ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE AND ENGINEERING

Ciclo 38

Settore Concorsuale: 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

Settore Scientifico Disciplinare: ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

ENABLING SECURE, INTEROPERABLE, AND TRUSTWORTHY URBAN DIGITAL TWINS

Presentata da: Andrea Roberta Costagliola

Coordinatore Dottorato

Paola Salomoni

Supervisore

Paolo Bellavista

Co-supervisore

Luciano Bononi

Esame finale anno 2026

Abstract

Modern digital infrastructures are evolving into decentralized and federated ecosystems that integrate data, services, and decisions across multiple domains and actors. In this context, Urban Digital Twins (UDTs) are emerging as dynamic platforms for simulating, analyzing, and governing urban information. By federating data and data sources from public authorities, private organizations and citizens under controlled sharing agreements, these systems enable secure and accountable data-driven planning and collaborative decision-making, ensuring that sensitive information is exchanged within well-defined trust and governance boundaries. However, their deployment still faces structural and systemic limitations, including semantic heterogeneity, data fragmentation, temporal inconsistency, and weak governance mechanisms, that hinder current implementations in managing heterogeneous models, preserving the evolution of spatial and administrative entities, and ensuring sovereignty and accountability across distributed infrastructures. These constraints limit the ability of UDTs to integrate and align data, processes, and governance rules across domains and administrative levels. The dissertation addresses these challenges from an architectural and methodological perspective, introducing a set of frameworks that together advance the interoperability, sovereignty, and trustworthiness of UDT. The research introduces two complementary integration architectures that address data fragmentation and interoperability across heterogeneous urban systems: a lightweight framework for modular service orchestration and scalable data exchange, and a semantic backbone that supports structured reasoning and cross-domain alignment. Building on the latter, a temporal data framework extends the semantic backbone with mechanisms for diachronic reconstruction and multi-temporal reasoning. To ensure sovereign and verifiable data exchange, a decentralized data space model guarantees ownership, integrity, and secure access control through decentralized identity and credential management aligned with the principles of Self-Sovereign Identity (SSI) and Findable, Accessible, Interoperable, and Reusable (FAIR) data. Finally, a federated process governance framework introduces mechanisms for tracing contributions, estimating influence, and mitigating adversarial updates, ensuring transparency and accountability in distributed decision-making. Collectively, these contributions provide the building blocks for interoperable, temporally aware, and trustworthy UDT ecosystems.

Contents

List of Figures

List of Tables

1 Introduction

Modern digital infrastructures are rapidly transforming into complex adaptive ecosystems composed of different participants, including individuals, organizations, and artificial agents that interact to share data, services, and decisions. These socio-technical environments are no longer based on centralized authorities, but depend on federated, decentralized, and often loosely connected architectures [3, 84]. In these contexts, interoperability, sovereignty, and trust are essential for cooperation across different domains, jurisdictions, and organizational boundaries.

One of the most significant technological changes driving this transformation is Digital Twins (DTs). A DT is a virtual model that continuously represent the state, behavior, and development of physical systems using real-time data streams, simulation models, and AI-based analytics [138, 80]. Digital twins are used in numerous sectors, including manufacturing, healthcare, transportation, energy, and smart cities [88] to optimize performance, monitor operations and support decision-making.

When applied to urban contexts, DT are called Urban Digital Twins (UDTs). UDTs act as dynamic, multi-layered platforms that bring together different data from public authorities, private organizations, and citizens to facilitate planning, governance, and urban service improvement[20, 158, 146].

They provide a shared digital space where urban knowledge can be collaboratively built, explored, and queried. Among the possible operations, planners can simulate the long-term impact of zoning policies; mobility experts can test alternative traffic scenarios; environmental agencies can assess pollution spread based on sensor data.

Despite their potential, the implementation of UDTs faces structural and systemic challenges. Urban data are often fragmented across different silos, available in heterogeneous formats and governed by inconsistent policies. The reliance on rigid standards and descriptors reinforces separation between systems and limits cross-domain interoperability. Urban systems consist of multiscale layers such as buildings, infrastructures, vehicles, citizen and administrative actors, each described and shaperd by distinct ontologies, standards and operational logics [108]. Filling the gap between these urban levels to enable consistent semantic interpretation remains a major challenge for research. For example, matching energy consumption

data with mobility flows or connecting zoning regulations with environmental quality indicators, requires interoperable models that current UDT implementations often lack [79].

However, the challenge is not only semantic. The growing complexity of urban environments is also linked to the increasing volume and distribution of data [65]. Cities generate vast and continuous data flows from a multitude of sources such as IoT sensors embedded in infrastructure, edge devices distributed throughout neighborhoods, and cloud-based services managed by public and private actors. This informations must be acquired, transformed, and aligned in real-time, requiring complex backend logic to guarantee reliability, consistency, and trust.

A further limitation concerns the evolving nature of urban knowledge itself. Urban systems are not static, as administrative boundaries, zoning categories, infrastructure networks, and demographic patterns are constantly changing. However, most urban data technologies focus on real-time simulation and operational control, neglecting the need for historical reasoning and multi-temporal alignment [111]. Without structured support for long-term urban transformations, digital twins cannot enable diachronic analysis, evaluate urban trajectories, or assess the cumulative impact of planning decisions.

As UDTs evolve into distributed ecosystems, the question of who owns, controls, and accesses data are becoming increasingly critical. The integration of heterogeneous data sources often includes sensitive information about entities, requiring robust protection mechanisms to prevent unauthorized access, ensure confidentiality, and maintain public trust in digital infrastructures [142]. In these federated environments, data are no longer centralized under a single authority but shared among actors operating under different governance models. Each actor retains control over its own datasets while contributing to a shared ecosystem through standardized interfaces and verifiable exchanges. Consequently, ensuring that data can be shared securely, accessed transparently, and verified reliably is essential to sustaining cooperation and accountability.

However, ensuring trust in data access alone is not sufficient. This is because processes based on these decentralized data flows also become decentralized. In UDTs, many critical operations such as model training, anomaly detection, and predictive analytics are distributed across multiple actors and infrastructures. As a result, it becomes essential to trace the origin of data contributions, assess their influence on shared models, and revoke incorrect or biased results when necessary [58]. In federated learning environments, where raw data remains local and only model updates

are exchanged, traditional oversight mechanisms are no longer applicable. This gives rise to the need for a framework that can ensure transparency, accountability, and trust throughout the entire lifecycle of collaborative computation.

To build reliable, interoperable, and sovereign digital ecosystems, it is necessary to consistently integrate and align various architectural dimensions, including:

- the need for semantic and technical interoperability across vertical and horizontal layers of urban systems, enabling modular integration and multi-level reasoning across heterogeneous domains;
- the integration of temporal reasoning and historical data management capabilities to capture, align, and reuse evolving urban datasets over time;
- the implementation of data sovereignty and verifiability mechanisms that ensure ownership, traceability, integrity, and privacy in federated environments, aligned with FAIR and SSI principles;
- the establishment of federated process governance frameworks capable of tracing contributions, estimating influence, and revoking flawed or malicious updates in collaborative learning pipelines

This dissertation proposes a set of architectural, semantic, and governance-oriented innovations to address the structural limitations of current UDT implementations. These contributions span multiple layers of the UDT ecosystem, from data interoperability and temporal reasoning to federated trust and process accountability.

- A modular integration approach is proposed to enable semantic and operational interoperability across heterogeneous urban domains. This architecture supports scalable data exchange and coherent orchestration of services, allowing different actors and systems to interact through lightweight and extensible mechanisms, without relying on monolithic standards.
- A simulation framework is developed to represent urban environments as layered networks, where entities at varying levels of granularity can interact dynamically. This structure facilitates hierarchical abstraction and structured data publishing, enhancing semantic alignment and enabling policy-aware reasoning across interconnected subsystems.
- Introduction of a semantic-temporal element in order to acquire and manage diachronic urban transformations. By codifying space-time relationships and

quality indicators, a framework is created that supports historical reconstruction, longitudinal querying, and time-sensitive reasoning on evolving data sets.

- A decentralized data space architecture is designed to ensure data sovereignty, verifiability, and secure access control within distributed urban ecosystems. Through identity-based mechanisms, this approach enables actors to certify datasets, enforce usage policies, and support dynamic revocation, promoting transparency and trust in cross-domain data exchanges.

- A federated learning governance framework is proposed to trace contributions, estimate influence, and revoke flawed or adversarial updates in collaborative model training. Designed for complex, data-intensive urban environments, this framework supports distributed analytics and decision-making across interconnected systems. By integrating identity and credential infrastructures, it ensures accountability and reliability throughout the lifecycle of computational processes embedded in digital representations of the city.

The remainder of this dissertation is organized as follows: Chapter 2 provides the conceptual background on Digital Twins, semantic technologies, decentralized identity, and trust models in open systems. Chapter 3 outlines the technical, organizational, and governance challenges involved in building trustworthy Urban Digital Twin infrastructures, framing them into four core dimensions. Chapter 4 focuses on semantic and operational interoperability across heterogeneous urban domains, introducing two complementary architectures that address data fragmentation and enable multi-level integration. Chapter 5 extends this semantic framework with a temporal data infrastructure that manages diachronic complexity and supports longitudinal reasoning. Chapter 6 presents a decentralized data space architecture to ensure verifiable data sovereignty in federated environments. Chapter 7 proposes a federated process governance framework to enable traceability, accountability, and trust in collaborative urban analytics. Finally, Chapter 8 synthesizes the dissertation's key findings, contributions, implications, and outlines directions for future research.

List of Publications

i. A. R. Costagliola, A. Sabbioni, A. Bujari, R. Montanari, P. Bellavista. *A Multi-faceted Interoperability Model for Reliable and Trustworthy Urban Digital Twins.* Proceedings of the 2024 International Conference on Information Technology for Social Good (GoodIT '24), pp. 373–376. ACM, 2024.
DOI: 10.1145/3677525.3678684.

ii. A. R. Costagliola, M. Montanari, P. Bellavista. *Interconnecting Urban Networks: A Novel Approach to Digital Twins Through GlassBox Adaptation.* Proceedings of the 2025 Conference, pp. 193–204, Jan 2025.
DOI: 10.5220/0013495100003953.

iii. *Accepted* A. R. Costagliola, M. Montanari, P. Bellavista. *Temporal Data Containers: A Semantic Framework for Longitudinal Spatial Data Integration and Quality Assessment. Communications in Computer and Information Science (CCIS).*

iv. A. R. Costagliola, C. Mazzocca, A. Bujari, R. Montanari, P. Bellavista. *VESPACE: A verifiable blockchain-based data space solution to empower the data economy. Computer Communications*, vol. 239, p. 108180, 2025.
DOI: 10.1016/j.comcom.2025.108180.

v. *Under revision (Round 2, Minor).* N. Romandini, A. R. Costagliola, A. Bujari, R. Montanari. *Trustflow: a Traceable Federated Learning Framework to Enable Trustworthy Digital Twins. Future Generation Computer Systems.*

2 Background

This chapter presents emerging technologies for the development of secure, interoperable, and reliable UDTs. First, the concept of UDTs as dynamic and multidimensional digital counterparts of urban systems is introduced. Next, current implementations of UDTs in various cities around the world are presented and highlighted. The architectural principles that typically underpin UDT platforms are then presented, emphasizing the layered and modular designs that ensure scalability, adaptability, and integration of heterogeneous data sources. The focus then shifts to decentralized and emerging technologies that extend these architectures, such as distributed ledger technologies (DLTs), which provide the infrastructure for secure and verifiable data exchange. Finally, the section explores the SSI paradigm, including decentralized identifiers (DIDs) and verifiable credentials (VCs), as a means of giving individuals and organizations control over their digital identities while promoting trusted interactions between federated ecosystems.

2.1 Urban Digital Twins

Cities are inherently complex systems that are difficult to fully control or program. A prerequisite for building a smart city is the definition of measurable attributes, which can be continuously monitored. Sensors and actuators are therefore deployed to acquire data describing urban processes, making the city quantifiable and evaluable. Once reliable data are available, AI and machine learning techniques can identify behavioral patterns and systemic dynamics, rendering the city "comprehensible." In this sense, it becomes possible to anticipate events and trace back their associated causes through data-driven analysis.

Building on this capability, UDTs provide a structured framework to represent and operationalize such data, serving as a digital replica of the city's assets, processes, and systems[164]. They use AI algorithms, data analytics and machine learning to create digital simulation models that can be updated and changed as their physical equivalents change. Real-time, near real-time and historical data can be used in various combinations to provide the necessary capabilities for data analytics (de-

scriptive, prescriptive, predictive), simulations and what-if scenarios. UDTs thus represent a further step toward the programmability of the city. By providing a digital representation of urban resources, they enable the design of applications that interact with the twin to influence the behavior of physical entities and optimize both resource usage and system performance. For example, in the mobility domain, DTs are increasingly employed to simulate traffic flows and optimize public transport schedules [17]. Similarly, in the energy sector, DTs contribute to the balancing of smart grids and the seamless integration of renewable sources, enhancing both stability and efficiency. In the environmental domain, DTs enable real-time monitoring of air quality and noise pollution, providing actionable insights for sustainable urban management [69]. Several ongoing initiatives and experiences have already introduced the concept of digital twins within smart city contexts, confirming their potential as a foundational tool for future urban management and innovation.

2.1.1 Industrial & Acedemic initiatives

Globally, several cities around the world have developed Urban Digital Twins tailored to their specific needs.

Virtual Singapore [115] is a pioneering high-resolution 3D digital twin of the entire city-state, co-led by the Singapore Land Authority, the National Research Foundation, and the Government Technology Agency. The project was launched in 2014 as part of the Smart Nation initiative and completed in 2022. Geospatial data, real-time IoT sensor feeds, aerial LiDAR scans, and topographic data are integrated to form a rich urban model. This platform continuously assimilates real-time data, such as pedestrian movements, environmental conditions, and public services, via thousands of IoT sensors and enables urban planners and decision-makers to perform complex simulations, ranging from wind and noise analysis to flood risk assessments, solar potential mapping, and traffic optimization.

Rotterdam's UDT [25] is designed to enhance climate resilience, integrating hydrological models and environmental data to improve flood risk management. The system uses high-resolution 3D hydrodynamic models that simulate interactions between surface water and the sewer network, enabling accurate prediction of storm and river flood scenarios. This interactive 3D digital replica combines detailed geospatial data with dynamic simulations to support both real-time decision-making and long-term planning. The architecture leverages open standards and interoperable platforms such as OGC CityGML/CityJSON, WMS/WFS web services, and the European INSPIRE framework to facilitate the visualization and sharing of knowl-

edge among stakeholders, while also serving as a communication interface between policymakers, experts, and citizens.

In Cambridge, a digital twin [163] is being developed to optimize transportation and mobility while more broadly supporting cross-sectoral urban governance. The initiative originates from the Digital Cities for Change (DC2) project, led by the Cambridge Center for Smart Infrastructure and Construction (CSIC), and supported by the Smart Cambridge program of Cambridgeshire County Council. Its core objective is to demonstrate how built environment data and digital tools can improve city planning, management, and the delivery of public services. The first phase (2017–2019), funded by the Center for Digital Built Britain, produced a prototype city-scale model co-designed with local policymakers, used to explore digital transformation scenarios such as remote working and the uptake of electric vehicles. A second phase (2019–2020), funded by Innovate UK, extended the CDT with a 'live experiment' on the Cambridge Biomedical Campus, applying large-scale transport monitoring data (Automatic Number Plate Recognition) to infer travel patterns, travel purposes and socioeconomic profiles of commuters. This experiment tested strategies for combining conventional datasets with emerging big data sources to improve both the quality of the model and its policy relevance, while involving local stakeholders in framing mobility challenges. Beyond its technical components, the Cambridge CDT has also highlighted the importance of governance frameworks, stressing the need to overcome organizational silos, ensure transparent data-driven decision-making, and involve affected communities.

Helsinki's 3D City Model [35] combines semantic 3D city data with real-time environmental and mobility information. This model allows for energy efficiency assessments, noise mapping, and simulations of sustainable urban development scenarios. Unlike traditional models based on point clouds or polygonal geometry, Helsinki employs semantic 3D city models that integrate not only spatial and graphical features but also an ontological structure of thematic classes, attributes and interrelations. Following standards such as CityGML [61], these models allow the fusion of heterogeneous datasets and represent urban objects such as buildings, roads, water bodies, or vegetation, across the aspects of semantics, geometry, topology, and appearance. They serve as a valuable basis for simulations ranging from environmental monitoring to disaster management, with outputs that can be reintegrated into the original model for thematic enrichment. In practice, visualization is supported by the HSY platform and computational models such as REMA, while data governance mecha-

nisms ensure that data owners retain control over publication and service exposure, thus complying with GDPR requirements.

Barcelona's CityOS [8] integrates IoT sensor networks, open data portals, and predictive analytics to support participatory governance, optimize urban logistics, and monitor the environment. The system is based on a robust digital infrastructure that includes over 19,000 active sensors, which collect data on noise, air quality, traffic, energy consumption, water flow, and more. These sensors are coordinated through the open-source Sentilo platform, developed by the Municipal Institute of Informatics (IMI) to promote interoperability and real-time data sharing. CityOS leverages this rich data environment through open data portals that offer transparent access to urban information and dashboards that inform both citizens and developers, supporting real-time decision-making and civic engagement. Predictive analytics further enrich the system, enabling proactive urban solutions in areas such as transportation, energy, and waste, as seen in applications such as sensor-based irrigation management, intelligent traffic light control, and coordinated logistics for urban services.

Another recent example of an urban digital twin is Downtown Dubai [49], developed through the integration of Esri's ArcGIS platform with Unreal Engine 5.1 for high-fidelity visualization. The model is based on GIS datasets, including cadastral maps, 3D building geometries, terrain elevation models, and urban infrastructure layers. This geospatial data are collected within ArcGIS CityEngine, which provides modeling capabilities to generate a semantically rich and scalable 3D environment. The environment is then connected to Unreal Engine's rendering pipeline, producing photorealistic textures, advanced lighting and interactive simulations. This combination allows users to not only visualize the city with a high level of realism but also run scenario-based simulations, such as evaluating new construction projects, verifying emergency response plans, or analyzing pedestrian mobility.

Following these international examples, Italy is also making strides in the field of UDTs, with the Digital Twin project of Bologna standing out as a significant initiative. This project focuses on developing a comprehensive Data Platform, hosted on CINECA's cloud infrastructure and leveraging FBK's DigitalHub, to harmonize data from diverse systems and ensure seamless interaction between urban layers such as energy, transportation, and healthcare. The platform supports the collection, correlation, integration, visualization, and analysis of city data, empowering stakeholders like the Municipality of Bologna and its subsidiaries to efficiently process and analyze data from various sources, including legacy systems, IT platforms, and IoT solutions.

Although this technology is becoming widely used, there is no universal approach or one-size-fits-all model for their implementation. in Italy, for example, the diversity of cities generates as many digital twin models, each adapted to territorial and operational peculiarities. This fragmentation hinder a lack of interoperability between systems, amplifying the challenges related to multilayer integration, i.e., managing complex interactions between different layers of urban systems, such as energy, transportation, and waste management.

2.1.2 Reference Architectures for UDTs

To design new architectures for interoperability and trust in UDTs, it is first necessary to understand how current implementations are conceptually structured. The literature shows that urban digital twins are typically implemented through modular and layered architectures that reflect the complexity of urban environments and the need for scalable and interoperable solutions [67]. Current works present a recurring architectural structure composed of data acquisition layer, integration and management layer, processing and simulation layer and interface and application layer (Fig. 2.1). This layered sequence represents the common architectural core of UDTs, ensuring modularity, interoperability, and adaptability to urban contexts, with specific variations based on the domain and objectives of each project. The data acquisition layer collects information from heterogeneous sources such as IoT devices, traffic sensors, environmental monitors, and participatory citizen data [41]. Lu *et al.* [166] propose integrating BIM, building management systems, and environmental sensors into a single information pipeline for the Cambridge campus, while [144] Schrotter and Hürzeler describe how the foundation of Zurich's digital twin consists of updated 3D models released as open data. Similarly, Boulos *et al.* highlight the role of IoT-based monitoring networks in providing real-time environmental data for their UDT of New Cairo, forming the basis for subsequent integration and modeling [6].

The second layer concerns information integration and management. Here, heterogeneous data are normalized, enriched with metadata, and made interoperable. Ferré-Bigorra *et al.* [53] formalize this step in terms of an "input–processing–output" structure, emphasizing the importance of standardization for the future widespread adoption of UDTs. This is followed by the processing and simulation layer, where data are translated into knowledge through predictive models, simulations, and machine learning techniques. Martella *et al.* [104] emphasize the role of platforms that combine AI/ML with complex urban scenarios, while Herath *et al.* [68] emphasize the need for an edge-cloud continuum to ensure scalable, real-time computing ca-

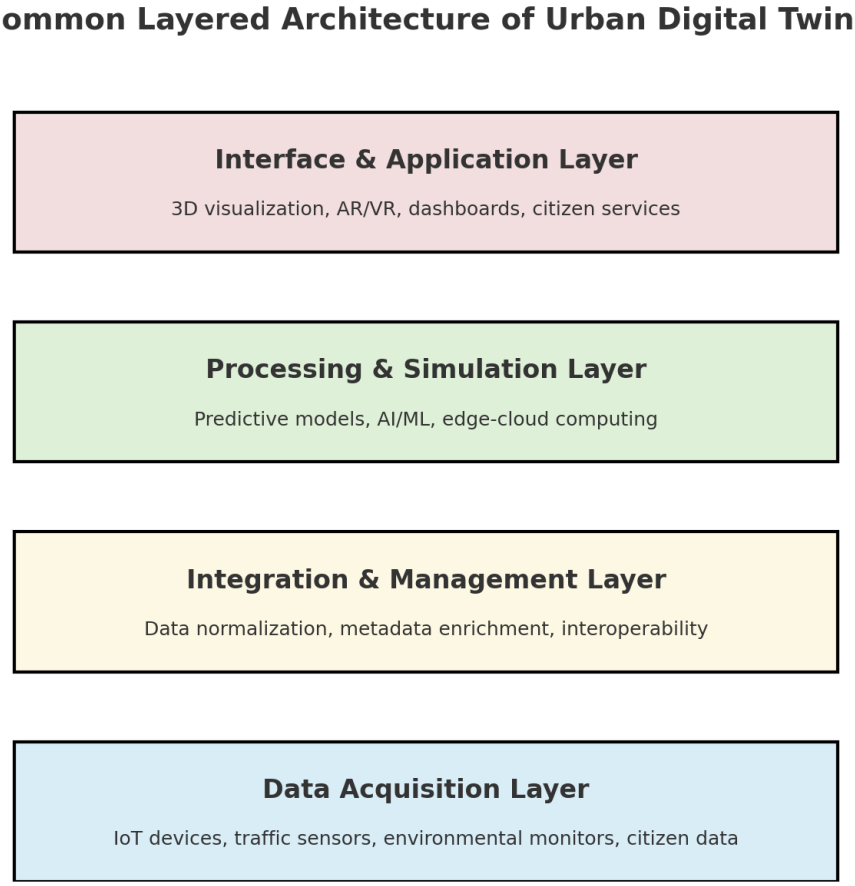


Figure 2.1: Common Layered Architecture of UDTs

pabilities. Finally, the outermost layer is the interfaces and services layer, which includes 3D visualization tools, immersive AR/VR environments, dashboards for public decision-makers, and urban implementation systems. This layer is exemplified in the Cambridge Digital Twin, where Lu *et al.* [163] integrate augmented reality applications for maintenance and asset tracking. Similarly, Herath *et al.* [67] demonstrate how their architecture supports autonomous traffic management in Issy-les-Moulineaux, showcasing the operational potential of interface-level services in real-world urban contexts.

