Urban Digital Twin. These automated workflows support flexibility and maintainability by enabling reproducible data pipelines.

D. Extending UDT Architectures Toward Robotic and Edge Systems

GIS4IoRT is an ongoing research and development project aiming at the design of a cloud-based middleware positioned between GIS clients (or Urban Digital Twin interfaces) and robotic systems (Wrembel *et al.*, 2025) (Figure 10). Its primary objective is to bridge the gap between robotic platforms, which are not natively compatible with GIS and OGC standards, and geospatial infrastructures used in Urban Digital Twins. By acting as an intermediary layer, GIS4IoRT enables bidirectional interactions between robots and GIS-based applications while preserving interoperability, scalability, and semantic consistency.

While Internet of Things (IoT) technologies have become a central component of Urban Digital Twins, emerging use cases increasingly rely on mobile, autonomous, and semi-autonomous systems, giving rise to the paradigm of the Internet of Robotic Things (IoRT). Unlike traditional IoT devices, robotic systems typically rely on dedicated communication frameworks and middleware, such as ROS (Robot Operating System), which are optimized for real-time control, sensor fusion, and task coordination, but remain largely disconnected from geospatial standards and GIS-oriented

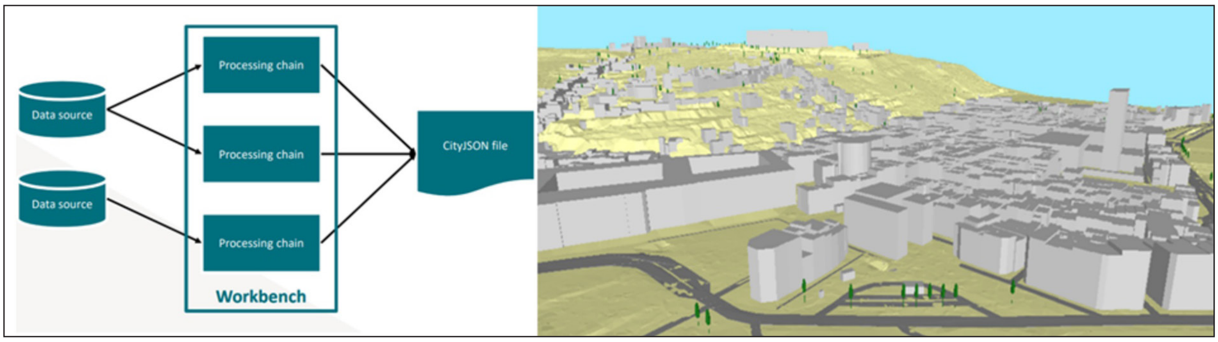

Figure 8. 3D city modeling in FME: an illustration of the model generated in Liège, Belgium (Nys, 2023)

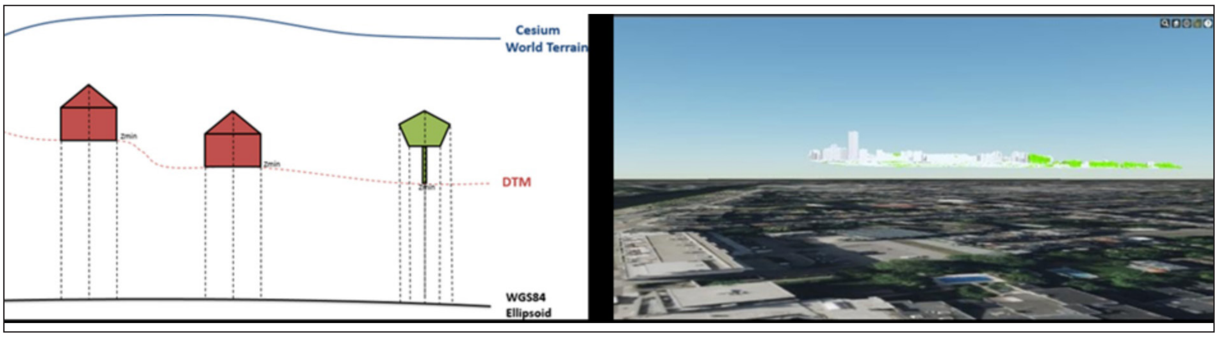

Figure 9. Ground object reprojections: (a) Difference between elevation levels of different surfaces, (b) 3D city model floating above the ground surface in CesiumJS (Rafamatanantsoa *et al*., 2024)