At the European level, the DUET (Digital Urban European Twins) project [87] is an initiative that operationalizes the multi-layered architecture of UDTs for public governance. This project implements a data acquisition and integration layer based on open data portals and IoT feeds, and a simulation layer based on environmental and mobility models. DUET is being implemented in various municipal administra-

tions, including Flanders, Athens, and Pilsen, and enables the assessment of real-time policy scenarios. The platform also incorporates an interface layer designed to promote transparency and citizen participation, offering a dashboard and visualization tools that are intuitive even for non-expert users.

These conceptual layers find concrete realization in existing platforms. For example, FIWARE[55] provides a suite of reusable Generic Enablers, including the Context Broker for managing context data, IoT integration tools, and components for visualization, access control, and monetization. Ubiwhere's Urban Platform extends this model by offering cross-domain analysis, real-time indicators, and sustainability metrics, while other solutions such as [ui!] UrbanPulse and CityOS adopt microservice-based architectures that enable flexible integration with external data sources and services [162, 9].

2.2 Decentralized and Emerging Technologies in UDT Architectures

UDTs are socio-technical ecosystems in which stakeholders (public administrations, companies, citizens, digital services) must interact with mutual trust, transparency, data governance, and auditability. Current solutions like FIWARE or examples like DUET are starting to address these aspects, but they lack a structured guarantee of trust, identity, and verifiability. To address these gaps, recent research has pointed to emerging technologies such as Distribuited Ledger like Blockchain, decentralized identifiers, and verifiable credentials as enablers of secure, interoperable, and trustworthy UDTs.

2.2.1 Distribuited Ledger

DLT are innovative systems designed to improve security and trust between untrusted parties by decentralizing data management. Unlike traditional centralized systems, DLT operate through a peer-to-peer (p2p) network in which each node maintains a synchronized copy of the ledger. This distributed structure promotes transparency and reduces the dependence on intermediaries. A defining characteristic of DLT is their append-only model, which makes the recorded data immutable and resistant to unauthorized alterations. By removing centralized control, DLTs also mitigate the risk of single points of failure, a common vulnerability in traditional systems. To maintain consensus and ensure data integrity, DLTs employ cryptographic protocols that allow participants to collectively validate and agree on transactions. This

decentralized validation mechanism promotes trust and reinforces the security and reliability of the system without the need for a central authority. Most DLTs support the execution of smart contracts, which are computer programs written in general-purpose programming languages that run on DLT. Each node executes the program, and correct execution requires the consensus of all nodes on the result. Smart contracts eliminate the need for intermediaries and enable trustless, transparent, and reliable computation within distributed environments.

BLOCKCHAIN Blockchain is a decentralized and immutable ledger, shared across all nodes of a p2p network, which records every transaction in a transparent and verifiable manner [131]. Unlike traditional databases that rely on a central authority to manage and authenticate entries, blockchain distributes this responsibility among all participants, thereby eliminating the need for trusted intermediaries such as banks, notaries, or centralized servers.

In this technology, operations are grouped into blocks that are cryptographically linked in chronological order, forming an immutable ledger structure commonly known as the 'chain' [182]. Each block contains transaction data and a cryptographic reference to the previous block, creating a sequential dependency that secures the ledger against tampering (Fig. 2.2). The chain begins with the genesis block and any attempt to alter past records would require recalculating all subsequent hashes, a computationally prohibitive task in sufficiently large networks [30].

Going into details, each block consists of a header and a body: the header contains the hash of the previous block, a timestamp marking the precise moment of creation, a nonce (a 32-bit arbitrary number used once), and the Merkle root. The Merkle root is the top hash of a binary Merkle tree constructed from all the transactions contained in the block. This structure allows for efficient transaction verification because, instead of recalculating the validity of each individual transaction, nodes can compare the Merkle root, which changes significantly if even a single transaction is modified. Cryptographic hash functions such as SHA-256 ensure that even minimal changes in input produce entirely different outputs, further securing the integrity of the chain.

Block addition is governed by consensus mechanisms, which ensure that all nodes in the decentralized network agree on the current state of the ledger. One of this mechanisms is known as Proof-of-Work (PoW) [114], where specialized nodes called miners compete to solve a mathematical puzzle. They repeatedly vary the nonce and compute the hash of the block header until they find a value lower than a target

threshold set by the network. This procedure, known as mining, requires substantial computational effort and energy consumption, making it prohibitively expensive for malicious actors to attempt to rewrite the ledger. Once a miner successfully finds a valid hash, it broadcasts the block to the network, where other nodes verify the solution and if correct, append it to their copy of the blockchain. The miner is rewarded with newly minted cryptocurrency and transaction fees, providing economic incentives to secure the network. Despite its robustness, PoW is criticized for its high energy consumption and limited transaction throughput, which has motivated the exploration of alternative consensus protocols.

One such alternative is Proof-of-Stake (PoS) [114], which replaces the energy-intensive competition of mining with a system where validators are chosen to create new blocks based on the amount of cryptocurrency they "stake" or lock as collateral. In PoS systems, validators are randomly selected to propose and attest new blocks, with their probability of selection proportional to their stake. This mechanism drastically reduces energy consumption and allows for faster transaction validation. Furthermore, PoS introduces economic disincentives for malicious behavior, since validators who attempt to manipulate the system risk losing their staked assets. However, concerns remain about the potential for centralization, as large stakeholders may exert disproportionate influence on the network. Nonetheless, PoS represents a significant step toward making blockchain more sustainable and scalable, and and has been adopted by major platforms such as Ethereum.

Beyond simple transaction recording, blockchain technology has been significantly extended through the introduction of smart contracts [157]. In blockchain, a smart contract is a self-executing program stored on the blockchain that automatically enforces the terms of an agreement when predefined conditions are met. Smart contracts are associated with unique blockchain addresses, contain executable code, private storage, and balance, and once deployed, they cannot be altered. They operate autonomously, executing actions such as transferring funds, issuing tokens, updating records, or even deploying other contracts, without human intervention or intermediaries. This automation ensures transparency, reduces the potential for disputes, and minimizes human error or fraud, as the contract's logic is verifiable and publicly accessible on the blockchain. Moreover, smart contracts introduce a new model of trust and instead of relying on an external authority, parties can trust the correctness of the contract's code and the integrity of the blockchain itself.

The deployment of smart contracts has given rise to decentralized applications (dApps) [177], which leverage blockchain's immutability, transparency, and decen-

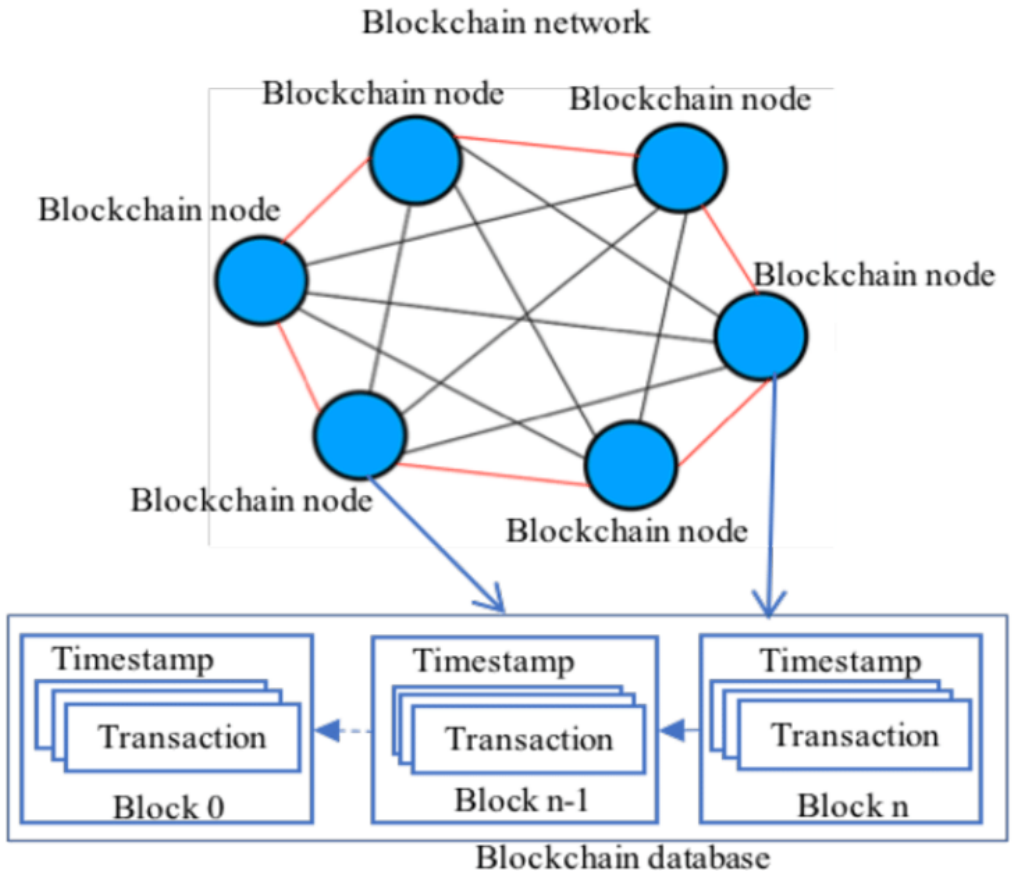


Figure 2.2: Blockchain structure

tralized consensus to build applications ranging from financial services (Decentralized Finance or DeFi) to supply chain management, healthcare, and governance systems. dApps eliminate the need for central servers and can operate globally without downtime, censorship, or unilateral control. By enabling programmable logic on top of secure transaction infrastructure, smart contracts transform blockchain from a passive ledger into an active computational platform.

In the context of UDTs, where multiple stakeholders contribute, update, and exchange information across federated infrastructures, maintaining trust and verifiability without central intermediaries is a key challenge. DLTs such as blockchain provide a decentralized mechanism to ensure data integrity, consensus, and traceability across heterogeneous environments. By enabling collective validation and immutable record-keeping, they establish the technical foundation for secure and auditable data exchanges in distributed urban ecosystems.

2.2.2 Self-Sovereign Identity

As UDTs evolve toward decentralized and federated data ecosystems, identity management becomes a central component for enabling trusted interactions among heterogeneous actors. In the federated model, the entity that effectively "holds" the user's digital identity and the data associated with it is the Identity Provider (IdP). The user is able to employ their digital identity across different services, always "passing through" the IdP, which remains at the center of the model [102]. The IdP issues the digital credential, providing a "single sign-on" experience that can then be seamlessly used elsewhere, reducing the number of separate credentials to manage and thus addressing the problem of multiple identities [13]. However, in this model as well, the user does not have full ownership of their data, which remain under the control of the identity provider. Moreover, it should be considered that major IdPs are among the primary targets of hacker attacks and, therefore, one of the main causes of online identity theft. A new approach to digital identity is introduced with the concept of SSI [145], which aims to return full control of identity to the user. The key objective of Self-Sovereign Identity is to empower data owners with full control over their personal information, allowing them to decide what data to share, with whom, and when. The concept of data sovereignty, pictorially depicted in Fig. 7.2, is typically achieved by using Decentralized Identifiers (DIDs) [4] and Verifiable Credentials (VCs) [2], both recently standardized by the World Wide Web Consortium (W3C).

2.2.3 Decentralized Identifier

A DID is a unique identifier that distinguishes entities within a decentralized system. This identifier is a simple text string composed of three parts:

- the DID URI scheme identifier
- the DID method identifier
- the DID method-specific identifier

Figure 2.3: A simple example of a decentralized identifier (DID)

Each DID is linked to a cryptographic key pair and is recorded in a DID Document, which is stored on a verifiable data registry (Step 1 in Fig.7.2), such as a blockchain. The DID Document contains publicly accessible information, including the public key, which facilitates decentralized identity verification, services and elements that provide a set of mechanisms that allow a DID controller to prove control of the DID. The controller or controllers are the only entities authorized to perform operations on the DID (update, delete, etc.). (Fig 2.4).

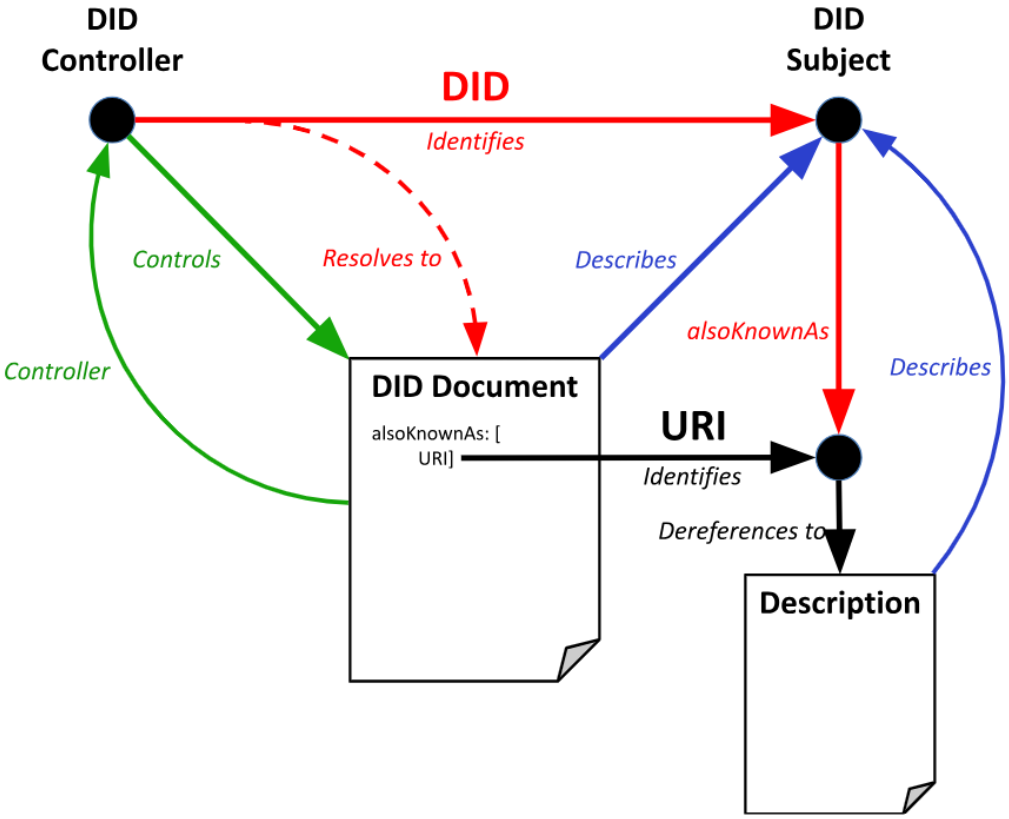


Figure 2.4: A DID is an identifier assigned by a DID controller to refer to a DID subject and resolve to a DID document that describes the DID subject. The DID document is an artifact of DID resolution and not a separate resource distinct from the DID subject

In UDT, DIDs can be used to grant organizations, data services, citizens, or IoT devices control over their own digital identity and the data they produce.For instance, an urban data provider or municipal infrastructure system could use its DID to securely share operational data with other domains within the UDT, while maintaining verifiable ownership and access control.

2.2.4 Verifiable Credentials

VCs are digital credentials which follow the relevant World Wide Web Consortium open standards. They are designed to authenticate and validate an individual's identity, qualifications, or other attributes through a tamper-proof and cryptographically verifiable process. Within the VC framework, trust is established through a well-defined interaction among three main roles: issuers (who create the credential), holders (who store and present it), and verifiers (who check its validity). Each credential contains a set of claims that describe specific attributes or properties. Claims can take different data types, such as string, integer, or boolean, representing attributes like identity, affiliation, or certification status. The holder can choose to share only some of these claims by generating a verifiable presentation (VP). The VP acts as a controlled container that groups one or more authorizations and selectively discloses only the necessary claims. The VP is also cryptographically signed by the holder, so that the verifier can confirm both the authenticity of the original authorization given by the issuer and the legitimate ownership by the holder.

Practically when a VC is issued, it's digitally signed by a trusted authority, such as a government, to attest to specific claims or attributes about an entity (Step 2 in Fig.7.2). It includes references to both the credential holder and issuer (via their DIDs). When sharing a credential, the holder generates a Verifiable Presentation (VP) by signing the VC with their private key (Step 3 in Fig.7.2). The verifier can

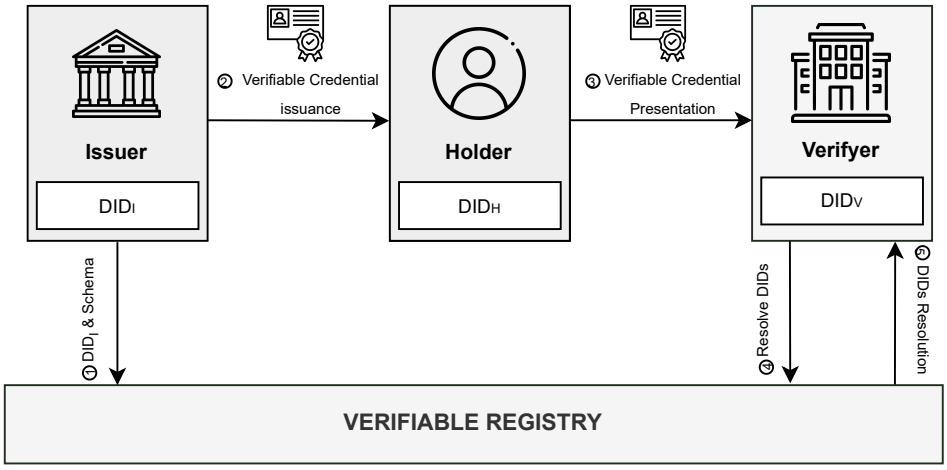


Figure 2.5: SSI Overview

then resolve the referenced DIDs (Step 4 in Fig.7.2) to retrieve the associated DID

Documents (Step 5 in Fig.7.2), which provide the public keys of both the issuer and the holder. Using the issuer's public key, the verifier confirms the credential's authenticity, while the holder's public key ensures ownership of the credential. In particular, because DID Documents are stored on a verifiable registry, verifiers can authenticate credentials without direct interaction with the issuer. Since VCs may contain sensitive information, they are typically shared off-chain. When a VC is compromised or outdated, it is crucial to invalidate it. For this reason numerous revocation mechanisms have been suggested in the existing literature [106]. A widely used mechanism is the *Bitstring Status List* [169], recommended by the W3C Verifiable Credentials Working Group. This specification uses a bitstring-based mechanism, where each bit corresponds to a VC. When a revocation is necessary, the associated bit is set, effectively changing the credential's validity. The default status list can contain up to 131,072 entries (equivalent to 16 KB), making it efficient to manage even for many credentials. Since, in most cases, only a few credentials are revoked, the list can be compressed down to a few hundred bytes using common techniques.

The main advantages of VCs include strong security, ensured by cryptographic proofs that prevent tampering or forgery; enhanced privacy, since verification can occur without revealing unnecessary personal data; and high interoperability, thanks to compliance with open and widely adopted web standards that allow seamless use across diverse organizations and applications.

Within UDT ecosystems, VCs make it possible for heterogeneous entities such as municipal departments, data providers, or IoT systems, to issue and verify claims about the quality, origin, or reliability of data. This decentralized verification process ensures that information circulating across domains remains authentic, auditable, and compliant with shared trust policies.

3 Open Issues in Urban Digital Twin Ecosystems

This section introduces some of the open issues related to the Urban Digital Twin domain, which are particularly critical to enabling secure, interoperable, and trustworthy UDTs. These challenges go beyond purely technical aspects and include semantic, temporal, and organizational dimensions that reflect the inherent complexity of modern urban ecosystems. In particular, this chapter identifies and analyzes four key challenges (Fig. 3.1): heterogeneity and data volume, which affect interoperability and multilayer integration; diachronic complexity, concerning the management of historical data and evolving urban entities; data federations and trustworthiness, focusing on ensuring verifiable data exchange and sovereignty across distributed environments; and federated process governance, addressing transparency, accountability, and control within collaborative computational workflows.

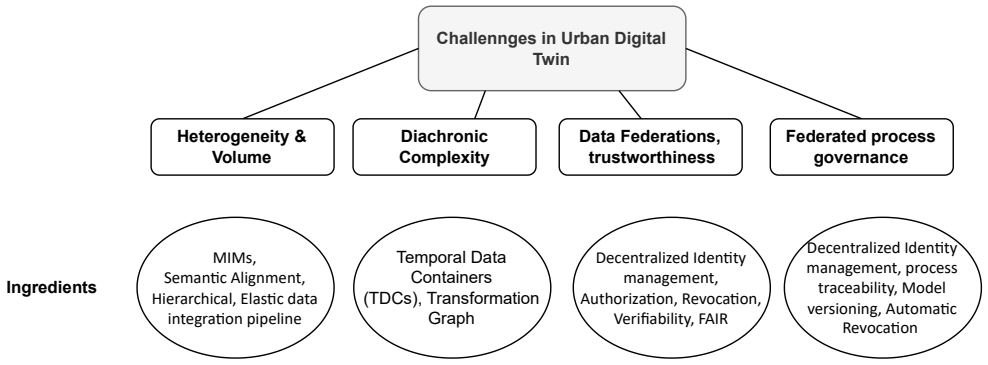


Figure 3.1: Overview of the main challenges addressed in this thesis and the corresponding ingredients proposed to tackle them. The upper part ("Challenges") summarizes the key open issues in UDT research, while the lower part ("Ingredients") lists the conceptual and technological components, such as MIMs, TDCs, and decentralized trust mechanisms,that are employed across the research works presented in this dissertation to tackle these challenges.

3.1 Heterogeneity and Data Volume

UDTs integrate data generated by a wide range of heterogeneous sources, including sensors, legacy information systems, simulation models, and administrative databases. This diversity manifests as differences in data formats and communication protocols, semantic misalignments between ontologies, infrastructural incompatibilities, and fragmented governance across organizational boundaries. These issues make it difficult to create a unified and reliable representation of urban systems, ultimately hindering the effectiveness of UDTs.

Three major aspects exemplify this challenge: (i) the difficulty of achieving interoperability across heterogeneous systems, (ii) the complexity of integrating multi-layered urban subsystems while preserving their interdependencies, (iii) the limited scalability of traditional architectures when exposed to high-volume, high-frequency heterogeneous data streams.

3.1.1 Dimensions of Heterogeneity

UDTs integrate information coming from highly diverse domains and infrastructures. This diversity manifests as syntactic, semantic, technological, and organizational heterogeneity, which collectively hinder the establishment of a coherent and reliable representation of urban systems.

Syntactic heterogeneity refers to differences in data formats, models and communication protocols adopted by various urban subsystems [15]. For instance, IoT devices, administrative databases, and simulation models may use incompatible data schemas such as XML or JSON, or proprietary formats, and rely on distinct communication standards (e.g., MQTT, REST, SOAP). These mismatches hinder the direct data exchange and require additional middleware [127] or translation layers, increasing integration costs and latency. Although syntactic interoperability can be partially achieved through standard data models or APIs, it does not address deeper inconsistencies in meaning or context.

Semantic heterogeneity arises when the same concept is represented differently in diverse datasets or ontologies.[128]. For example, one platform may use the term "domestic energy consumption," while another may refer to "residential consumption," with different units of measurement, time scales, or spatial definitions. Ontologies such as Web Ontology Language (OWL) or Next Generation Service Interface -

Linked Data (NGSI-LD) [1], attempt to bridge these gaps by introducing shared vocabularies and contextual relationships. However, maintaining semantic alignment across evolving datasets and independent actors remains difficult. Differences in domain ontologies, inconsistent metadata practices, and the absence of universal reference standards for data representation lead to semantic drift and ambiguity [134]. Technological heterogeneity concerns the coexistence of multiple, non-compatible infrastructures at the network and platform level. For instance, different communication technologies such as Bluetooth, Wi-Fi, or LoRaWAN (Long Range Wide Area Network) and network configurations result in interoperability barriers between IoT devices and cloud services. Moreover, discrepancies in data collection frequency, storage mechanisms, and processing architectures complicate synchronization across urban layers. While standard network protocols such as TCP/IP enable connectivity, they do not guarantee functional interoperability within complex UDT ecosystems. Organizational heterogeneity extends beyond technology, reflecting the fragmentation of governance structures, policies, and procedures between different stakeholders and systems [64]. Municipal departments, private operators, and national agencies often apply distinct access rules, privacy constraints, and contractual agreements. Even when technical integration is possible, misaligned governance and data-sharing policies prevent effective cooperation. Legal frameworks like the General Data Protection Regulation (GDPR) highlight the need for harmonization, but in practice, responsibilities and liabilities remain distributed across institutional boundaries [139]. Hybrid interoperability frameworks, combining syntactic, semantic, technological, and organizational layers, are emerging to reduce the impact of these forms of heterogeneity. Middleware platforms can perform translation and alignment, while European initiatives such as Gaia-X [159] promote federated governance and standardized interfaces.

Although interoperability standards provide partial solutions, the coexistence of diverse formats, meanings, technologies, and governance frameworks continues to hinder the realization of fully integrated and trustworthy UDTs. While interoperability frameworks attempt to mitigate these discrepancies, achieving seamless data exchange and consistent interpretation across heterogeneous systems remains one of the most persistent challenges in the UDT domain [26, 133, 66].

In addition to differences in format and meaning, data heterogeneity in UDTs also manifests in the scale and dynamics of information exchange [151]. Continuous data

[1] NGSI-LD is a specification developed by ETSI (European Telecommunications Standards Institute) for data management and exchange in the context of smart cities and the Internet of Things (IoT).

streams from sensors, simulation models, and legacy systems generate massive and rapidly evolving datasets that strain storage, computation, and integration capabilities. This amplifies interoperability challenges, as UDTs must harmonize large-scale, asynchronous inputs while preserving reliability and temporal consistency.

3.1.2 Complexities of Multilayer Integration in UDTs

A direct consequence of heterogeneity is the challenge of multilayer integration. Urban data are not only heterogeneous in format, semantics, and governance, but are also distributed across multiple interconnected domains such as energy, mobility, water, waste management, and the built environment. Urban systems are inherently multilayered, exhibiting complex physical, functional, and temporal couplings [11]. Each of these domains operates with distinct technologies, temporal resolutions, and spatial granularities, yet their behaviors are deeply interdependent [126]. Integrating these different layers into a coherent UDT requires the management of cross-domain dependencies while preserving the internal logic, accuracy, and granularity of each subsystem. The main difficulty lies in capturing these interdependencies in a unified framework without oversimplifying individual layers or losing the granularity necessary for accurate analysis and decision-making.

Hierarchical modeling frameworks [95],are commonly adopted to represent urban layers at appropriate abstraction levels while maintaining links to detailed submodels. For example, energy systems might be modeled at a high level as grids or hubs, but with the ability to drill down into finer details, such as individual solar panel outputs or battery storage levels, when needed. However, these hierarchical abstractions inevitably rely on simplifying assumptions that may obscure local behaviors or feedback loops between layers. As a result, aggregated models often struggle to reflect the full causal complexity of urban interactions, reducing their predictive reliability.

Data synchronization and alignment represent another critical difficulty [148]. Different layers generate data at heterogeneous temporal and spatial scales. For instance, transportation systems might generate data in seconds, while water management systems might use hourly or daily data. Integrating such data streams requires techniques like temporal resampling, spatial aggregation, and interpolation to harmonize datasets without losing essential information. Yet, maintaining temporal and semantic consistency across continuously evolving sources remains an unresolved problem, especially when data collection is managed independently across domains.

Cross-layer optimization algorithms aim to coordinate decision-making processes across interdependent urban systems [31]. For example, optimizing energy usage might involve not only balancing supply and demand but also adjusting transportation schedules to reduce peak loads on the grid. However, cross-layer optimization algorithms have to face still open challenges such as computational complexity and conflicting objectives between different domains.

Visualization and interaction tools also play a key role of achieving a comprehensive understanding. Advanced dashboards, 3D city models and immersive visualizations, can help stakeholders navigate the complexity of integrated systems. However, a significant limitation of many current visualization systems is their passive nature. They often display multilayer interactions but do not allow users to directly interact with or manipulate the data. This lack of interactivity restricts the ability to test scenarios, explore dynamic responses, or implement real-time interventions, thereby reducing the practical utility of the insights provided.

Further complexity is given by coordination across neighbor operators that manage complex digital twin infrastructures with specific integration formats. The absence of a de-facto industry standard for cross-domain integration continues to hinder interoperability across digital twins, increasing technical complexity and organizational fragmentation.

3.1.3 Beyond the State of the Art of existing Models in Modular and Semantic Interoperability

3.1.3.1 Middleware and Modular Architectures for Interoperability

Research on interoperability across digital twin and smart city platforms remains limited, with most existing works addressing domain-specific integration rather than cross-platform interaction. For example, Farhan *et al.* [52] propose a Semantic Interoperability Middleware Architecture (SIMB-IoT) for big data in heterogeneous IoT infrastructures to improve interoperability between health devices and medical systems. The paper describes the components and functionalities of SIMB-IoT, which uses semantic annotation, ontology mapping, and cloud services to achieve data integration. However, the approach is limited to semantic aspects, lacking consideration of syntactic interoperability and security issues, and it is validated only on static datasets rather than real-time healthcare scenarios.

Similarly, Akanbi *et al.* [10] present a Semantic Interoperability Middleware Architecture (SIMA) for heterogeneous environmental data sources in the context of drought monitoring. Their framework integrates data from sensor devices and local

indigenous knowledge using the RDF data model, yet it remains largely conceptual and unvalidated beyond test mode.

Aziz *et al.* [16] compare and analyze three data integration models for heterogeneous industrial systems, Event-Driven Architecture (EDA), Service-Oriented Architecture (SOA), and the Arrowhead framework. The comparison is qualitative and based on proof-of-concept implementations, without quantitative benchmarks or large-scale deployment, thus remaining confined to conceptual evaluation within the industrial domain.

Although most of these approaches remain domain-specific, a gradual shift can be observed toward more modular and semantically enriched architectures. Badawi *et al.* [18] move the focus toward the urban domain, developing a unified model (DT-DNA) that enables data and information sharing across cities. Their framework, based on the ISO 37120 standard, ensures semantic interoperability and facilitates cross-city comparison of services. Nonetheless, the proof-of-concept remains limited to standardized datasets from two cities, without testing on real-time or heterogeneous data.

More recently, the DUET project introduced a practical approach to interoperability through its DUET-Cell architecture, which supports the integration of new data sources, simulation models, and visualization clients. Data from different sources are mapped to a shared ontology and stored in a knowledge graph, while APIs and message streaming enable real-time exchange and discoverability via a federated Data Catalog.

Overall, while a tendency toward modular, ontology-driven integration can be observed, current solutions still fall short in achieving full scalability, standard compliance, and cross-platform interoperability. Building upon these observations, this work proposes a practical layered architecture and a concrete technological stack designed as foundational building blocks for UDT implementations. The proposed architecture adopts a microservice-oriented design and a serverless processing engine capable of scaling computation across a continuum of resources, while ensuring MIM compliance through customized data processing pipelines.

3.1.3.2 Semantic Standards and Ontology-based Interoperability

The modular and microservice-based architecture outlined above establishes the structural foundation for scalable and interoperable UDT implementations. Building upon this foundation, the next contribution extends the interoperability paradigm beyond the technical layer, introducing semantic mechanisms that ensure shared

meaning, consistency, and reasoning across heterogeneous domains. This subsection therefore focuses on ontology-based and standards-driven approaches that complement the architectural framework by enabling knowledge-level interoperability within complex urban ecosystems.

In this context, the increasing complexity of UDT ecosystems has made interoperability and multilayered integration central challenges in smart city research [14]. While several platforms and standards have emerged to address these issues, they often fall short when confronted with the semantic heterogeneity and structural fragmentation of real-world urban systems [133].

Among the most prominent initiatives, FIWARE promotes interoperability through the NGSI-LD standard for context data exchange. Although NGSI-LD offers a structured and extensible model, its integration with external platforms, such as robotic operating environments (e.g., ROS 2), frequently requires complex data transformations and schema mappings. These operations introduce significant maintenance overhead and limit scalability, especially when dealing with legacy infrastructures or evolving standards [165, 7, 86].

Urban modeling platforms such as UrbanSim [170] and CityZenith [113] further exemplify these limitations. UrbanSim integrates land use, transportation, and economic data, but its reliance on domain-specific datasets complicates cross-platform interoperability. CityZenith offers advanced 3D visualization tools, but remains constrained in its ability to support inter-city collaboration and integrated planning.

Ontologies and semantic frameworks, such as OWL, RDF play a crucial role in enabling meaningful data exchange. However, the fragmented nature of urban systems, each governed by distinct schemas and terminologies, often impedes semantic alignment. Integrating multiple ontologies demands significant effort, particularly when harmonizing real-time data streams with static semantic models [79]. Standards like CityGML aim to support 3D urban model interoperability, yet practical implementations frequently encounter inconsistencies due to divergent ontological assumptions [27].

In summary, while existing platforms and standards provide valuable foundations, they remain insufficient to address the full spectrum of interoperability challenges in UDTs. Most approaches emphasize syntactic and technical integration, neglecting the semantic and structural coherence required for consistent interpretation and reasoning across heterogeneous urban domains.

To overcome these limitations, an extension of the GlassBox model introduces a semantic interoperability layer that generates a dynamic ontology by explicitly defin-

ing relations, dependencies, and governing rules among urban entities. Through this relation-driven ontology construction, the extension enables cross-domain reasoning, traceability, and adaptive alignment between data, models, and processes. Unlike conventional data-exchange frameworks, it establishes a unified semantic backbone that contextualizes information and integrates heterogeneous sources transparently, turning interoperability from a merely technical procedure into a knowledge-driven process.

3.2 Diachronic Complexity

UDTs integrate real-time and historical data across urban domains allowing decision-makers to assess current conditions, test future scenarios, and coordinate urban interventions. However, most current UDT implementations prioritize real-time data collection and visualization, often overlooking the management of longitudinal information [153]. The term longitudinal refers to the ability to represent and analyze the evolution of spatial entities over time, going beyond static snapshot to capture how urban elements change over time. As a result, many existing systems provide effective monitoring tools based on instantaneous states but lack the semantic infrastructure required to model historical transformations, such as modifications in administrative boundaries, infrastructure reconfigurations, or jurisdictional shifts [63].

Diachronic complexity is further increased by the coexistence of multiple temporal scales within the urban systems [124]. Real-time sensor feeds, monthly mobility patterns and multi-decade infrastructure records, often coexist within the same digital environment but they are rarely aligned in a semantically consistent way. This misalignment hinders cross-scale reasoning, making it difficult to trace the evolution of urban entities, correlate short-term dynamics with long-term trends or support temporally informed decision-making. A key challenge lies in the absence of mechanisms for synchronizing datasets that differ in granularity, update frequency, and historical depth [92]. For instance, traffic data may be updated every few seconds, while zoning plans or cadastral maps may reflect changes only every few years. Without a shared temporal framework, integrating these sources risks producing fragmented or contradictory representations of urban reality.

3.2.1 Beyond the State of the Art in Addressing Diachronic Complexity

Existing platforms, including CityGML [33], 3D Tiles [32], and Urban Pulse [109], provide advanced visualizations of dynamic data streams but rely on essentially static city representations. This synchronic focus limits their ability to capture diachronic complexity, namely the structured representation of how spatial entities evolve over time through processes such as splits, mergers, renamings, or jurisdictional reclassifications [63].

Semantic technologies have been widely adopted to improve interoperability in the urban domain.

Ontologies and knowledge graphs are used to describe entities, characteristics, and relationships in datasets, ensuring both human interpretability and machine reasoning [94]. They represent a fundamental step toward semantic integration, but existing approaches often differ in scope, expressiveness, and their ability to manage temporal and contextual dynamics. For instance, Zhao *et al.* [184] proposed Urban Knowledge Graphs as a tool for semantic alignment based on shared vocabularies and entity linkage. While this model effectively connects heterogeneous urban datasets through common ontological references, it focuses primarily on static relationships and lacks mechanisms to represent temporal evolution or data provenance, which are essential for dynamic urban processes. The GIVA architecture [40] relies on formal geospatial ontologies to provide a semantic infrastructure for spatio-temporal integration. Although it demonstrates the potential of ontology-driven models for managing geographic entities, it adopts a domain-specific approach centered on spatial reasoning. Its temporal component remains limited to timestamped observations, without addressing diachronic reconstruction or causal dependencies among evolving entities. The Data Polygamy system [34] leverages topological and spatial reasoning to uncover relationships among polygonal datasets. This approach is effective in detecting correlations and spatial overlaps across data layers, yet it operates at the analytical rather than semantic level.

In summary, these contributions illustrate the evolution of ontology-based and graph-based approaches toward urban data integration, yet they still fall short in unifying spatial, semantic, and temporal dimensions within a coherent reasoning framework. Addressing these gaps requires models capable of representing not only relationships and alignments but also the historical trajectories and interdependencies of urban entities over time.

Accurately interpreting historical and spatio-temporal datasets requires not only temporal representation but also a solid understanding of lineage, provenance, and quality. Over the past years, several initiatives have addressed these aspects from different perspectives.

The Frictionless Data initiative [160] promotes lightweight standards for packaging and describing datasets, while complementary tools within the same framework enable the representation and replication of data transformations. However, its approach remains primarily syntactic and structural: it focuses on schema description and validation but lacks semantic expressiveness and reasoning capabilities. Consequently, it facilitates data sharing and reproducibility but does not support the contextual interpretation or cross-domain alignment required for complex urban datasets.

Enterprise platforms such as Open Metadata [123] and IBM Data Lineage [71] extend these ideas by offering visualizations of data flows, documenting processing steps, and ensuring regulatory compliance through auditability and traceability. These solutions provide static lineage visualization rather than semantically grounded provenance models, and therefore cannot capture evolving entity relationships or support automated reasoning about data dependencies over time.

In the geospatial domain, projects like *Open History Map* [111] experiment with versioning and community-driven documentation of territorial changes, focusing on source reliability and precision. Yet, these approaches often rely on manual curation and lack formal semantic integration, making it difficult to interlink geographic evolutions with administrative, infrastructural, or social dimensions. As a result, they remain valuable for documenting spatial change but are not suited to automated, large-scale reasoning about diachronic phenomena.

Collectively, these approaches remain mostly centered on metadata modeling or visualization of transformation pipelines, with little integration into operational semantic frameworks. In particular, they rarely support the explicit encoding of transformation rules (e.g., splits, mergers, renamings, reclassifications), reasoning over incomplete or inconsistent temporal sources, or systematic assessment of dataset quality across spatial, temporal, and semantic dimensions. These limitations are especially critical in longitudinal applications such as historical reconstruction, multi-year urban planning, or policy simulation, where decisions depend not only on the origin of data but also on its fitness for temporal analysis.

To address the limitation, namely the lack of formal mechanisms for representing transformations and for assessing the quality and lineage of evolving datasets,

this dissertation introduces a semantic–temporal element (TDCs). This component extends the GlassBox architecture by embedding temporal semantics and temporal transformation logic directly within data structures. In doing so, it enables the automated reconstruction of longitudinal datasets, formal reasoning over entity evolution, and rule-based validation of temporal consistency, thereby overcoming the structural and operational limitations identified in existing systems.

3.3 Data Federations and Trustworthiness

UDTs collect data from different urban subdomains, each managed by actors with their own rules and responsibilities. Centralizing this fragmented landscape is neither feasible nor desirable. Instead, UDTs rely on federated data architectures, where datasets remain under the control of their providers and are integrated on demand to support cross-domain analytics and real-time decision-making [58].

Within these federated ecosystems, however, trustworthiness becomes essential, as participants must be able to verify the authenticity, integrity, and compliance of shared information to guarantee accountability, sovereignty, and resilience against adversarial behavior.

Traditional trust models, typically based on Public Key Infrastructures (PKIs), centralized certification authorities, or informal reputational mechanisms were originally conceived for stable and hierarchical systems with well-known actors. These models don't account for behavioral changes or adversarial interference, and they introduced central points of failure and non-trasparency trust chains [47, 150].

To overcome these limitations, several proposals have explored multidimensional trust modeling. Ion et al. [91] support the idea of multidimensional trust by combining social and contextual signals in peer-to-peer settings, while the TIDE framework [129] incorporates time, context, and feedback reputation.

Beyond conceptual limitations, trustworthiness in UDTs is also challenged by concrete vulnerabilities affecting the entire data lifecycle. Attacks on data integrity, such as data poisoning, model poisoning, or cache pollution [161, 178], compromise the reliability of simulations and decision-making. Risks of desynchronization between twins, inconsistencies in replicated models and weaknesses in data backup further erode confidence [90].

Authentication and access control are equally problematic. UDTs are exposed to impersonation, unauthorized access, rogue devices, and privilege escalation [173]. Multiple studies have demonstrated systemic weaknesses across IoT, OT, and IT

infrastructures, including impersonation attacks, credential reuse, and unauthorized access due to flawed revocation mechanisms or plaintext transmission of credentials [75, 44].
While various mitigation strategies have been proposed, from anomaly-based intrusion detection and radio-frequency fingerprinting to fine-grained access control and credential analytics, these solutions often remain domain-specific and lack interoperability across federated urban infrastructures. The handling of personal and mission-critical data from different urban domains introduces additional ethical and legal dimensions. These flows must comply with governance frameworks that ensure transparency, consent, and the protection of individual rights such as the General Data Protection Regulation (GDPR) [172] in Europe while remaining adaptable to sector-specific and jurisdictional requirements.
Addressing these vulnerabilities requires a trust architecture that supports decentralized identity, verifiable access control, and dynamic revocation, in line with the FAIR principles of Findability, Accessibility, Interoperability, and Reusability [174].

3.3.1 Beyond the State of the Art in Federated and Decentralized Data Sharing

Recent years have seen the emergence of several blockchain-based and decentralized data sharing platforms aiming to enhance transparency and sovereignty in data exchange. These systems provide valuable insights into the design of federated architectures for Urban Digital Twins, where trust and verifiability are critical to ensuring reliable collaboration among autonomous actors.

FAST [43] represents an IoT data marketplace in which users are authenticated through DIDs, while blockchain ensures the registration of available data streams, identity verification, and payment settlement. Although the system ensures fair trading and privacy-preserving data delivery, its focus remains confined to the IIoT domain, lacking general mechanisms for verifiable data provenance, cross-domain interoperability, and decentralized policy enforcement.