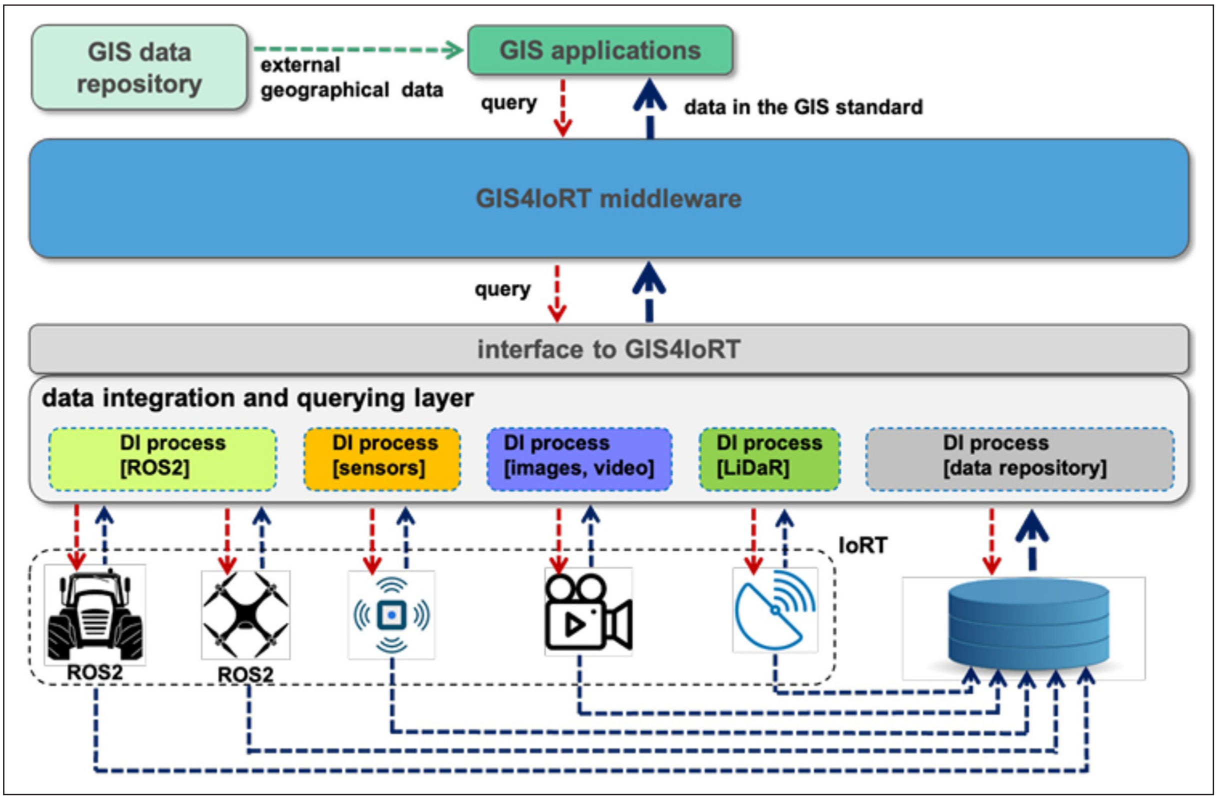

Figure 10. GIS4IoRT architecture (Wrembel *et al*., 2025)

workflows. GIS4IoRT specifically addresses this incompatibility by providing adaptation mechanisms between ROS-based ecosystems and GIS/OGC-compliant services.