Similarly, Sober *et al.* [154] propose a blockchain-based IoT data marketplace that employs smart contracts to regulate transactions and participation between producers and consumers.Their design, however, remains a proof of concept centered on transactional logic and it does not address verifiable identity, decentralized trust management, or legal compliance (e.g., auditability and data erasure), thereby limiting its applicability to federated or large-scale ecosystems.

Missier *et al.* [110] introduce a decentralized infrastructure for fair and trusted IoT data trading, where blockchain ensures transparency in the flow of data streams from IoT devices to value-added services. While pioneering the idea of granular metering and automated contract enforcement, their architecture still relies on trusted brokers for traffic mediation, which introduces a single point of trust and prevents fully decentralized operation.

Bajoudah *et al.* [19] also present a decentralized architecture for IoT data exchange that relies on Ethereum-based smart contracts to ensure integrity and settlement, although their focus remains primarily on data delivery rather than verifiable storage.

Ramachandran *et al.* [136] outline a distributed data space model for smart cities leveraging DLTs, where the blockchain stores metadata while off-chain storage solutions such as IPFS manage the actual data assets. DEON [116] offers a decentralized marketplace for various IoT applications, using IPFS for off-chain storage and DIDs with VCs for identity and access management. However, DEON focuses on general-purpose network decentralization and provides only high-level specifications.

Bernabé-Rodríguez *et al.* [23] combine blockchain with secure multi-party computation (SMPC) to guarantee both transparency and privacy-preserving data processing, while Yoon *et al.* [183] introduce a DID- and VC-based personal data exchange framework using Hyperledger Fabric to ensure identity sovereignty. Despite these advances, most existing systems remain limited in scope. They address specific aspects such as identity management, transactional integrity, or privacy, but seldom integrate them into coherent trust architectures suitable for large-scale federated ecosystems.

Overall, existing solutions demonstrate substantial progress toward decentralized and transparent data sharing. However, they remain constrained by fragmented architectures, limited semantic interoperability, and the absence of verifiable trust mechanisms aligned with SSI principles.

To overcome these limitations, this dissertation introduces VESPACE, a verifiable data space architecture designed to ensure trustworthy and sovereign data exchange across federated urban ecosystems. VESPACE integrates SSI principles by assigning each participant a DID and managing access through VCs that attest dataset ownership, certification, and usage rights.

Data access and authorization events are anchored on blockchain to provide tamper-evident auditability, while datasets remain stored off-chain in decentralized repositories. This separation ensures both transparency and GDPR-compliant revocability. Access is dynamically controlled through credential-based policies that define

purpose, scope, and duration, enabling fine-grained and revocable sharing across multiple organizations.

Finally, semantic policy enforcement allows these mechanisms to operate coherently across heterogeneous domains, linking data provenance, identity, and governance into a single interoperable framework. In doing so, VESPACE overcomes the fragmentation and limited verifiability of existing blockchain-based data marketplaces, providing a foundation for trustworthy and auditable urban data federations.

3.4 Federated Process Governance

As UDTs evolve into decentralized ecosystems, not only data but also the workflows that rely on them such as simulations, predictive analytics, and federated learning, must remain transparent, traceable, and accountable across autonomous domains. This shift moves the focus from trust in data to trust in the processes that continuously transform, aggregate, and act upon data.

Traditional governance models, based on static permissions or centralized control, are unsuited to dynamic federated environments. They lack mechanisms to ensure end-to-end traceability or to isolate and revoke corrupted contributions once detected. Consequently, federated processes risk being compromised by contaminated models, biased data, or opportunistic behavior, with limited means to monitor, explain, or correct their effects [147].

Several recent studies have embedded verifiability and accountability directly into the process layer. Marchioro *et al.* [103] propose blockchain-based provenance tracking for process transparency, while Cai *et al.* [28] and Mugunthan *et al.* [112] introduce accountability mechanisms to evaluate and detect malicious contributions. Yang *et al.* [181] complement these approaches with reputation-based consensus and incentive schemes to discourage opportunistic behavior. However, these solutions remain fragmented because they typically record isolated transactions or model updates but do not connect data, models, and actors through verifiable identities, nor do they implement automated revocation to neutralize the influence of corrupted components.

3.4.1 Beyond the State of the Art in Trustworthy Federated Digital Twins

FL has emerged as a crucial enabler for distributed intelligence in DT ecosystems and hence UDTs, offering collaborative training without exposing raw data [81, 97, 155,

117, 118]. Several works combine FL with DLTs to improve transparency and coordination. The authors in [98] introduce the concept of DT Wireless Networks (DTWN), advocating the integration dt into wireless network infrastructures to enable real-time data processing and computation at the edge. They propose a blockchain-enhanced FL framework operating within the DTWN to support collaborative intelligence. The blockchain serves as a tamper-proof ledger to record training models and to enforce the permission control of participants in the process. Similarly, in [96], the authors present a permissioned blockchain-basedFLscheme for DT Edge Networks (DITENs), incorporating asynchronous aggregation and spectrum-aware scheduling to improve communication efficiency and data privacy. In this framework, the blockchain functions as a coordination layer, facilitating a secure and transparent FL process. The work in [76] proposes an approach for model update verification in decentralizedDT edge networks using a Directed Acyclic Graph (DAG)-based DLT and a double auction mechanism to incentivize participation. Moreover, the work in [132] introduces FedTwin, an adaptive asynchronous FL paradigm that employs a custom Proof-of-Federalism (PoF) consensus protocol, differential privacy, and falsified model detection to enable robust DT network collaboration. Finally, the authors in [12] explore a blockchain-assisted hierarchical FL platform for Industry 4.0 applications, integrating dt into Cyber-Physical Systems (CPS) and employing a two-stage aggregation mechanism to improve scalability and accuracy. The blockchain is used to verify and validate trained models by leveraging designated validation nodes.

The works presented primarily leverage blockchain to track model updates, ensure data immutability, or facilitate secure model aggregation. However, they fail to provide a comprehensive solution for enhancing trustworthiness in FL processes within DT systems. Specifically, they lack mechanisms to track the entire lifecycle of contributions by linking them to unique identifiers, such as DID, similar to the proposal of this chapter. Furthermore, they do not support the inclusion of detailed metadata such as data provenance and collection context, which is essential for auditing and trust. In addition, they fail to estimate the influence of individual client updates or data on the global model, which is critical for assessing contribution value and potential risk. Most notably, these approaches lack revocation mechanisms enabling the system to revoke and reduce the impact of faulty or malicious updates or models. TrustFlow, introduced in this dissertation, extends beyond these limitations, integrating SSI and VCs to authenticate entities and link every data contribution and model update to a persistent, verifiable identity. All updates are semantically annotated and anchored on a blockchain ledger, creating a continuous chain of prove-

nance from data origin to model outcome. An embedded influence estimation module quantifies the contribution of each participant to the global model's performance, supporting adaptive weighting and policy-based revocation of corrupted updates via on-chain smart contracts.

3.5 Beyond the State of the Art: An Integrated Contribution Map

This section provides an integrated perspective on how the five contributions collectively advance UDT ecosystems toward interoperability, temporal coherence, sovereignty, and process-level trust. As illustrated in Figure 3.2, the proposed framework forms a vertically layered pipeline that connects the raw heterogeneity of urban data sources to the governance of federated computational processes and their outcomes.

At the foundation, **MIMs** establish the execution substrate for modular interoperability, defining lightweight interfaces and serverless pipelines that enable distributed urban systems to exchange data and services in a standardized manner. On top of this structural layer, the **GlassBox model** reinterprets interoperability as a semantic property of the digital twin infrastructure itself, introducing an ontological backbone that explicitly links units, resources, maps, and rules. This ensures that heterogeneous components are not only connected but also contextually aligned and interpretable across domains.

Building upon this semantic foundation, the **TDCs** introduce diachronic reasoning and temporal continuity. They extend the static semantic model with mechanisms for versioning, transformation tracking, and snapshot reconstruction, thereby enabling longitudinal consistency and reproducibility in urban analyses, an essential capability for understanding how cities evolve over time and for maintaining the traceability of historical decisions.

The subsequent layer, represented by **VESPACE**, shifts the focus from semantic and temporal integration to verifiable and sovereign data exchange across federations. By embedding SSI, DID, and VC mechanisms into the data-sharing workflow, VESPACE ensures that every transaction, access request, and policy evaluation is cryptographically verifiable, auditable, and revocable. This component bridges interoperability and trust, transforming distributed data spaces into accountable ecosystems of collaboration.

Finally, **TrustFlow** extends these principles to the process layer, where data are no longer the sole object of verification: the computational workflows themselves,

such as federated learning or simulation, become accountable. Through blockchain-anchored provenance, influence estimation, and policy-based revocation, TrustFlow provides federated governance mechanisms that ensure fairness, resilience, and transparency in cross-organizational intelligence processes.

Together, these five components constitute a coherent progression from raw data heterogeneity to governed, trustworthy decision-making. The layered design depicted in Figure 3.2 embodies a unified UDT pipeline in which interoperability, semantics, temporal reasoning, data sovereignty, and federated trust form an integrated continuum bridging technical, semantic, and organizational boundaries toward the creation of reliable, sovereign, and verifiable urban intelligence.

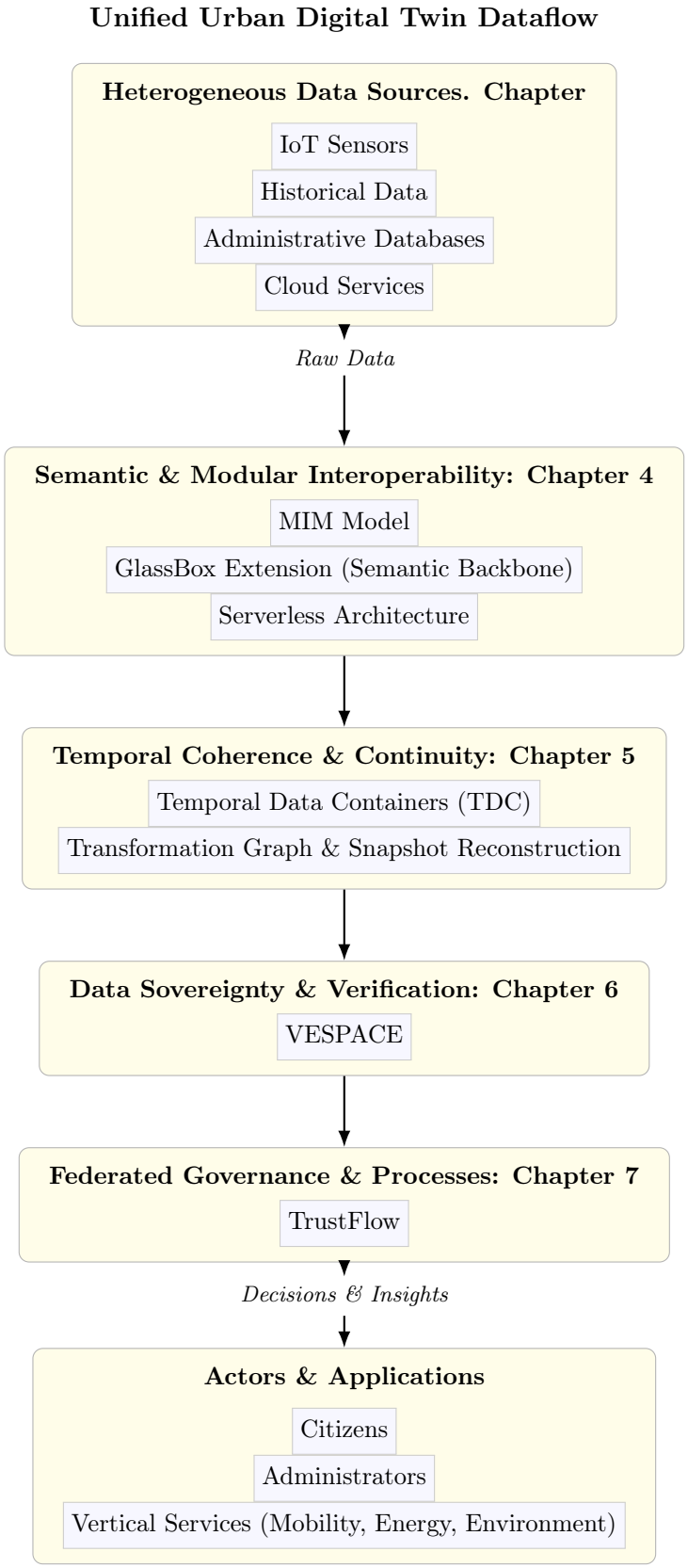


Figure 3.2: Unified view of the Urban Digital Twin ecosystem: from heterogeneous data sources to semantic interoperability (MIM + GlassBox), temporal coherence (TDC), data sovereignty (VESPACE), and federated governance (TrustFlow), leading to actionable insights for citizens and vertical urban services.

4 Architectural Approaches for Interoperable Urban Digital Twins

This chapter describes how the design of modular architectures and the use of interoperability mechanisms contributes to the construction of reliable and flexible UDTs. Modern cities can be understood as "systems of systems," encompassing transportation, energy, environmental monitoring, and many other infrastructures. Each subsystem generates large volumes of data, which differ not only in format but also in temporal resolution and semantic granularity. For example, mobility systems operate with real-time, high-frequency data streams, energy systems work with data at aggregated time scales, and data pertaining to the environment is collected and analyzed at yet another rhythm.

This heterogeneity and scale make the integration of subsystems particularly challenging, highlighting the need for modular architectural frameworks capable of harmonizing diverse sources, supporting interoperability, and enabling consistent cross-domain cooperation within UDT ecosystems [108]. Traditional centralized solutions are often rigid and vendor-dependent, and they struggle to adapt to dynamic data flows and heterogeneous infrastructures. To overcome these limitations, UDTs require an architectural foundation that is both technically elastic and semantically coherent.

To this end, the work presented in this dissertation proposes a progressive approach combining standard-driven and model-driven mechanisms. The first focuses on technical and operational interoperability, through a multi-faceted interoperability model based on Minimal Interoperability Mechanisms (MIMs) and serverless computing, providing a lightweight and vendor-neutral baseline for elastic data processing and system interaction. The second explores semantic and structural interoperability, extending the GlassBox paradigm as a flexible backbone for multi-layer integration and cross-domain alignment. Through component-based ontological design, metrics, and rule-based synchronization, the GlassBox framework supports coherent data and

model integration across heterogeneous subsystems, offering a unifying conceptual layer for UDT ecosystems.

4.1 A Multi-faceted Interoperability Model for Urban Digital Twins

MIMs [120] are a set of sufficient capabilities to share, use, and reuse data across systems. MIMs aim to address key challenges such as consistent data models, access and usage rules, protection of rights, transparency, and location data management. Implementing a traditional architecture with minimal interoperability mechanisms can result in a rigid, inflexible system that is unable to adapt to changing requirements or integrate with new systems. This can also impact scalability, making it difficult to manage an increasing volume of data from various sources.

Within this framework, a secure layered architecture and corresponding software prototype are introduced, promoting a multi-faceted interoperability model based on MIMs and the serverless processing paradigm. The MIMs framework addresses multiple dimensions of data interoperability and compliance, encompassing data modeling, communication protocols, exchange formats, semantic alignment, and context management. This comprehensive approach at the data and communication layers ensures that information can be exchanged and interpreted consistently across heterogeneous systems. The integration of the serverless paradigm enhances the processing layer by enabling the definition of composable and reusable data pipelines, consisting of chained transformation functions that can be seamlessly deployed across distributed resources, while maintaining MIM compliance both within vertically specialized components and across platforms.

4.1.1 MIMs for Data Interoperability

MIMs are vendor-neutral and technology-agnostic, meaning that anybody can use them and integrate them into existing systems and offerings. The MIM standard includes several levels, currently accounting for six formally defined MIMs, partially depicted in Figure 4.1, including Context Information Management, Shared Data Models, etc. The subject of MIM1 is the context that provides comprehensive status information about real-world entities and enables access to various data sources and analysis of the information for event detection. A context management API and data models are necessary for accessing the information, which is available in a catalogue that supports different types of interoperable models. Without loss of

generality, NGSI-LD [21] can be readily adopted to provide an API for managing and requesting context information, while Orion-LD [54] can act as a Context Broker for context data management that supports the NGSI-LD API. MIM2 provides a set of guidelines and a catalogue of common data models for cross-vertical cooperation, such as smart cities and communities, smart agri-food, smart utilities, etc. Following the prior example, the NGSI-LD compliant data models for smart cities have been defined by organizations and projects, including FIWARE, GSMA[62], and the SynchroniCity project [156]. Existing data models and ontologies, e.g., the SAREF [1] (Smart Applications REFerence ontology) standard by ETSI/oneM2M [119], can be mapped for use with NGSI-LD by identifying the entities, properties, and relationships that can be managed and requested by the NGSI-LD API. The implementation of a MIM-compliant architecture typically involves the design as a monolithic architecture or as a microservice-based one. A monolithic architecture would involve ensuring MIM compliance via single, cohesive business logic units, bundling all the interoperability functionalities and data transformations together as part of the functional logic. This approach can limit the scalability, flexibility, and independent evolution of the MIM functionalities. On the other side, in a microservice-based architecture, MIM compliance can be developed and enforced by tailored data pipelines responsible for specific data transformations. This approach allows for modular development, independent scalability, and easier maintenance of the interoperability components. This proposal embraces this design philosophy and goes a step further by relying on the serverless paradigm, allowing one to define customized lightweight functional units whose execution can be chained together to reach a desired compliance/objective.

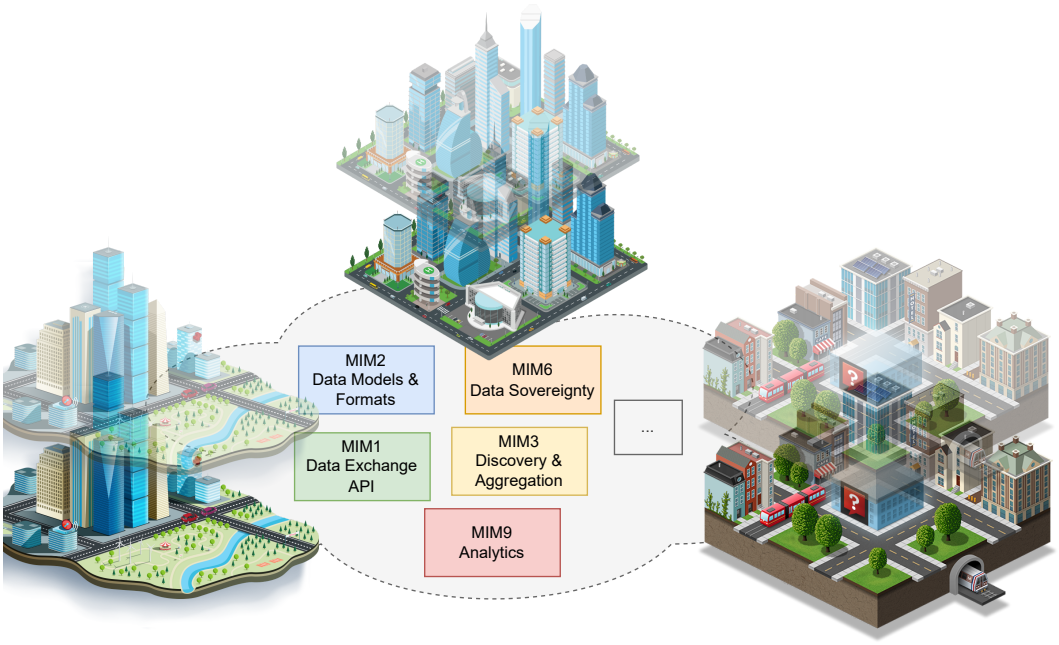


Figure 4.1: Illustration of a networked ecosystem of Urban Digital Twins where seamless cooperating and sharing is ensured via multi-faceted MIMs.

4.1.2 Serverless Computing

The integration of serverless computing presents numerous advantages, including the ability to accommodate diverse requirements, facilitate fine-grained scalability, expedite development processes, and harness the potential of heterogeneous and distributed resources while abstracting their complexities [45].

In this context, Function-as-a-Service (FaaS) represents a novel paradigm in cloud computing where code, representing customer business logic, is dynamically executed in response to incoming events. A FaaS platform follows the serverless computing model, which shields users from infrastructure details and management tasks, focusing solely on the creation of business logic functionality. It should be noted that customers have no control over the specific instantiation and execution location of the function.

Consequently, the programming model resulting from FaaS is inherently stateless, as subsequent invocations of the same function may be executed in different environments, and allocated resources may be reclaimed after function termination. This one-to-one mapping between functions and triggering events enables FaaS platforms

to achieve finer-grained scalability, ensuring that computational resources allocated for user requests match the incoming load at any given moment.

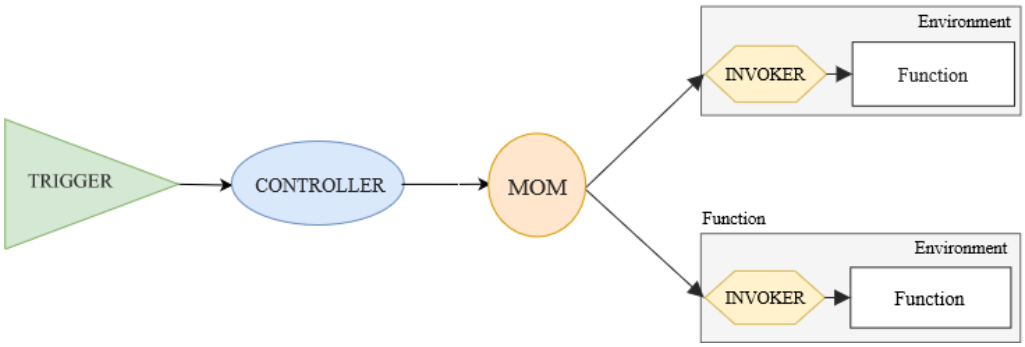


Figure 4.2: High-level FaaS architectural approach. In a *MoM-based* setting, a middleware decouples the controller and the function, whereas in the *direct invocation* scheme the function invocation is enacted by the controller.

Figure 4.2 depicts the typical components of a FaaS platform. The first component is the trigger, which directly interfaces with incoming events such as end-user requests. The trigger receives external events from various sources, potentially using different protocols, and converts them into local events for the FaaS platform. These events are then managed by a controller, which forwards them to the appropriate function based on user-defined configuration parameters. Overall, the controller handles the lifecycle management of functions. The function is the fundamental execution unit in FaaS, encapsulating the business logic required to process specific events. It consists of an environment (e.g., Java), an invoker, and the business code. The invoker, also known as the watchdog, receives events, injects the end-users deployed business code, and subsequently launches the function for execution. The code always executes within a dedicated environment that includes all necessary dependencies, such as system libraries. Mainstream FaaS platforms implement the direct function invocation scheme using either a client/server pattern or a publish/subscribe approach, leveraging a Message-Oriented Middleware (MoM) as an additional component in the architecture (see Figure 2). Utilizing a publish/subscribe scheme allows the controller to offload certain responsibilities, such as load balancing and event delivery to the MoM.

4.1.3 Architecture

A practical approach is proposed to tackle the problem of integrating data, security, and processing platforms within the Digital Twin was proposed. The idea is to make

use of the serverless paradigm across the IoT-Edge-Cloud continuum. In essence, the proposed architecture can be summarized into four functional layers: i) the Adaptation Layer handling the basic functionality of data and service ingestion/integration, ii) the Data Management Layer exploiting raw data coming from the Adaptation Layer, memorizing it in a convenient format, applying configurable syntactic and/or semantic transformations, iii) the Processing Engine embodying streaming analytics and batch processing capabilities in a serverless, event-driven fashion, and (iv) vertical Security Layer tasked with lightweight authentication of IoT data sources and the implementation of a decentralized data access mechanism exploiting DLTs.

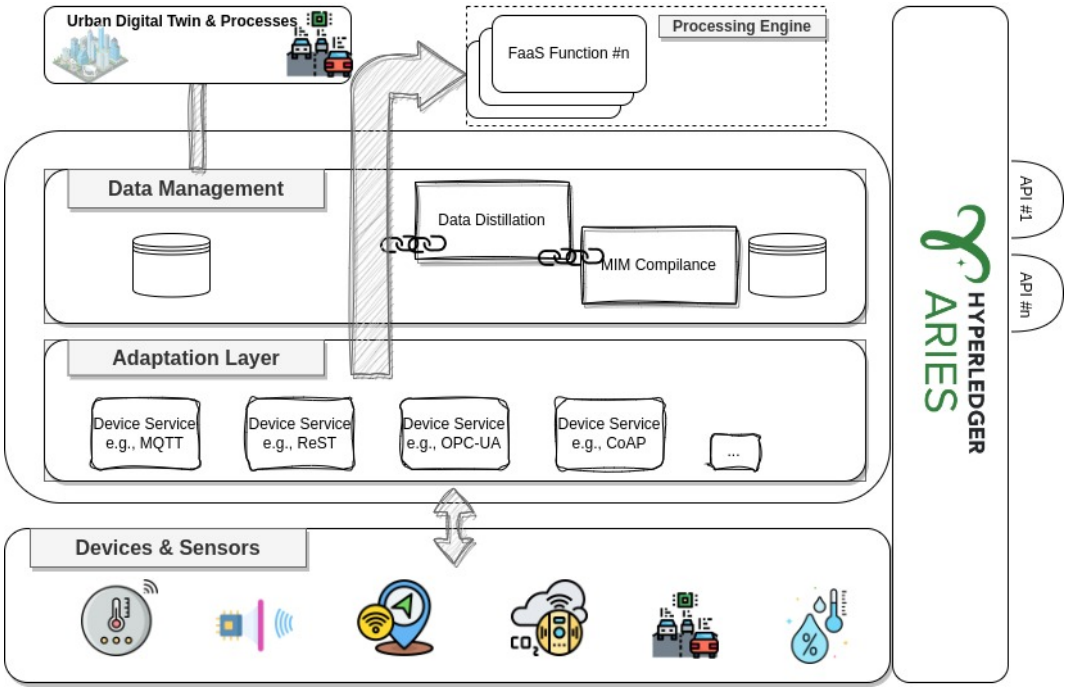


Figure 4.3: Serverless Architecture for Interoperable Urban Digital Twins Data Sharing: enabling scalable, flexible, and cost-effective data exchange.

The Adaptation Layer is the first layer facing the physical and digital smart city world and interacting with it in order to extract information. This layer has the duty of abstracting to the rest of the platform the many complexities of smart city assets, derived also by the exploitation of heterogeneous protocols and formats. This layer is in practice implemented by relying on the EdgeX framework, an open-source framework designed to facilitate the development and deployment of edge computing solutions [93]. EdgeX addresses the complexities associated with edge computing by providing a vendor-neutral platform that fosters interoperability and collaboration

among different hardware and software components. It offers a set of microservices that can be deployed on edge devices, enabling them to collect, process, and analyze data in a distributed manner. The framework includes various modules, such as device services, data management, and data ingestion, which work together to create a comprehensive edge computing ecosystem.

Data collected from the Adaptation Layer could be locally stored and/or further processed by performing lightweight semantic and/or syntactic transformations. Here, data are enriched with time and spatial information, e.g., a timestamp denoting the time the data are acquired by the gateway - in case the attribute is missing from the sensed raw data - or spatial, point-wise information derived from the gateways' location. Data at the edge are ephemeral, stored temporally according to configurable retention policies, and are used to feed the Data Management and/or the Processing Engine. In the Data Management layer also reside the data processing pipelines, used to infer relationships among objects which are stored on different technologies such as graph databases.

The Processing Engine has the capability of performing both batch and stream processing of (raw) data sourced from the Data Management layer and/or streamed from the Adaptation Layer.This capability is achieved through the integration of a FaaS platform, which enables the expression and implementation of complex workflows as compositions of simpler computational units [5]. Thanks to this computing model, the proposed approach can associate with (a stream of) data or events, a single function or a composition of functions used to transform and/or analyze its information content.The composition capability of the FaaS processing engine enables the straightforward augmentation of existing analytics processes with new services, which can be created by recomposing existing functions or by adding only the necessary business logic. Moreover, from a cost perspective, this model enables the activation of an arbitrary number of workflows that are executed only when needed by the platform without any costs incurred during idle times.

This solution advocates the use of MIMs as an abstraction to access standardized data, addressing various aspects of interoperability such as data modeling, data formats, semantic understanding, context management, and security. Delving deeper into the data semantic part, EdgeX managed devices are qualified resources allowing to express knowledge about what the sensor is measuring, its units of measure, its location, and accuracy without any direct relation to the overall context for the application(s) it feeds. The Adaptation Layer (EdgeX managed data sources) device profiles can be expressed or enriched via the JavaScript Object Notation (JSON).

The core data model specifies the required information and can be supplemented by optional extensions. JSON-LD is an RDF notation based on JSON. JSON-LD 1.1 is a W3C Recommendation from July 2020, defining a mechanism for mapping JSON resources into RDF graphs via "@context" definitions for translating JSON strings to RDF identifiers [168]. These definitions may be given explicitly within JSON resources or indirectly via links to external definitions. In principle, the addition of "@context" at the Adaptation Layer profiles enables a straightforward mapping to RDF, and a natural means to support semantic enablement.
When dealing with highly distributed scenarios, the problem of verifiable data sources is paramount.To this end, an access control mechanism based on the concept of Decentralized Identifiers is incorporated into the proposal. DIDs have a wide range of applications in smart city scenarios, including self-sovereign identity, supply chain management, digital credentials, IoT device management, secure communication, and more. Their flexibility makes them suitable for various applications where decentralized and secure identity management across many smart city actors and devices is needed. Thanks to DIDs each source of information and service manipulating data can be universally identified and verified by all other components and partners of the extended ecosystem.To address these requirements, Hyperledger Aries is adopted, an open-source project developed under the Hyperledger Foundation that aims to provide a set of tools, protocols, and components for building decentralized identity applications and solutions [56]. The integration of Aries through Aries Agents components developed in each layer of this architecture enables interoperable and secure peer-to-peer interactions using DIDs and verifiable credentials. In practice, Aries agents act as proxies among services and data sources, mediating interaction among architecture components.

4.1.4 Experimental Assessment

This section evaluates the scalability of a concrete technological stack implementing elements of the layered architecture. The focus is on the system's ability to scale gracefully under sudden variations in ingress data load, while maintaining operational continuity. Without loss of generality, two complementary approaches are identified to address scalability: (i) computation offloading to a nearby, more capable managed edge deployment, and (ii) load balancing at the ingress layer. The evaluation of Scenario #1 is conducted on a real testbed depicted in Figure 4.4.

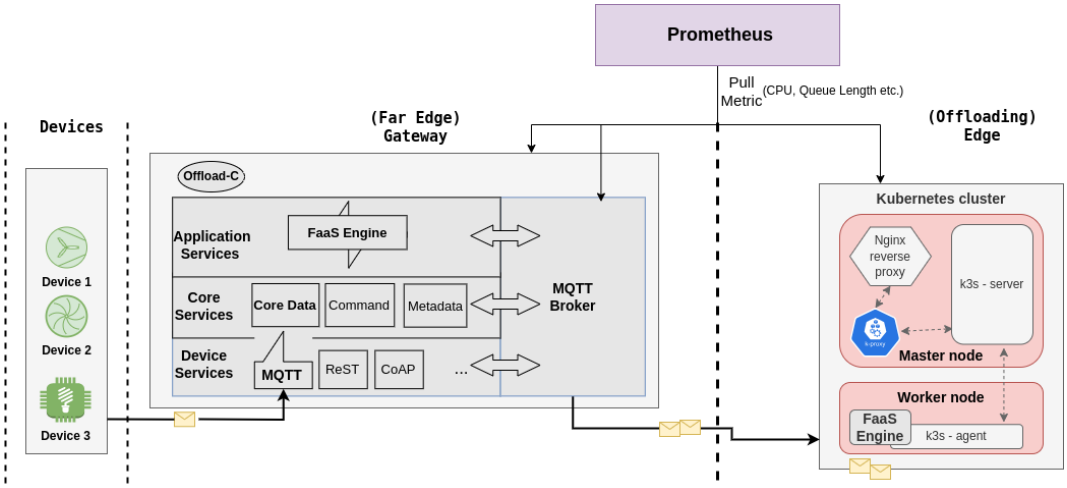


Figure 4.4: Deployment used to assess the scalability of processing engine, whereby the Gateway deployment is subjected to an increase in the traffic load.

Experimental Settings The deployment architecture in Figure 4.4 depicts a cloud-native deployment scenario, consisting of state-of-the-art technological blocks aimed at the management and orchestration of resources, monitoring, and scaling of loosely coupled microservices depending on a metric of interest.On the left-hand side, EdgeX-managed devices (sensors) generate data towards the gateway, which is exposed through an MQTT endpoint. The gateway hosts an all-in-one EdgeX instance consisting of a device service (MQTT ingress service), Core Data service which in this scenario does not persist data, that is the far edge deployment is stateless and data are simply forwarded towards the FaaS Processing Engine. The engine enacts the following compositional logic: (i) performs a complex mathematical function with data as an input, taking on average 500ms (Gateway) and 100ms (Edge) to be computed, and (ii) formats (wraps) the resulting data point in a JSON NGSI compliant format, forwarding it to upper layer services, which persists the MIM-compliant data enriched with context information. Depending on the capabilities of the node hosting the function, operations might prove resource-consuming, delaying ingress messages, and resulting in queues building up in the MQTT broker infrastructure. This is unwanted behavior, and it is up to the monitoring plane to detect the phenomenon, triggering a predetermined control logic.

On the right-hand side, is the edge deployment consisting of a Kubernetes (K3s) cluster comprised of 2 VMs: one master (driver) VM and 1 worker, the latter an eligible candidate where parts of the computation are offloaded. The environments export host-level metrics (e.g., CPU, RAM, etc.) and component-centric ones (i.e.,

MQTT queue length) via dedicated exporters to a Prometheus instance, installed on a dedicated machine. The monitoring plane enacts the offloading control logic, consisting of a re-configuration operation of the data path, entailing the following high-level operations: (i) reduce the ingress data flow on the Gateway FaaS Engine instance, (ii) cold starting a containerized function on the K3s cluster, configuring the ingress data path, and (iii) subscribing the latter instance to the device data topic. This rewiring operation requires some time to take place, and the time required for this operation is dominated by the operation at point (ii) which takes on average 3.5s to complete.

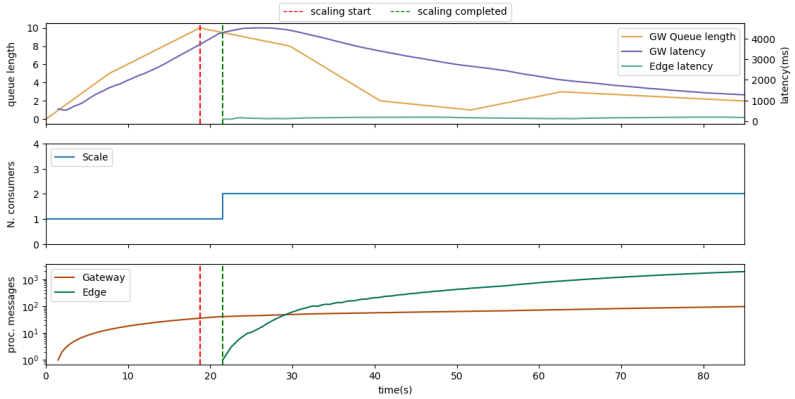


Figure 4.5: Processing Engine Scalability Analysis.

4.1.5 Discussion

In the experiment shown in Figure 4.5, an increasing traffic load is generated at the Gateway, starting from two sensed values per second and doubling every second, until reaching the threshold of 16 sensed values per second. Initially, the FaaS Engine instance on the Gateway node subscribes to all sensor (A) topic partitions and the output of the individual computation is published on a dedicated topic (B) where a (sink) service resides. The first graph shows the message latency calculated as the absolute difference between the instant of sending the message on topic A and the instant of arrival of the relevant processing on topic B. The Gateway latency (GW latency) remains constant in the first phase, around 500 ms, but as the load increases, data values get queued up, and the corresponding end latency increases over time. This is due to the FaaS Engine (subscriber) inability to keep up with the pace of incoming messages. This phenomenon, that is queues building up and contributing to an increase in latency, is observed by the monitoring plane, currently

configured to enact the datapath rewriting logic when >=10 queued messages are observed (dotted red vertical line). Once the control logic is enacted, it takes on average 3.5s to bootstrap the FaaS function pipeline on the K3s cluster (dotted green vertical line), which upon activation, begins the processing of parts of the message stream (A/[1-4]). This offloading procedure contributes to lower the end latency of the Gateway stream, which oscillates in the interval [500ms, 1200ms], while the newly added Edge stream remains stable around 100ms with few fluctuations. The difference between the latency of the edge vs. the gateway stream is to be attributed to the capabilities of the two nodes, the Edge having more computational power when compared to the Gateway node. In the second graph, one can observe the total number of active FaaS function instances running on the infrastructure concerning the progress of the experiment, while the third graph shows the cumulative number of messages processed over time on the Gateway and the Edge node since the start of the experiment. It is noteworthy to point out that a different control logic could be conceived, e.g., based on the measured latency metric. At the same time, a different datapath rewiring logic could be enacted, e.g., all the data stream is offloaded at the edge. In this current scenario, both the Gateway and the Edge nodes are active and processing messages.

4.2 A Semantic and Modular Backbone for Urban Digital Twins

While the adoption of MIMs provides a foundational baseline for interoperability within UDT ecosystems, these mechanisms alone cannot fully capture the semantic, structural, and cross-domain intricacies of urban environments. These limitations become especially evident when dealing with heterogeneous, multi-layered urban systems that require not only syntactic compatibility but also dynamic semantic coherence across domains. To address these constraints, the second contribution of this dissertation introduces a complementary modeling paradigm based on the GlassBox framework. This approach redefines the city as a multilayered network of interacting entities, enabling modular abstraction, semantic enrichment, and cross-layer synchronization. By extending GlassBox beyond its original simulation scope, the model provides a flexible backbone for integrating both real-time and historical data, supporting scalable and semantically aware UDT architectures.

4.2.1 GlassBox model

The limitations of current interoperability frameworks highlight the need for modeling paradigms capable of capturing the layered complexity of urban system. One particularly interesting approach is the GlassBox simulation engine, originally developed by Maxis in 2011 for city-building games like SimCity [105]. Presented at Game Developer Conference 2012 it displayed an innovative and general-purpose approach to the simulation environment. Designed to provide a rich and dynamic simulation experience, GlassBox stands out for its ability to manage large volumes of data and create highly reactive and interconnected virtual systems in a 2013 technological infrastructure. With its flexible architecture, the engine was conceived not only to support the iconic city-building game but also a variety of other simulation titles, making it a versatile tool for creating complex interactive experiences.

The core of GlassBox operations is the detailed management of resources, units, maps, networks, agents, and rules (Fig. 4.6). Resources, such as oil, electricity, wood, and water, are fundamental elements within the game, managed through containers called "bins" that track their quantity and distribution. Each unit, representing entities such as houses or factories, interacts with resources through the rules that describe the needs and conditions specific to each unit in the game as well as its production. These needs and resources move along specific networks. The simulation is further enriched by the presence of maps, which represent environmental variables

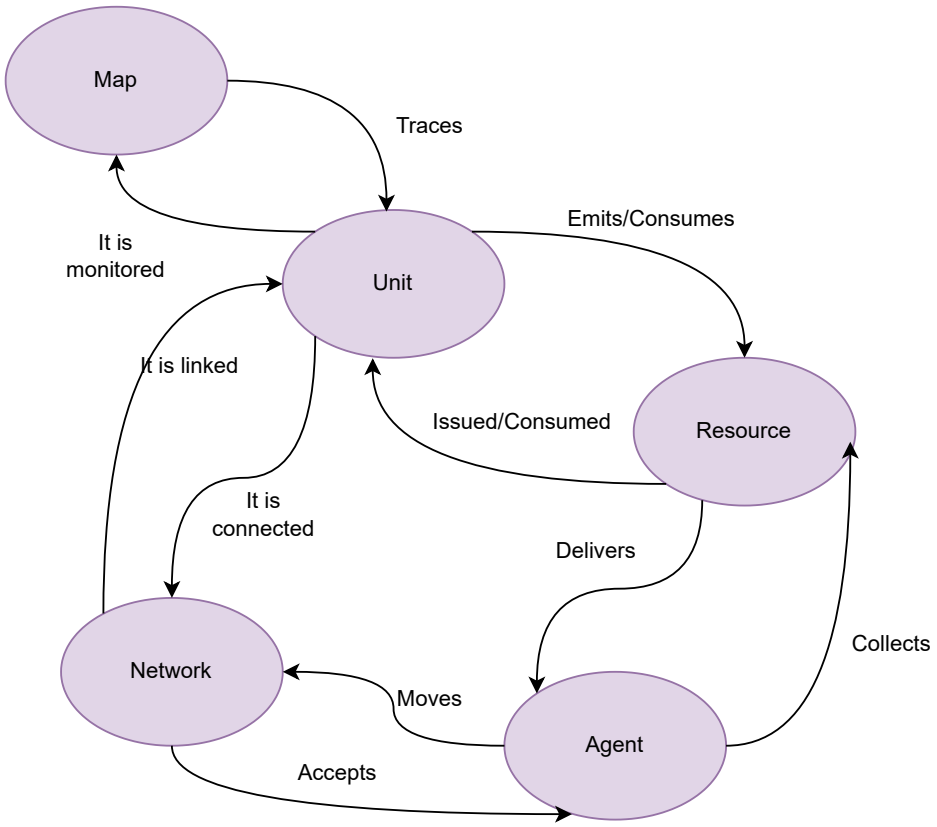


Figure 4.6: GlassBox Model

such as the availability of natural resources, pollution, and land desirability, in general factors that influence the decisions of agents and the evolution of the virtual system.

One of the distinctive features of GlassBox is its ability to easily adapt to new game dynamics, thanks to a system that manages behavior rules defined by customizable scripts. These rules allow for accurate simulation of resource transfer and transformation, creating a complex and interconnected ecosystem where every action has a tangible impact on the game environment. This approach, which integrates real-time data management with the simulation of emergent behaviors, enables players to interact with a virtual world that dynamically responds to their choices.

4.2.2 An Original Proposal for Extending the GlassBox Model

To build a semantic, modular backbone for Urban Digital Twins, the GlassBox model is adopted and extended. While GlassBox was designed to act in a simulation environment where players could never "rewind" and see what a different decision would have brought, a UDT has to cover both simulation-related aspects as well as in-the-field flows of monitoring data, which cannot be presumed to be in real-time but are often aggregated over non-homogeneous time periods.

For these reasons, the GlassBox-extended model uses the same concepts (resources, units, maps, networks, agents and rules) but adds the "metrics" aspect to them, defining snapshots of the various elements valid for specific timeframes. For example, a unit is not just an entity associated with a given set of resources and a given set of networks, but also defines a metric, a value and a timeframe for which that value is valid. The same happens with the maps: the global situation is defined by temporal and spatial aggregations that represent the current state of the system. Thus, maps are not limited to describing geographic locations or the physical distribution of resources, but incorporate a set of temporal metrics associated with each node in the network. For example, an energy map might include data on generation capacities, consumption flows, and storage levels, each with values valid for specific time intervals, such as hours, days, or weeks.

Similarly, networks are no longer static, but dynamic, updated through real-time data streams or periodic aggregates. Each link in a network is enriched with parameters that define the context of interactions, such as maximum transport capacity, energy losses, or latency times. These parameters are constantly reassessed in light of systemic changes, allowing more realistic simulations and prediction of bottlenecks or inefficiencies.