A key distinction between IoT and IoRT lies in communication protocols and interaction models. While IoT systems commonly use lightweight publish–subscribe protocols such as MQTT or HTTP-based REST services, IoRT applications adopt robotic middleware that supports continuous data exchange, feedback loops, and command execution. Current developments within GIS4IoRT focus on exposing robotic observations and capabilities through GIS-compatible service interfaces, enabling their integration into UDT platforms and geospatial analysis pipelines.

Another defining characteristic of IoRT is the distribution of processing across multiple levels of the system. Data processing and decision-making may occur directly on robotic nodes (edge computing), within intermediate stream-processing platforms, in cloud-based infrastructures, or in client-side GIS applications. GIS4IoRT investigates flexible cloud-based architectures capable of dynamically routing data and tasks depending on latency constraints, computational resources, and application objectives. To support these requirements, stream-oriented technologies such as Apache Kafka and NebulaStream are currently explored to enable scalable ingestion, processing, and dissemination of heterogeneous data streams.

Beyond passive data collection, IoRT introduces a strong tasking dimension, where robotic agents (e.g., drones, ground robots, autonomous vehicles) can be assigned missions and dynamically re-tasked based on contextual information or analytical results. GIS4IoRT aims to integrate this tasking capability into GIS and UDT interfaces, allowing spatial analyses, simulation results, or decision-support tools to directly influence robotic behavior. This bidirectional interaction extends the traditional role of GIS from data visualization to active orchestration within the UDT ecosystem.

The mobility of robotic platforms further raises specific challenges related to the management of moving features. Unlike static IoT sensors, robotic agents continuously change position and orientation, generating spatio-temporal trajectories that must be stored, queried, and analyzed efficiently. Ongoing work within GIS4IoRT explores the use of spatio-temporal data models and operators, notably through MobilityDB and its MEOS library, to support advanced spatio-temporal queries involving mobile agents and urban objects.

IoRT-based UDTs must also cope with extreme data heterogeneity, both in terms of sources and data structures. Typical inputs include drone imagery, LiDAR point clouds, video streams, in-situ IoT sensors, robotic telemetry, and derived analytical products. These data streams operate at very different frequencies, ranging from high-rate sensor measurements and video feeds to lower-frequency status updates or batch-oriented datasets. GIS4IoRT addresses the challenge of harmonizing these multi-frequency streams and ensuring their spatial and temporal alignment within a unified geospatial framework.

Finally, GIS4IoRT is designed with transferability across application domains in mind. While initial developments and experimental deployments have focused on agricultural use cases—where robotic systems and real-time sensor data are already widely adopted—the project explicitly targets future applications in Urban Digital Twins. The cloud-based middleware approach and architectural principles being developed interoperability, distributed processing, tasking, and spatio-temporal reasoning are intended to be reusable across domains, supporting evolutive and scalable UDT infrastructures. This approach extends UDT architecture toward scalable and evolutive infrastructures capable of integrating mobile sensing systems.

CONCLUSION

This paper has argued that Urban Digital Twins (UDTs) should be conceived as evolutive, data-centric infrastructures rather than as monolithic software systems. From this perspective, flexibility, accessibility, and maintainability emerge not as secondary implementation choices, but as defining properties of sustainable UDT architectures. In particular, flexibility is a fundamental property of UDTs because urban systems, data sources, institutional arrangements, and application needs continuously evolve over time. As shown throughout this paper, such flexibility can be achieved through the combined use of semantic 3D city models, lightweight encodings, interoperable APIs, modular data pipelines, and loosely coupled architectural components.