Building upon this dynamic nature, the model conceptualizes the network as not just a representation of connections and flows, but as a modular framework that can operate at multiple scales. A critical advantage of this approach is its ability to support cross-city interoperability. Using the inherent flexibility of the network model, entire systems, such as energy grids or transportation networks, can be abstracted into a single node capable of emitting and receiving resources. This abstraction enables seamless integration between urban systems of varying complexity or granularity, fostering scalability and collaboration across different cities or regions. At the same time, the model retains the capacity to analyze and optimize interactions within each subsystem, maintaining a balance between global and local perspectives. To support this dual-level functionality, the model incorporates global and local metrics

as key elements for capturing and analyzing system states. Global metrics provide a macroscopic view by summarizing the overall state of the system at regular intervals, forming the basis for historical analysis and validation of simulation results. Local metrics, in contrast, detail the specific states of subsystems or individual nodes, such as buildings or electric vehicle charging stations, enabling precise optimization and decision-making at a granular level.

Finally, the concept of "rules" is expanded to integrate more complex conditional logics that govern interactions among system components. These rules define not only the flow of data between entities but also incorporate temporal constraints and critical thresholds, reflecting both urban realities and simulation requirements. For instance, a rule might specify that during peaks in energy demand, certain urban sectors should be prioritized for energy distribution based on dynamically calculated criticality metrics.

4.2.3 Architecture

In order to define a flexible architecture that can be helpful in a complex data landscape like the UDT, it is important to leverage both adaptable data description systems as well as structured data publishing endpoints. Figure 4.7 illustrates this architecture, which is structured into four main layers:

- **Presentation layer**:This layer is responsible for interfacing with external systems and users, ensuring that data are both ingested and accessed in a standardized and meaningful way. It is divided into two sub-layers:
  - **Semantic Ingestion layer**: this sub-layer focuses on the semantic description of datasets and data streams entering the platform.
  - **Unified Data Access layer**: this sub-layer provides standardized access to data for external systems or users. It ensures that data output is consistent, secure, and accessible via well-defined APIs or protocols.

  The Semantic Ingestion Layer enriches incoming data with metadata and semantic descriptions, while the Unified Data Access Layer uses this information to provide consistent and structured access via standard APIs and protocols.

- **Application layer**: this layer is the core of data processing, where ETL (Extraction, Transformation, and Load) operations are defined and executed. In addition,there are modules for planning and coordinating workflows, making sure that ETL processes are executed at the right time and in the right order;

- **Data layer**:this layer is responsible for managing, storing, and accessing raw and transformed data. Its main function is to ensure that information is organized in a scalable, secure and efficient manner, supporting analysis, simulation and visualization operations.

- **Integration layer**: this is a vertical cross-cutting layer that spans across all other layers. It is designed to facilitate advanced data operations, such as simulations, evaluations, and integrations with external systems. More specifically, it is responsible for elaborating the rules to apply in simulations and advanced data operations. The simulation context depends on the descriptions of the rules and semantic metadata from the Semantic Ingestion Layer, on raw and processed data from the Data Layer, and on transformation logic and processing workflows from the Application Layer. Once the simulation context is ready, the Integration Layer communicates with the Unified Data Access Layer to ensure that the processed data and results are made accessible to external systems or users in a standardized and secure way. Moreover, this layer monitors and tracks the flow of data and the operations performed in the various layers.

4.2.4 Implementation

Following the structure of the previously defined architecture, the core of the infrastructure is a MediaWiki installation with a semantic module, which enables the use of semantically enhanced templates for creating data descriptors. This forms the foundation of the Semantic Ingestion layer within the Presentation layer, providing tools for non-technical operators to describe datasets in various formats. The templates facilitate the correct definition of the many descriptive aspects of the model, enabling both internal and external descriptor definitions. In addition to the definition of the models, the tool also collects the rules for the simulation itself.

The Unified Data Access layer, also part of the Presentation layer, ensures that data are exposed through standard APIs and formats, making information reusable and integrable into common libraries and tools. Examples include the publication of maps as Web Map Service (WMS) raster layers, allowing navigation of complex datasets as simple visual layers, while maintaining flexibility for reuse in new, complex visualizations. Real-time calculations and updates are made available via MQTT, supporting both 2D and 3D visualizations of simulated items.

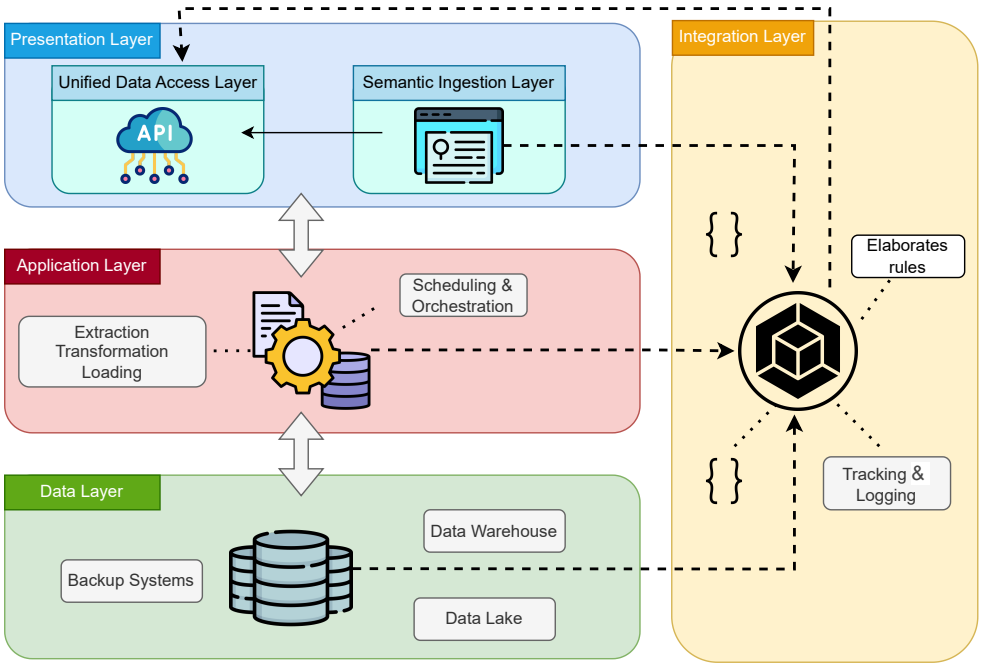


Figure 4.7: Architecture of the extended GlassBox Model

The Application layer is where workflows for data ingestion, transformation, and distribution are managed. Tools such as OpenMetadata and CKAN [36] (Comprehensive Knowledge Archive Network), operating within this layer, enable seamless internal and external descriptor definitions. OpenMetadata supports internal use cases, while CKAN facilitates external data sharing. These tools ensure that data ingestion and management processes are streamlined, supporting a wide range of applications and user needs.

The Data layer is responsible for raw data storage. MinIO, an Amazon S3 replacement, is used in this layer to store temporary datasets. Geographic data are stored for example in the netCDF format, enabling efficient management and retrieval of structured data, while purely tabular data are stored in parquet format. This layer provides the foundational infrastructure required for handling large volumes of data, supporting both simulation and visualization processes.

The Integration layer handles backend calculations and the simulation engine, enabling complex data evaluations. Libraries such as pandas, geopandas, and movingpandas help processing and analyzing data, ensuring that disconnected simulations can be visualized in various formats through the Unified Data Access layer. The sim-

ulation itself is based on a custom developed engine, relying on the original concepts of GlassBox but enabling the tracing of the metrics. This is important for the quality analysis of the model as well as of the collected data: a discrepancy of the metrics could have origin in a wrong description of the dataset, but also in an incomplete coverage of the data available in the platform. This approach decouples the simulation engine from the visualization process, enabling flexible visual representation, whether through 3D analysis tools or simpler 2D maps for broader accessibility.

Finally, the infrastructure also supports the dynamic growth of an emerging ontology for the city. As new datasets are integrated and new requirements arise, the ontology evolves to interconnect data over time, progressively enriching the city's description and supporting an adaptable data ecosystem.

4.2.4.1 Use Case

Millions of people move around the city every day for work, study or leisure using buses, trains, shared bikes, e-scooters, and private cars. The use case under analysis is mobility, which is one of the core infrastructures of the city and a key application domain for Urban Digital Twins. This use case focuses on how urban mobility systems interact dynamically with other urban layers, such as energy and environmental factors. Using the extended GlassBox model, the elements of the use case can be described with precision. Units are all nodes that interact with the networks. Examples include:

- **Traffic Spire**: Sensors located along roads that monitor the flow of vehicles in real-time. Each Traffic Spire is a unit that collects data on vehicles and records them as resources (number of vehicles).
- **Charging Stations**: Units that provide power to EV. The associated resource is the electrical capacity available for charging.

Each of these units interacts with the networks through defined rules, contributing to data collection and analysis. For example, Traffic Spires measure vehicle flow and serve as input to air quality calculations. Similarly, EV chargers interact with the electric grid to monitor energy consumption and demand patterns.

Resources tracked include:

- **Car (Car.carbon and Car.electric)**: Each car is a numeric entity that is tracked through the road network. The entry of a car into a unit, such as a Traffic Spire, is recorded as an incoming resource.

- **Electricity**: The capacity of charging stations and of the electric network is limited.

Maps are visualizations that represent these resources, providing a comprehensive view of the entire urban system.

- **Air Quality Map**: Shows pollution levels in different areas of the city.
- **Noise Map**: Shows noise pollution levels in different areas of the city.

Networks in this context represent the set of connections between units that collect, process, and share information or resources within the city. The networks involved include

- **Road network**: Representing the infrastructure for vehicles, pedestrians, bicycles, and public transport.
- **Electric grid**: Supporting electric vehicle (EV) infrastructure, including charging stations and energy distribution.

Municipality open or internal data sets are often structured as CSV files in various formats, which can be normalized according to the Extended GlassBox Model, as described in Table 4.1.

Table 4.1: Definition of a Traffic Spire csv file - Unit.

Field	**Type**	**Description**
ID	String	Identifier for the spire
Position	GeoPoint	Coordinates of the spire
Time	Timestamp	Timestamp of the beginning or end of the collected metric
Vehicles	Integer	# of vehicles sensed by the device in the last valid timeslot
Accuracy	float	% of validity of the collected metric (Vehicles)

Table 4.2 represents the csv file structure for the data definition of a static sensor-based Air Quality Map.

In a typical smart city scenario, Traffic Spires along the urban roadways detect an increase in traffic.

Table 4.2: Definition of an air quality csv file - Map.

Field	Type	Description
Area	GeoPolygon	Area for the data collection
Station	GeoPoint	Location of the station
StationName	String	Name of the station
Pollutant	String	Identifier of the measure
Time	Timestamp	Timestamp of the beginning or end of the collected metric
Value	float	# measure of the pollutant during the valid timeslot

Listing 4.1 shows a practical representation of of a rule applied to the Traffic Spire unit. As a car (via) enters the area monitored by a Traffic Spire, it is immediately detected by the sensor (appliesTo). The Traffic Spire records the number of vehicles passing through (log resource: Vehicles and Vehicle.{type}), along with the timestamp for each passage (-timestamp=Time). This means that as the vehicle moves through the monitored area, it is counted as an entering resource (local Car.{type} in 1) and contributes to the overall traffic flow data. Once the vehicle enters the area, it is "destroyed" in the sense that its data are finalized and recorded as an exiting resource, which helps calculate the impact on the traffic conditions at that moment. A new vehicle is generated and sent as output of the node itself (agent Car.{type} out 1).

Listing 4.1: Unit Rule Code Example - Spire Trace

```
unitRule spireTrace
    appliesTo TrafficSpire
    via Car
    using RoadNetwork
    local Car.{type} in 1
    agent Car.{type} out 1
    log -timestamp=Time Vehicles 1
    log -timestamp=Time Vehicles.{type} 1
end
```

At the same time, air quality monitoring stations report a rise in PM2.5 levels, measuring safety limits. Listing 4.2 represents the rule applied to a single agent during its lifetime defining that every 10 timeslots (repeatAfter 10) if the agent contains a carbon-fueled car (condition resource=Car.carbon), the AirQuality map(-map=AirQualty) will increase its measure of pollutants by a specific amount in a specific area round the position of the agent(area pm10 10 5, area pm5 15 10, area pm25 12 15). This generated air quality map can be compared to the sensor based air quality map coming from institutional providers in order to validate the simulation.

Listing 4.2: Unit Rule Code Example - Car Pollution

```
agentRule carPollution
    condition resource=Car.carbon
    repeatAfter 10
    area -timestamp=Time -map=AirQuality pm10 10 5
    area -timestamp=Time -map=AirQuality pm5 15 10
    area -timestamp=Time -map=AirQuality pm25 12 15
    area -timestamp=Time -map=NoiseMap noise 8 10
end
```

Charging stations attract EV, which, in order to refuel, generate increased traffic in their vicinity, often critical during rush hour or in areas with limited road infrastructure. This congestion involves not only electric vehicles, but also internal combustion vehicles, which, stuck in traffic, release pollutants such as PM2.5 and NOx, worsening air quality.

Moreover, during peak traffic periods, charging stations can experience very high energy demand, creating pressure on the remaining capacity of the local electric grid, especially if multiple stations in the same area are overloaded. Pollution data, resulting from road congestion, can inform policies to encourage electric vehicle adoption, such as charging incentives or the introduction of low-emission zones (LZs) in highly polluted areas. An example of a unit rule applicable to charging stations can be seen in Listing 4.3

This process allows tracking of the interactions within and across networks over time. Additionally, the collected metrics support the validation of the simulated model. This interconnectedness between networks and data structures facilitates complex decision-making and validation processes.

4.2.5 Discussion

The extended model proposed focuses on solving interoperability and multilevel integration issues within a complex urban digital context. To tackle the interoperability

Listing 4.3: Unit Rule Code Example - Charging Station

```
unitRule charginStation
    appliesTo ChargingStation
    via Car
    using TrafficNetwork
    using ElectricGrid
    local Car.electric in 1
    local Electricity 1000
    wait 14400
    agent Car.electric out 1
    log -timestamp=Time -event=in Car.electric 1
    log -timestamp=Time -event=out Car.electric 1
end
```

challenge, the adopted approach introduces the concept of metrics for each entity (units, networks, maps) within the urban network. The metrics are not only numerical values but also integrate temporal and spatial dimensions, providing dynamic validity to the information. For example, considering the previously discussed mobility example, the flow of vehicles through an intersection is represented not only as a number (e.g., the number of cars per hour) but also with a temporal connotation (e.g., during rush hour) and spatial connotation (e.g., in a specific area). This approach enriches the data with contextual information, making it more meaningful and facilitating the integration of mobility systems using heterogeneous protocols and formats, such as sensor data, traffic management APIs and public transportation systems.

The simple GlassBox model acts as a "*lingua franca*" that facilitates data exchange between technically diverse systems. The temporal and spatial metrics added by in this proposed extension help overcome semantic barriers by contextualizing the data and aligning systems with different ontologies or vocabularies. In this way, semantic interoperability is not just about "sharing" data, but about understanding and consistency in interpreting information, improving its usability and the ability to integrate it without losing meaning.

Although metrics do not directly solve organizational challenges, they provide a solid technical foundation for collaboration among different stakeholders, such as public transportation authorities, private operators, and urban infrastructure managers. The standardization, both syntactic and semantic, enabled by metrics reduces ambiguities and conflicts that typically hinder cooperation among different entities, promoting the adoption of interoperable frameworks at the organizational level as well. Additionally, the introduction of data management rules, which include the temporal and spatial validity of information, helps apply clearer management poli-

cies, increasing the reliability of shared data and the level of collaboration among various stakeholders.

Regarding the vertical challenge of multilayer integration, the proposed model focuses on managing the interactions among the different components of the urban system without sacrificing the necessary granularity for accurate analysis. In this context, the idea of "networks of networks" fits as a key principle: each network (such as energy, transportation, water) is not just seen as an isolated entity, but as part of a larger network that dynamically interacts with other networks. Local metrics make it possible to analyze the behavior of individual units (e.g., energy flows at an EV charging station), while global metrics provide an overall view of the state of the system, enabling forecasting and resource optimization at the macro level. These metrics, which are constantly updated, allow the dynamics between layers to be synchronized, overcoming the challenges associated with data from systems operating on different temporal and spatial scales. In addition, inter-domain optimization rules, which balance the trade-offs between different domains, make it possible to address cross-domain optimization challenges while maintaining a holistic view of urban interdependencies. In this way, the proposed model concretely addresses the difficulties arising from complex multilevel interactions, enhancing the ability to make timely and informed decisions.

4.2.5.1 Remarks

The approach adopted relies on introducing the concept of metrics for each entity within the urban network, while integrating temporal and spatial dimensions. This enrichment allows for a more precise and meaningful representation of information from heterogeneous systems, such as public transport or shared mobility systems. The proposed model aims to overcome the traditional limitations of digital twins by offering a holistic view of the city, which facilitates the integration of different urban infrastructures and networks.

The model presents several advantages. First, being simple and based on bottom-up modeling, it generates limited additional work/overhead and allows for scalable adoption in different urban contexts. The ability to model complex informational structures linearly enables addressing urban situations with various needs and characteristics, without compromising the consistency of the system. Finally, one of the most innovative aspects of this model is its ability to promote interoperability between UDTs, as each network can be represented by a single point, simplifying communication between diversified sub-systems.

However, this original model also has some limitations. The integration of data and metrics, while crucial for a comprehensive view, adds complexity to the model itself, making the management and maintenance of simulations more challenging, especially when dealing with constantly evolving environments.

5 Temporal Data Containers: A Semantic Framework for Longitudinal Spatial Data Integration

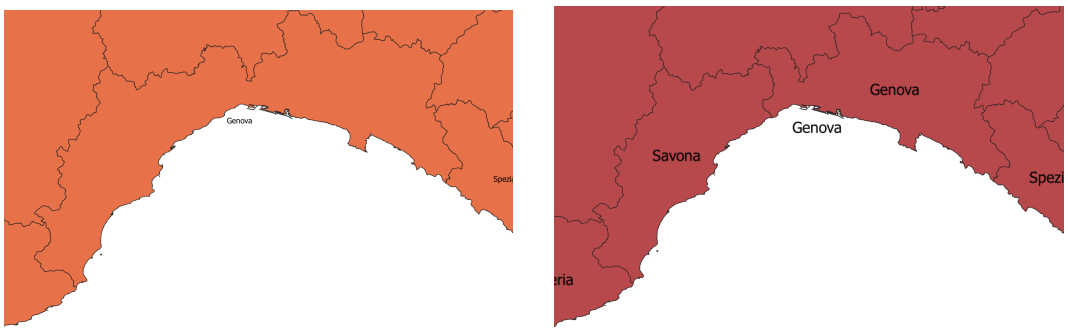
The proposed extension of the GlassBox model allows cities to be represented as multilevel networks of units, resources, maps, and rules, introducing the notion of associating temporal metrics with model elements to support asynchronous data flows. This result highlighted how the GlassBox perspective, although born in a simulation context, could be adapted to real UDT scenarios, maintaining modularity and interoperability.

However, during the course of the applications, a crucial limitation emerged, namely the difficulty of representing and reconstructing the evolution of territorial entities over time. One of the limitations of UDTs is their strong focus on the present. These systems are built to operate in real-time, and consequently, the most common reference model is the snapshot, a synchronous representation of the state of the city at a given moment. However, this current-focused approach is insufficient when trying to understand how the city has evolved over time, much less when seeking explanations or trends in urban transformations. To overcome this limitation, Temporal Data Containers (TDCs) have been introduced as a semantic abstraction that enables the rule-driven integration of spatial datasets across time. TDCs introduce three fundamental elements: the formal encoding of diachronic transformations (such as splits, mergers, renamings, and hierarchy transitions); mechanisms for retrospective reconstruction of consistent snapshots; multidimensional quality metrics for assessing spatial precision, temporal granularity, and semantic completeness. This approach allows to reconstruct past configurations, navigate across successive versions of the same entities and the evaluation of dataset reliability. It also supports the alignment and querying of heterogeneous datasets that differ in scale, format, or temporal reference, thus enabling more consistent longitudinal analyses. The TDC

framework is implemented on the SMW infrastructure [85], extended with temporal indexing, inference modules, and rule-based transformation logic. Conceptually, it aligns with semantic interoperability initiatives such as Frictionless Data [160] and GeoSPARQL [121], while addressing the specific need for temporal reasoning and qualitative reuse of historical urban data. By combining the modular and semantic principles of GlassBox with diachronic reasoning and data quality assessment, TDCs provide a unifying foundation for the longitudinal management of urban knowledge.

5.1 Extended GlassBox Model with TDCs

Historical territorial datasets are often characterized by structural inconsistencies and the absence of stable identifiers, reflecting the data management practices of their time. For example, historical ISTAT datasets did not spatially identify municipalities, which were listed in official documents by name rather than stable codes. As a result, transformations such as the split of the historical province of Genova (code 10) into Genova (10) and Savona (11), with contributions from other provinces (e.g., 45), require explicit reconstruction through semantic modeling (Fig. 5.1).



(a) Genova province before split (b) Genova & Savona after split

Figure 5.1: Example of historical territorial transformation: the split of the Province of Genova into Genova and Savona.

Building around the previous Extended GlassBox framework, the new concept of Temporal Data Containers (TDCs) was introduced. These elemtents are considered ad semantically enriched units that encapsulate datasource provenance metadata, the spatial geometry of territorial entities and the transformation logic these undergo over time. This logic is expressed through explicit rule objects that document the lineage of each change, thereby enabling the precise reconstruction of how spatial objects evolve from state A to state B. In doing so, TDCs provide a semantic bridge

between raw historical datasets and analytically meaningful information, supporting longitudinal reasoning and the generation of Historical Digital Twins.

Concretely, the introduction of the Temporal Data Container allows the Extended GlassBox Model to:

- **Contextualize spatial entities within specific historical realities**:rather then treating a territorial unit as a timeless object, the framework models each entity with temporal boundaries, capturing its governance, spatial layout, and administrative status during a given period.
- **Model transformations such as splits, mergers, renamings, and hierarchy changes**: each transformation is encoded as a formal rule in the system, specifying the nature of the change, the entities involved, and the temporal boundaries maintaining a link between different elements across historical records.
- **Apply rule-based reasoning to automate alignment and integration across datasets**: by associating transformation rules with datasets and entities, the framework enables dynamic generation of harmonized views over time. When querying a specific year, the system identifies the closest temporal anchor dataset and applies relevant transformation sequences to derive the correct configuration.
- **Support robust longitudinal analysis and quality-aware reconstruction**: In addition to transformation logic, each TDC is annotated with data quality descriptors, including spatial resolution, temporal granularity, and semantic completeness.

Fig. 5.2 shows the steps to create TDCs and navigate within the graph.

5.2 Semantic Infrastructure

The semantic backbone of the system is implemented in SMW, which serves both as the knowledge base and reasoning engine using BlazeGraph. In the extended GlassBox model, Semantic MediaWiki was located in the presentation layer, since it primarily acted as a semantic interface exposing observability of entities and interactions. However, SMW was not limited to passive presentation. Thanks to its reasoning modules, it already provided inferential capabilities, effectively bridging the presentation and application layers.

Modeling Datasets and Entities The starting point is historical datasets, which are imported into SMW. They are not loaded in their raw state, but enriched with explicit metadata that defines their fundamental characteristics. This includes structural metadata such as:

- **DID**: a decentralized identifier
- **Temporal validity**: explicit start and end dates for which the dataset is applicable, often derived from census years or administrative decrees;
- **Spatial granularity**: level of administrative detail (e.g., province, district, municipality);
- **Measures**: fields containing measures of phenomena (e.g., population, GDP, cars passing in area);
- **Source provenance**: the origin of the dataset;

as well as data quality metadata:

- **Source Reliability**: the quality level of the source (e.g. insitutional data, academic, vs. crowd-based);
- **Spacial precision**: creation method (e.g., digitized shapefile vs. scanned map);
- **Temporal granularity**: the coarseness of the temporal information (where it applies)

Each spatial entity, such as `Commune of Torino (1861)` or `Region of Emilia-Romagna (1970)`, is also modeled as a separate SMW page with:

- **geometric attributes** linked via external files or spatial property templates,
- **temporal scope of validity**,
- **hierarchical relations** (e.g., isPartOf, isCapitalOf)

Modeling Transformation Rules Transformations between entities are represented a separate pages. Each rule page declares:

- **Input and output entities**, using typed relations;
- **Execution logic**, the applicable list of Transformation Rules;

- **Temporal range of applicability**;
- **Linked sources or official acts** (e.g., legislation, cartographic evidence).

There are different Transformation Rules defined:

- **SplitRule** — describing the division of an entity into two or more parts (e.g., *Piemonte-Liguria* → *Piemonte* + *Liguria*);
- **MergeRule** — representing the amalgamation of several entities into one;
- **RenameRule** — capturing name changes while preserving identity continuity;
- **ReclassifyRule** — mapping changes in administrative levels (e.g., from district to province).

Temporal Data Container Temporal Data Containers are defined by the following major elements:

- **Data Collector** defining the local and remote references to all files to be used and made available for the transformations to work;
- **Data Descriptors** defining the metadata for the various files and datasets;
- **Transformation Descriptors**, defining the transformations applied to the geographic entities; The collection and interconnection of Transformation Descriptors defines the Transformation Graph;
- **Metadata**, a metadata descriptor for the single TDC.

The TDC metadata contains also its global quality indicators, that can be calculated from its contained data descriptors:

- **Source Reliability**: the mean value of source reliability across all files (usually the same value);
- **Spacial precision**: the smallest value across all files;
- **Temporal granularity**: the largest value across all files;

In addition to these indicators, the quality global quality of a TDC is represented also by a **Data Coherence** indicator, that represents the average variability in the three indicators.

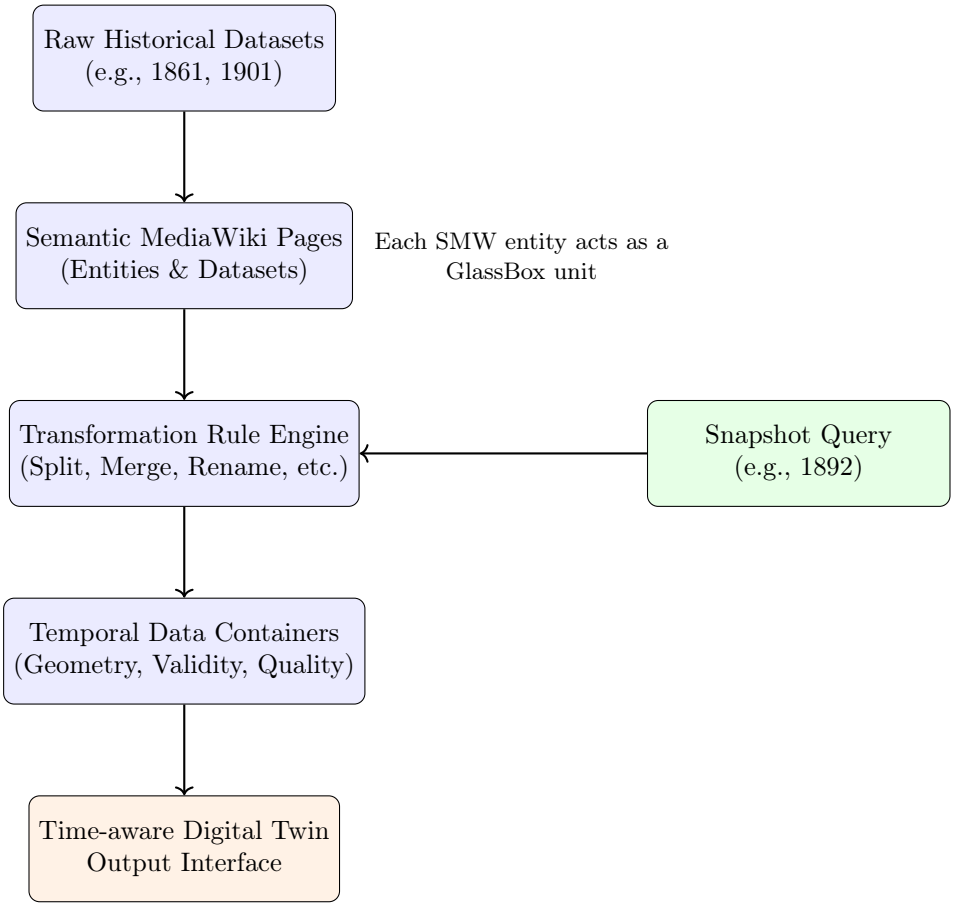


Figure 5.2: Rule-Driven Semantic Integration of Historical Urban Datasets into Temporal Data Containers (TDCs). The vertical pipeline highlights how raw datasets are semantically modeled in SMW, enriched through transformation rules, and organized into TDCs for snapshot reconstruction and time-aware digital twin applications.

5.2.1 Trasfromation Graph

As showed in Fig.5.3, nodes correspond to administrative entities and incorporate information such as geometry, validity period, and data origin. Arcs are the previously defined Transformation Descriptors using split, merge, rename, or reclassify operations, documenting how an entity changes from one historical phase to the next. Each arc can also be manually enriched with additional attributes, such as the source certifying the transformation (e.g., an official decree or an ISTAT census) and a confidence level that expresses the degree of reliability of the change based on available evidence.

In addition to this, each rule, can be also used in validation workflows, where transformed datasets are checked against: temporal coherence (e.g., overlapping validity ranges), spatial consistency (e.g., disjoint union of split geometries) and Semantic plausibility (e.g., hierarchy violations or gaps in lineage).

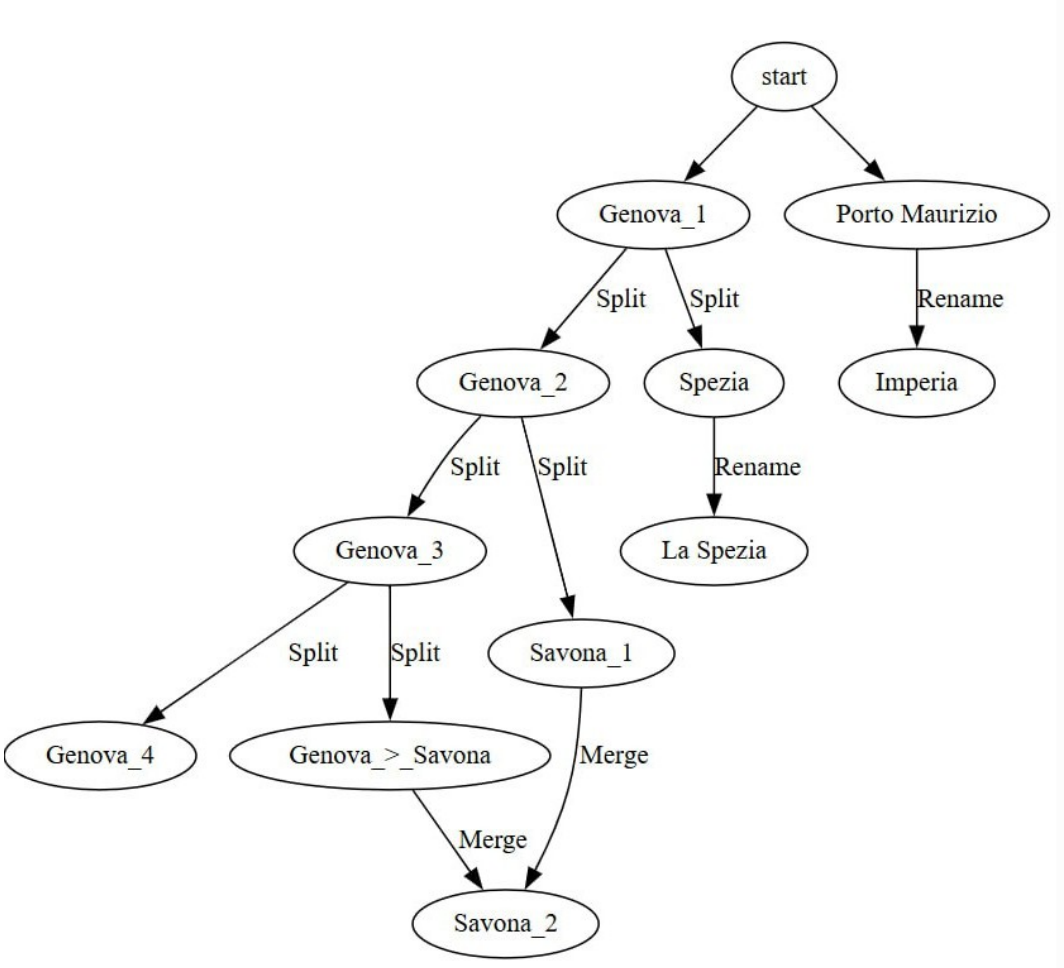


Figure 5.3: Trasformation Graph

5.2.2 Temporal Querying and Snapshot Reconstruction

Thanks to this infrastructure, users can issue temporal queries that trigger automated reconstructions. For example, a query such as:

"What was the municipal structure of Emilia-Romagna in 1897?"

is handled by:

1. Identifying the closest datasets (e.g., 1891 census and 1901 census),

2. Checking whether a direct representation exists,

3. If not, applying a forward sequence of rules (`MergeRule`, `RenameRule`, etc.) on the earliest of the two datasets using only rules that apply before the seleted date,

4. Reconstructing the state at 1897 as a derived, inferred snapshot.

5.3 Implementation

To demonstrate the practical applicability of the framework, a prototype system was instantiated. As previously defined, the TDC is a composition of elements describing various aspects of complex data management and transformations. The implementation of the TDC defines multiple approaches to the management of the structure: a file based approach, where the descriptors are managed in folders and can be collected as compressed archive, a db-based approach, where these descriptors are shared as a sqlite file with specific table structures and a REST protocol.

5.3.0.1 Use Case

Building a TDC based on ISTAT historical territorial subdivisions was an interesting use case as the data spans from 1861 until now on the whole territory of Italy. When evaluating the overall quality of the T TDCs, the Source Reliability and Spatial Precision remain consistently stable, whereas the Temporal Granularity varies across different periods (10 years, 2 years, 1 year). As a result, the overall level of data coherence is high, though not fully complete.

An analysis of the data reveals a complex pattern of change: while the first subdivisions were Regions, Provinces and "Circondari", and this subdivision remained stable until mid XXth century, and became then Regions, Provinces and Municipalities, eliminating the Circondari from the equation and adding the Municipalities. For this reason, in long duration Provinces are considered the main elements, while on recent transformations the municipal level represent the most important and atomic level of analysis.

5.3.0.2 Prototype Development

In order to create a prototype, a Python application (`tdc.py`) was developed. This script is called from the command line and handles a series of options to filter data

based on time interval, attributes, or indexes, and to merge map layers with statistical datasets. The goal is to reconstruct territorial transformations and generate polygonal layers that can be used in GIS environments such as QGIS or Mapbox.

The Fig. 5.4a is an example call to generate a graph for a specific region between two years.

```
# Example of temporal
# graph generation
tdc.py graph
--fromtime 1861
--tottime 1911
--fltr "Region=Genova"
--idx TEXT filter based on index
```

(a) Example of temporal graph generation

```
# Example of join operation
tdc.py graph
--internal "administrative_level=6"
--joinfile population.csv
--joinon "ProvinceCode"
--fromtime 1982
--tottime 1991
```

(b) Example of join operation between two entities

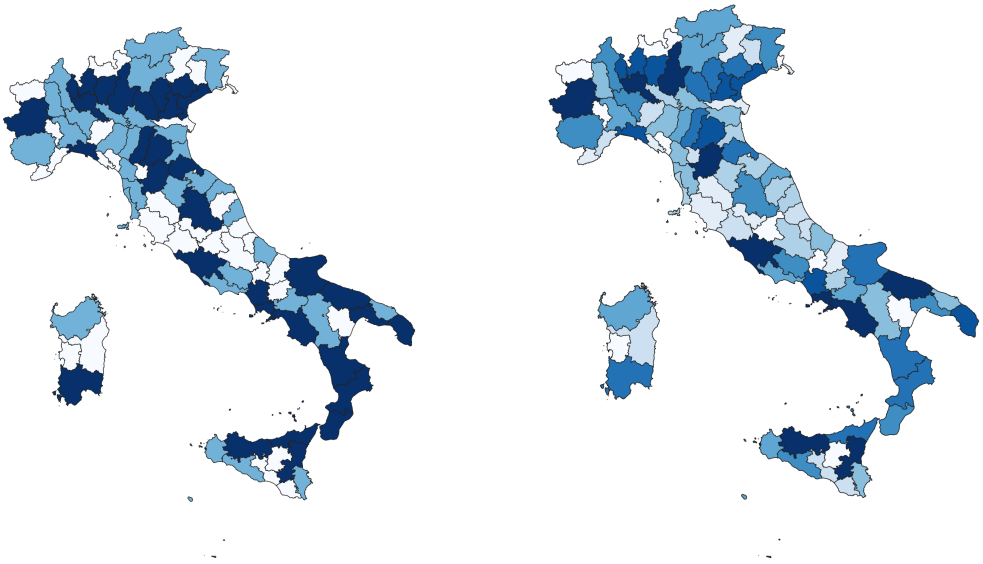
Figure 5.4: Command-line examples: (a) temporal graph generation, (b) join operation.

The Fig 5.4b is an example call to join between an external dataset (popultion.csv) and the spatial data between two dates. In this example, the entities within the TDC, filtered by `administrative_level=6` (representing the provices) provides the geometries of administrative units, while `population.csv` contains demographic values indexed by the field
`ProvinceCode`. The parameter `-joinon` specifies the common key used for the join (in this case the external `ProvinceCode` matches with the internal `COD_PROV` but is not repeated as it is defined in the metadata, else it would be defined by `-on`). For each temporal slice between 1982 and 1991, the script enriches each polygon with the corresponding population data, thus creating a year-by-year set of spatial layers.

5.3.0.3 Map Generation and Visualization

The output of the system consists of GeoJSON files, each containing the geometries of the reconstructed entities enriched with joined attributes. These files can be directly imported into GIS environments such as QGIS or Mapbox. Once loaded, they appear as standard polygon layers that can be styled according to the selected attribute, for example, a choropleth visualization of population density across provinces.

Figure 5.5 shows an example visualization of Italian provinces in 1982 and 1991, where the system highlights both demographic trends and boundary transformations. In this way, the implementation bridges the abstract transformation graph with



(a) italian provinces in 1982 (b) italian provinces in 1991

Figure 5.5: Examples of visualization of Italian provinces in 1982 and 1991

practical spatial outputs, enabling reproducible and interpretable reconstructions of historical territories.

5.3.0.4 Limitations

The implementation confirms the feasibility of the approach but also highlights some intrinsic limitations. In particular, the data available for certain dimensions and time periods are incomplete, and the process of joining external sources can lead to indexing misalignments when entities have undergone mergers or splits. Observing ISTAT data, these issues emerge as incomplete coverage of municipalities across all census years as well as potential inconsistencies in reconstructing population values after territorial transformations due to a lack of identifiers.

5.4 Discussion

This contribution builds upon the extended GlassBox model to provide a unifying semantic structure for robust spatio-temporal analysis using both legacy and contemporary (i.e., currently maintained and updated) datasets. The resulting framework facilitates longitudinal studies without requiring strict schema harmonization; promotes reuse of heterogeneous public data sources through semantic linking; supports

analytical workflows in urban history, planning, territorial governance, and cultural heritage studies

Although the prototype focuses on a national case study, the framework is explicitly designed to generalize across spatio-temporal domains. Its semantic architecture, based on Temporal Data Containers, Transformation Rules, and Rule-Based Reasoning, can be extended to multiple application areas such as environmental monitoring, cadastral systems, infrastructure management, and smart mobility.

6 Federated Trust and Governance Framework for Urban Digital Twin

The third challenge addressed in this dissertation concerns data federation and trustworthiness in UDT ecosystems. As UDTs increasingly rely on distributed and independently governed data sources, ensuring interoperability, sovereignty, and verifiable trust across domains becomes a fundamental requirement. This chapter explores how the emerging paradigm of data spaces can support federated data management while preserving control, accountability, and data integrity among autonomous actors. Building on these foundations, the section introduces the VESPACE framework, a verifiable blockchain-based architecture designed to implement trust and sovereignty in federated UDT environments.

6.1 VESPACE: A verifiable blockchain-based data space solution for Urban Digital Twins

UDTs are designed to integrate heterogeneous data flows from multiple domains to optimize and coordinate city operations. However, these datasets are distributed across independent organizations that apply different policies, formats, and governance rules. This fragmentation makes centralized integration infeasible and highlights the necessity for federated architectures that enable secure and controlled data exchange among autonomous actors. In such environments, collaboration is achieved through federated data sharing, where data remain under the sovereignty of their original providers, but can be queried and reused on demand. This approach aligns with the concept of data spaces, as defined in recent European initiatives, which establish a common governance framework for secure data exchange across domains, jurisdictions, and infrastructures [58].

A data space provides the organizational and technological foundation for data federation, defining how independent entities can exchange data while preserving

control, traceability, and compliance. When applied to UDTs, this paradigm enables cities to evolve from isolated systems towards interconnected, verifiable, and sovereign digital infrastructures, where data, models, and services can be composed dynamically to support city-scale decision-making. However, the shift toward such federated architectures introduces new challenges related to trustworthiness. Participants must be able to verify the authenticity, integrity, and policy compliance of shared data while maintaining privacy, accountability, and ethical use across distributed environments. Although several data-sharing platforms have been proposed, most focus on supporting real-time data streams, overlooking the need to access and verify historical or pre-generated datasets. To address these limitations, the VESPACE framework is introduced as a verifiable blockchain-based data space solution for trusted and sovereign data sharing within federated ecosystems [106]. VESPACE embodies the principles of SSI by leveraging DIDs and VCs, empowering participants with full control over their digital identities and datasets. Each entity is uniquely identified through a DID, while VCs are used to verify dataset authenticity and define access rights. Since these credentials are issued directly by data providers, they can be dynamically revoked or updated, enabling flexible and privacy-preserving access control. All information related to dataset certification and access events is recorded on the blockchain, ensuring immutability, auditability, and transparency across the ecosystem. Datasets are stored in a decentralized storage infrastructure operated by the participants of the platform, guaranteeing both resilience and sovereignty over the data lifecycle.

To validate this design, a prototype implementation of VESPACE has been developed and evaluated using multiple open datasets from the Municipality of Bologna (Italy). The results confirm the feasibility and scalability of the proposed architecture, demonstrating how decentralized trust mechanisms can effectively support federated and verifiable Urban Digital Twin environments.

6.2 Design Guidelines

Designing a data space environment where customers can directly share and verify the authenticity of data are essential. In this section, a set of principles is introduced to define a verifiable data space that facilitates cross-domain data sharing.

6.2.1 Functional Requirements

This subsection highlights some fundamental functional requirements of a dataspace environment in accordance with the EU initiatives discussed previously.

FR-1 Data Sovereignty. In data spaces, data sovereignty refers to the ability of organizations, governments, and individuals to maintain full control over their data [70]. This capability encompasses how data are shared and used by others, as users should have the ability to dynamically update access to their information at any time. IDSA [72] offers guidelines and a framework for ensuring data sovereignty in data spaces, emphasizing the importance of clearly defined data usage policies and contracts. This approach ensures that entities maintain control over their data, sharing them based on their specific preferences and conditions.

FR-2 User Management. A data space must protect all participants and systems within the ecosystem. All entities involved should be registered, and mutual authentication must be established before accepting any requests. Users can interact with the system and perform actions based on their authorization privileges, which determine their permitted operations. These include access to specific datasets and authorization to share data. Additionally, since access conditions may change, the system must include mechanisms to dynamically update user permissions.

FR-3 Discovery and Selection. Given the potentially large amount of heterogeneous data that populate data spaces, product discovery and selection are crucial capabilities of dataspaces. The system should empower users to search for information within the data space, supporting queries that match their specific requirements with relevant products. In EOSC [48], these features are implemented through catalogs (e.g., for datasets, services, standards), which allows identifying information through machine-readable metadata. Thus, each dataset should be associated with a comprehensive set of metadata detailing its attributes and characteristics, enabling users to conduct targeted searches based on specific criteria. This functional requirement addresses the first FAIR principle of findability, ensuring that users can locate and access data products efficiently.