Building on this perspective, the paper examined how these requirements can be operationalized through the GeoScITY experience. Through a set of concrete tools developed within the GeoScITY lab namely City2Twin, CERBERE, Measur3D, automated ETL workflows, and GIS4IoRT—we showed how standards-based and modular design choices can support the incremental integration of heterogeneous urban data, simulation outputs, sensor observations, and robotic systems within a common UDT ecosystem. In this sense, GeoScITY does not define flexibility as a property of one specific framework but rather illustrates how flexibility can be implemented in practice within a coherent and extensible UDT architecture. More broadly, the results highlight that semantic 3D city models can provide a stable spatial and semantic backbone for integrating heterogeneous datasets, while lightweight encodings and database solutions improve web interoperability and operational usability. At the same time, validation middleware, automated transformation workflows, and API-based services help ensure that UDT infrastructures remain accessible, maintainable, and adaptable as technologies and use cases evolve. These elements are particularly important in public-sector and research contexts, where UDTs must support gradual development, cross-domain interoperability, and long-term sustainability rather than rely on rigid and monolithic deployments. The GeoScITY tools discussed in this paper provide concrete evidence of how such principles can be translated into operational components. City2Twin demonstrates the incremental integration of static and dynamic urban datasets around semantic 3D city models. CERBERE and Measur3D show how semantic consistency and validation can be maintained independently from storage back ends, thereby supporting evolutive data management. Automated ETL workflows contribute to reproducibility and accessibility by simplifying recurrent integration and update tasks. GIS4IoRT further extends this logic by enabling bidirectional coupling between geospatial digital twins and robotic ecosystems, opening new possibilities for action-oriented UDTs in which sensing, analysis, and intervention are more tightly connected.

Overall, this paper concludes that flexibility is a defining property of UDTs and that it can be achieved through modular, standards-based, and interoperable design strategies. The GeoScITY experience shows how these principles can be implemented through concrete tools and workflows, but the broader contribution of the paper is to propose a transferable perspective on UDT design: one that prioritizes openness, incremental development, and system interoperability over exhaustive but rigid digital replication. Such an approach is essential if UDTs are to remain reusable, evolutive, and meaningful in the face of changing urban conditions, technologies, and governance contexts.

Future work will focus on strengthening governance and provenance mechanisms, improving support for high-frequency and real-time data streams, and further validating these tools in long-term operational settings with public authorities and external stakeholders. In this respect, GeoScITY should be understood not only as a technical ecosystem, but also as an empirical basis for reflecting on the next generation of flexible, lightweight, and action-oriented Urban Digital Twins.

NOTES

[1] https://fme.safe.com/

BIBLIOGRAPHY

Abdelrahman, M., Macatulad, E., Lei, B., Quintana, M., Miller, C., Biljecki, F., 2025. What is a Digital Twin anyway? Deriving the definition for the built environment from over 15,000 scientific publications. *Building and Environment* 274, 112748. https://doi.org/10.1016/j.buildenv.2025.112748

Acharya, S., Khan, AA., Päivärinta, T., 2024. Interoperability levels and challenges of digital twins in cyber–physical systems. *Journal of Industrial Information Integration* 42, 100714. https://doi.org/10.1016/j.jii.2024.100714

Alva, P., Biljecki, F., Stouffs, R., 2022. Use cases for district-scale urban digital twins. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*

XLVIII-4/W4-2022, 5–12. https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-5-2022

Batty, M., 2025. How relevant is the digital twin to the urban world? Dialogues in Urban Research 27541258251315136. https://doi.org/10.1177/27541258251315136

Batty, M., 2018. Digital twins. Environment and Planning B: *Urban Analytics and City Science* 45, 817–820. https://doi.org/10.1177/2399808318796416

Bauer, M., Cirillo, F., Fürst, J., Solmaz, G., Kovacs, E., 2021. Urban Digital Twins – A FIWARE-based model. at - *Automatisierungstechnik* 69, 1106–1115. https://doi.org/10.1515/auto-2021-0083

Benirina Parfait, R., Jeddoub, I., Yarroudh, A., Hajji, R., Billen, R., 2024. City2Twin: an open urban digital twin from data integration to visualization and analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-2/W8-2024, 387–394. https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-387-2024

Coenen, T., Chantillon, M., Croket, I., Raes, L., 2025. Local Digital Twins: A Central Urban Integrator to Break Down the Silos. *In* Raes, L., Ruston McAleer, S., Croket, I., Kogut, P., Brynskov, M., Lefever, S. (Eds.), *Decide Better: Open and Interoperable Local Digital Twins. Springer Nature Switzerland*, Cham, pp. 59–86. https://doi.org/10.1007/978-3-031-81451-8_4

Jeddoub, I., Nys, GA., Hajji, R., Billen, R., 2024. Data integration across urban digital twin lifecycle: a comprehensive review of current initiatives. *Annals of GIS* 0, 1–20. https://doi.org/10.1080/19475683.2024.2416135

Jeddoub, I., Nys, GA., Hajji, R., Billen, R., 2023. Digital Twins for cities: Analyzing the gap between concepts and current implementations with a specific focus on data integration. *International Journal of Applied Earth Observation and Geoinformation* 122, 103440. https://doi.org/10.1016/j.jag.2023.103440

Kasprzyk, JP., Nys, GA., Billen, R., 2024. Towards a multi-database CityGML environment adapted to big geodata issues of urban digital twins. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-4-W10-2024, 101–106. https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-101-2024

Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. *Digital Twin in manufacturing: A categorical literature review and classification*. IFAC-PapersOnLine, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018 51, 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

Lei, B., Janssen, P., Stoter, J., Biljecki, F., 2023. *Challenges of urban digital twins: A systematic review and a Delphi expert survey. Automation in Construction* 147, 104716. https://doi.org/10.1016/j.autcon.2022.104716

Lu, Q., Parlikad, AK., Woodall, P., Xie, X., Liang, Z., Konstantinou, E., Heaton, J., Schooling, J., 2019. Developing a dynamic digital twin at building and city levels: A case study of the West Cambridge campus. *Journal of Management in Engineering* 36. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000763

Noardo, F., 2022. Multisource spatial data integration for use cases applications. *Transactions in GIS* 26. https://doi.org/10.1111/tgis.12987

Nys, GA., Billen, R., 2021. From consistency to flexibility: A simplified database schema for the management of CityJSON 3D city models. *Transactions in GIS* 25, 3048–3066. https://doi.org/10.1111/tgis.12807

Nys, GA., 2023. *From consistency to flexibility: shifting the structure. Towards a new generation of geographical information systems*. PhD thesis, ULiège.

Santhanavanich, T., Coors, V., 2021. CityThings: An integration of the dynamic sensor data to the 3D city model. *Environment and Planning B: Urban Analytics and City Science* 48, 417–432. https://doi.org/10.1177/2399808320983000

Stoter, JE., Arroyo Ohori, GAK., Noardo, F., 2021. Digital Twins: A Comprehensive Solution or Hopeful Vision? *GIM International: the worldwide magazine for geomatics* 2021.

Wrembel, R., Kasprzyk, JP., Billen, R., Bimonte, S., Sacharidis, D., Skrzypczyński, P., 2025. *On integrating robotic data with GIS tools in a cloud environment*, *in:* Proceedings of the Workshops of the EDBT/ICDT 2025 Joint Conference. Presented at the 9th International Workshop on Data Analytics solutions for Real-LIfe APplications (DARLI-AP), Barcelona (ES).

Yan, J., Lu, Q., Li, N., Chen, L., Pitt, M., 2025. Common data environment for digital twins from building to city levels. *Automation in Construction* 174, 106131. https://doi.org/10.1016/j.autcon.2025.106131

Coordinates of authors:

Imane JEDDOUB
GeoScITY Lab
ULiège
I.Jeddoub@uliege.be

Jean-Paul KASPRZYK
GeoScITY Lab
ULiège
jp.kasprzyk@uliege.be

Roland BILLEN
GeoScITY Lab
ULiège
rbillen@uliege.be