FR-4 Data Access. This requirement aligns with the second FAIR principle of accessibility. It is essential to have well-defined data access policies that allow data owners to decide with whom to share information and when to deny access to specific users. These policies should ensure that only authorized users can access sensitive data while also making public data readily available to the research community. This granular control over data access helps protect the privacy and security of information

while maintaining a high level of accessibility. Authorized users must therefore be able to access and share data through trusted data repositories, ensuring the long-term sustainability of research data. OpenAIRE guidelines for repository managers highlight the importance of ensuring data accessibility through standard protocols and facilitating data reuse by complying with community standards [122].

FR-5 Data Transfer and Exchange. A key functionality of data spaces is the transfer of data from one participant to another. Data owners should be able to publish verifiable data, while consumers can easily retrieve them. Data space platforms must ensure that consumers can always detect unauthorized alterations of requested data and that data delivery is efficient, minimizing latency between data access and delivery. For example, the DBSA reference architecture incorporates mechanisms to detect unauthorized alterations, thus supporting efficient and secure data transfers.

FR-6 Data Interoperability and Portability.

Secure and efficient data transmission across platforms is vital for collaboration and maximizing data utility. Projects and initiatives like EOSC and IDSA leverage interoperability to enable seamless, mutually beneficial data exchange. Standardized protocols and formats ensure dataset integration while preserving integrity and context. Embracing interoperability fosters dynamic ecosystems where information flows freely, driving data-driven insights, innovation, and cross-border collaboration.

FR-7 Auditing and Compliance. Data spaces must monitor and record the entire lifecycle of data, with particular emphasis on data accesses. This capability is crucial for detecting unauthorized use or potential data breaches. Monitoring operations are also necessary to ensure compliance with existing data protection regulations, such as GDPR. OpenAiRE infrastructure supports the tracking of data usage and compliance with legal and ethical standards, which is crucial for maintaining data integrity and ensuring that data reuse adheres to established norms.

6.2.2 Non-functional Requirements

Non-functional requirements define the qualitative characteristics and constraints of a system, influencing how it satisfies functional requirements. This subsection outlines the main non-functional requirements identified for verifiable data spaces.

NFR-1 Security and Privacy. Data spaces must guarantee secure storage and transmission of data. To promote participation, data owners should be confident that their data are managed responsibly and ethically following existing regulations and using secure technologies. Unauthorized entities that are not directly involved

in a specific data exchange must be prevented from accessing the data. This requires robust encryption protocols for both data at rest and data in transit, ensuring that data remain confidential and tamper-proof throughout its lifecycle.

NFR-2 Reliability and Usability. The data space must ensure continuous availability and functionality, minimizing downtime and errors. This requirement requires implementing redundancy and fault tolerance measures, as well as robust backup and recovery processes to maintain seamless operations. In addition, the system should be user-friendly and offer intuitive interfaces that simplify the process of identifying and accessing the desired information.

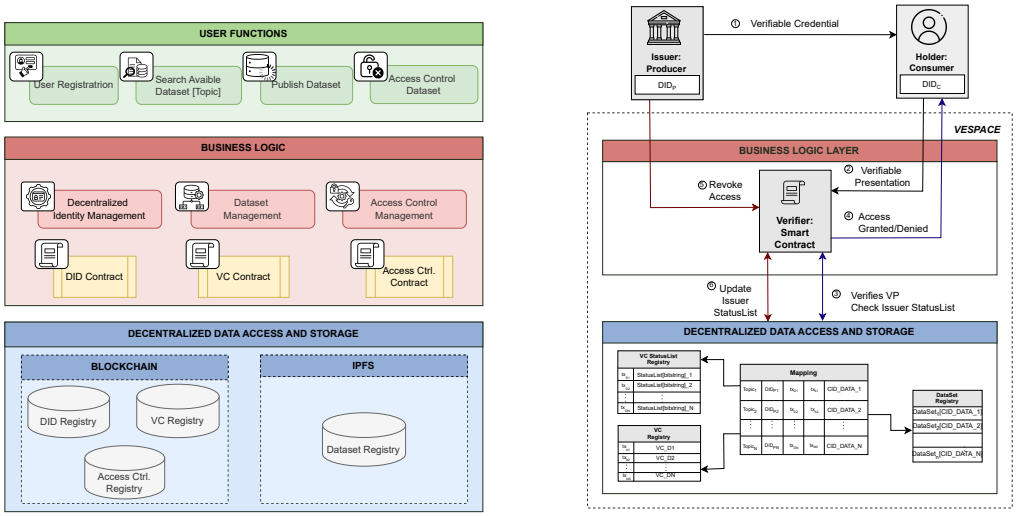
NFR-3 Verifiability and Trustworthiness. Users of data spaces should feel confident that the collected data have been released by reputable organizations and individuals. Verifiability ensures that the data are neither manipulated nor altered during transmission or storage. This is essential to maintain the trustworthiness of the data and to ensure the validity of any evaluations or analyses based on them.

NFR-4 Scalability and Performance. Data spaces should be able to adapt to varying numbers of users and volume of data. In addition, they should support high-performance data analytics use cases and facilitate rapid data transfers between the platform and its participants.

NFR-5 Auditability and Transparency. Users, governing bodies, and regulators should be allowed to conduct comprehensive analyses of operations within data spaces. This involves accessing information on the types of data exchanged, the purposes for which they are used, and the duration of their use. Individuals should be fully informed about their ability to contribute to the data space, the entities with access to their data, the storage locations, the security measures to safeguard their data, and the methods available for interacting with it.

6.3 Architecture

VESPACE provides a platform for sharing data between different UDT stakeholders according to the principles of SSI. Each participant in the ecosystem is identified through a DID, which enables accessing corresponding public keys from DID Documents. Data owners, referred to as producers, retain full control over their data, since access is granted through self-issued VCs. Thus, producers can eventually revoke access permission to their generated data. The credentials contain the hash of the corresponding dataset, which guarantees the integrity of the dataset, and the necessary information used to revoke access to the data.



(a) 3-tier architecture.

(b) Data access management.

Figure 6.1: On the left is shown VESPACE architecture comprising user functions, which act as abstractions built on the business layer's functionalities. The business layer manages data transactions through on-chain interactions while facilitating storage and retrieval via decentralized storage systems. The right side illustrates the flow of interactions for granting and revoking data access to consumers.

As shown in Fig. 6.1a, VESPACE is logically organized in a 3-tier model consisting of a Presentation (User Functions), a Business Logic, and a Decentralized Access & Storage Layer. The Presentation Layer is the entry point for users, offering high level functions to securely share and access data. The Business Logic Layer encapsulates the core functions, handling data processing, access control management, and decision-making operations. Finally, the Data Access Layer manages all the information for the storage, retrieval, and access control rights of data. In the following, a detailed description of the main components of the platform is provided.

6.3.1 System Users

In a data space, users refer to individuals (municipalities, utility providers, transport operators, companies, and citizens), who may assume the role of data producers, data consumers, or both, as detailed below.

Producer. The producer p is any entity that shares data within the data space. Each producer is uniquely identified by a DID (did_p), linked to a pair of cryptographic keys (sk_p, pk_p). Producers act as issuers as they release VCs to authorize

data access and certify data. Given a dataset D, the producer p certifies the authenticity of the dataset with a credential vc_D and provides an authorized consumer $c \in \mathcal{C}$ with a credential vc_D^c. These credentials are signed using sk_p (Step 1 of Fig. 6.1b), ensuring decentralized verification of access rights and data authenticity. Furthermore, producers retain fine-grained control over data access and revocation through the W3C Bitstring Revocation List vb_t [169], comprised of a bitstring b_t, where each bit indicates the state of vc_D^c: if the bit is set to 1, the credential is revoked; if it is set to 0, the credential is still valid.

Consumer. Consumers are entities that seek to use datasets shared via VESPACE. Each consumer c is uniquely identified by a DID (did_c), associated with a pair of cryptographic keys (sk_c, pk_c). Only authorized consumers can query the platform and access the data using VCs issued by the producers. Given a credential vc_D^c, VESPACE verifies its authenticity using the producer's public key pk_p. To access data, consumers generate VPs signed with their secret key sk_c as shown in Step 2 of Fig. 6.1b. This mechanism prevents replay attacks, as only the legitimate holder has sk_c corresponding to did_c referenced in vc_D^c.

6.3.2 User Functions

Part of a presentation layer, they offer an interface to producers and consumers to use VESPACE's high-level abstractions. The platform is implemented as a Decentralized Application (DApp), which interacts with the back-end of the blockchain through smart contracts. The essential functionalities are:

- *User Registration*: Register users by presenting their DID and necessary information as VC issued by a trusted authority such as a government or any accredited institution;
- *Publish Dataset* and correspondingvc_D acting as dataset certification;
- *Search Available Dataset*: View available dataset metadata-based research such as topic t_D;
- *Access Control Dataset*: This function allows data producers to grant or revoke access permission by updating the access control list associated with that dataset.

6.3.3 Business Logic

The business logic constitutes the core of the VESPACE platform, as it encapsulates the main system functionalities and the processes required to manage and regulate data access. This layer relies on blockchain technology and smart contracts, which ensure the secure, transparent, and automated execution of key operations. In VESPACE, the blockchain stores the DID Document of each user, vb_t and vc_D for each dataset D, and the binding with the information to manage data access. This eliminates the risk of tampering and establishes a transparent audit trail accessible to all authorized parties.

User functions abstract several management flows enacted at the business logic level used to, e.g., control, grant or revoke access to a dataset. Most of these operations are supported by smart contracts that require interactions with the blockchain:

- *Decentralized Identity Management*: Support DID and VC operations, such as their verification during the registration phase, uploading both vc_D to verify data set authenticity and vc_D^c for operations related to access control rights.
- *Dataset Management*: Manages operations related to storing, retrieving, and modifying datasets.
- *Access Control Management*: Tracks every dataset uploaded by each producer p, its Topic t_D, did_p, and the associated certification vc_D and revocation list vb_t. It checks if a consumer is allowed to access a specific dataset.

Indeed, smart contacts play a key role in the VESPACE architecture, used to connect applications that implement business logic with the data access layer. The delivery of VESPACE services is achieved through a collection of smart contracts deployed on a blockchain, as follows:

- *DID Contract*: Manages the DID lifecycle, including registration and resolution.
- *VC Contract*: Manages all credentials involved in the registration, certification, access, and revocation processes.
- *Access Control Contract*: Handles all the operations on the Access Control Registry (ACR), which maintains a binding among the dataset topic t_D, did_p, the transaction identifier tx_b corresponding to the dataset's revocation list, the transaction identifier tx_D linked to the dataset's certification, and the content identifier cid_D used to reference the dataset itself.

6.3.4 Decentralized Data Access and Stroage

The Data Access Layer is essential for efficient and secure data access management within VESPACE. This layer comprises a blockchain and a storage component, implemented through IPFS.

Decentralized Storage. In VESPACE, datasets are stored in an IPFS cluster [73]. Each dataset D is assigned a content identifier cid_D which acts as a unique identifier used to denote and retrieve the dataset upon authorization. Distributing data across multiple nodes reduces server load and improves scalability, allowing this architecture to handle increased user traffic without performance degradation. Moreover, IPFS improves resilience by seamlessly retrieving data from alternative nodes if primary access points fail, thus increasing VESPACE availability and fault tolerance. Finally, IPFS uses highly connected pinning services and parallelization, which makes it particularly beneficial for large or frequently accessed datasets.

Verifiable Registry. blockchain is used to record DID documents, VCs, and the Access Control Registry. Whenever a dataset is saved to IPFS, a vc_D is issued and stored on the blockchain to certify its authenticity. As a result, the blockchain returns a confirmation transaction tx_D. Simultaneously, the producer p generates a status list vb_t, saves it on the blockchain, and stores the returned tx_b along with the t_D, the did_p, the tx_D and the dataset cid_D in the ACR. This data structure allows the efficient retrieval of all information to prove the dataset authenticity, also regulates access.

The transparency and immutability of the blockchain ensure that the information is securely stored, preventing any alteration and boosting the trustworthiness of participants in using the data space. The blockchain component is implemented on the Sepolia testnet, which enables the testing of smart contracts and decentralized applications (DApps) within a real-world Ethereum environment. To access the blockchain and interact with Sepolia's nodes, Alchemy is used, providing reliable, scalable, and seamless communication with the blockchain.

6.4 Secure and Verifiable Data Sharing

This section describes how VESPACE enables secure and verifiable data sharing, meeting the requirements identified in Sec. 6.2. It should be noted that all participants must undergo a registration process, in which their identity is verified based on DIDs and VCs issued by a trusted organization, such as a government. In Data Sharing and Certification, producers make their data available and certify them,

leveraging vc_D. In the data retrieval and verification phase, consumers interact with VESPACE to retrieve datasets related to a given topic. The VESPACE checks against the access policies for the requested data, and if the user can access the information, it grants the requested data. Finally, in the access control management phase, producers can interact with VESPACE to check and update the list of users who can access the information.

6.4.1 Registration

To interact with VESPACE, users must be registered within the system. The system operates under the assumption that the entities involved adhere to the SSI framework, in which participants possess DIDs associated with VCs issued by trusted entities such as governments or other authorized third parties, ensuring the reliability of data producers. This approach aligns with the principles outlined in the eIDAS 2.0 regulation, enacted in May 2024, which aims to enhance trust and security in electronic transactions throughout the European Union by promoting the use of digital identities and trust services [50].

In VESPACE, both producers and consumers have a key pair associated with their DID, generated by the Ed25519 algorithm $(1\lambda) \rightarrow (sk, pk)$. Each user also has a VC issued by a trusted authority, which is used to attest some of their properties to the User Agent for registration purposes. As users must already be registered with the entity issuing the VC, verification of the provisioned data is performed by interacting with the blockchain storing the DID Documents, referred to as the Identity Blockchain. In case the verification process succeeds, VESPACE registers the user information on the blockchain and issues a new credential, namely, VC Registration vc_r^c, serving as an authorization token to interact with the platform.

This registration procedure meets the requirements of FR-1 (Data Sovereignty), FR-2 (User Management), and NFR-1 (Security and Privacy). Moreover, using the VC standard also satisfies the FR-6 requirement of Data Interoperability and Portability.

6.4.2 Data Sharing and Certification

Authorized producers can share data in VESPACE following the data sharing and certification process reported in Alg. 1. First, the producer produces a hash of the dataset D using the SHA256 hash algorithm $h_d \leftarrow h_{sha256}(D)$. Then, it issues a self-signed VC:

$$vc_D = [did_p \| t_D \| h_D \| m_D] \tag{6.1}$$

Algorithm 1 VESPACE data sharing and certification.

```
Input Initialization
h_D ← Hash Dataset using SHA256 Algorithm
vc_D ← issue VC to certify Dataset
vb_t ← issue VC status list containing b_t managing access to the dataset

function PublishDataset(
      did_p, D,
      vc_D[t_D, h_D, MD], vb_t )
result_vc ← Verify vc_D using pk_p on VESPACE
if result_vc then
   result_id ← Verify vb_t using pk_p on VESPACE
   if result_id then
      tx_D ← store vc_D
      tx_b ← store vb_t
      cid_D ← upload D to IPFS
      newEntry ← Update ACR
      t_D, did_p, tx_D, tx_b,
      cid_D )
   end if
end if
```

where t_D is the topic associated with the dataset, h_D is the dataset hash and m_D contains metadata, including the title, detailed description, quality certifications, provenance information, and other pertinent details that improve the reliability and usability of the dataset. Metadata offers consumers a clear view of the data available in the data space, in compliance with FR-3 (Discovery and Selection). Furthermore, for each consumer c, the producer issues a vc_D^c containing its DID did_p, the DID of the consumer did_c, the topic associated with that dataset t_D, and the index i of the credential in the bitstring b_t:

$$vc_D^c = [did_p \| did_c \| t_D \| i] \tag{6.2}$$

Then, a VC status list vb_t is generated, used to regulate access to D. The dataset D, the corresponding certification vc_D, and the revocation list vb_t (FR-1) are shared with VESPACE, while consumers are provided with vc_d^c.

VESPACE resolves the DID of the producer did_p contained in the credential, and collects the corresponding public key pk_p from its DID Document. The public key is used to verify the authenticity of vc_D and vb_t (FR-5). Once the verifications are confirmed, VESPACE saves the vc_D certifying the dataset on the blockchain, along with its vb_t. This produces two transaction identifiers, respectively, tx_D and tx_b, which will be used to retrieve information in the following operations. The dataset

is then stored on IPFS (FR-3), which binds it to a content identifier cid_D. Finally, all this information is included in a new entry of the ACR, which is structured as:

$$acr \leftarrow [t_D \| did_p \| tx_D \| tx_b \| cid_D] \tag{6.3}$$

Storing the vc_D on the blockchain, the dataset on decentralized storage, and the subsequent ACR mapping on the blockchain fulfills the functional requirements FR-7 and FR-8. It is worth noting that a producer can associate more data to the same topic.

Regarding non-functional requirements, the VESPACE framework adheres to the NFR-2, NFR-4, and NFR-5 requirements. The use of blockchain and decentralized storage improves NFR-2 by providing fault tolerance, data redundancy, and continuous availability, reducing downtime risks. NFR-4 is addressed by leveraging decentralized architectures that support increasing data volumes and user demands while ensuring efficient data discovery and rapid retrieval through content identifiers. Finally, NFR-5 is inherently supported by the immutability of blockchain records, which store verifiable credentials and revocation lists, allowing traceability, verification of data provenance, and comprehensive oversight of data exchange operations within VESPACE.

6.4.3 Data Retrieval and Verification

Consumers can interact with the platform to browse and access datasets related to topics of interest. Algorithm 2 details the interactions among consumers and VESPACE to retrieve datasets for a specific topic. Given a valid credential vc_D^c, the consumer c generates a verifiable presentation vp_D^c by signing it with its own private key sk_c. VESPACE verifies the signature by resolving $did_c \in vc_D^c$, and obtaining pk_c.

Before accessing the dataset, VESPACE must ensure that the consumer's access has not been revoked (FR-4). Using t_d and did_p contained in vc_D^c, VESPACE retrieves tx_D and tx_b from acr for each D associated with t_d. These transaction identifiers are used to retrieve vc_D and vb_t, respectively. VESPACE platform verifies whether the following equation holds $b_t[i] \neq 1$ where $i \in vc_D^c$ (Step 3 of Fig.6.1b). If the equation is verified, VESPACE retrieves D through cid_D (Step 4 of Fig. 6.1b).

Consumers receive the datasets along with associated vc_D. For each retrieved dataset D_r, the consumer verifies if $h_{sha256}(D_r) = h_D \in vc_D$ (FR-5, NFR-3). This allows each consumer to directly verify the authenticity of the collected data in a fully decentralized manner.

Algorithm 2 VESPACE data retrieval and verification.

```
Function SearchAvaibleDataset(Topic)
vc^c_D[did_p, did_c, h_D, i] ← c request(p,Dataset[t_D])
vp^c_D ← c issue VP from vc^c_D
result ← Verify vp^c_D on VESPACE
if result then
    for (topic, did, tx_D, tx_b, cid) ∈ ACR do
        if topic == t_D then
            if did == did_p then
                tx_D ← tx_D
                tx_b ← tx_b
                break
            end if
        end if
    end for
    vc_D ← recover VC Dataset using tx_D
    vb_t ← recover VC status list using tx_b
    b_t ← extract bitsrting from vb_t
    i ← extract index from vp^c_D
    if  b_t[i]! = 1 then
        cid_D ← from ACR
        D ← IPFS[cid_D]
        return D,vc_D
    else
        Access Denied
    end if
end if
```

6.4.4 Access Control Management

In VESPACE producers have direct control over who can access their data, including the right to decide when a consumer can no longer access them (FR-1). This property is aligned with the SSI principles. Algorithm 4 shows operations performed by VESPACE to revoke access for a topic t_D to a specific consumer c. Specifically, the producer is required to update b_t setting to 1 the bits in the i-th position where $i \in vc^c_D$ of the consumer whose access rights must be revoked (Step 5 of Fig. 6.1b). Consequently, the updated version of b_t is stored on the blockchain, generating a new tx_b. Finally, the ACR is updated by adding a new entry that contains the transaction identifier of the latest computed status list (Step 6 of Fig. 6.1b).

Evaluation

This section provides a comprehensive evaluation of VESPACE. A series of experiments was conducted to evaluate dataset certification and access control functionalities, with a focus on assessing the scalability of the mechanisms under varying

Algorithm 3 VESPACE Data access management.

```
Function Revoke Access Dataset(vc^c_D, vb_t)
i ← extract index from vc^c_D
extract b_t from vb_t
search i in b_t
b_t[i] ← 1
Update b_t in vb_t
issue vb_t
tx_b ← store vb_t
Update ACR with new tx_b
```

numbers of participants and datasets. In addition, the experiments were designed to assess the usability of the system in terms of latency. For this purpose, the Response Animation Idle Load (RAIL) model, a performance model proposed by Google, was adopted to provide metrics for evaluating the usability of a platform from a user perspective [135]. Each experiment was executed 50 times, and the results were averaged.

6.4.5 Implementation

The evaluation is based on a real implementation of VESPACE available in [39]. The generation and verification of DIDs, DID documents, and VCs were carried out using the Digital Bazaar library [42], a widely adopted solution for managing digital identities and credentials in compliance with W3C standards. Blockchain-based operations were performed on the Sepolia Ethereum testnet. Smart contracts, written in Solidity, handled mappings between topics, producer DIDs, transaction IDs, CIDs, and bitstring-based status lists. Dataset storage was implemented through a decentralized IPFS cluster consisting of three nodes. This cluster employed a Conflict-Free Replicated Datatype (CRDT)-based consensus mechanism to maintain a global pinset, ensuring data availability and redundancy. To evaluate the system in a realistic scenario, urban datasets from the Municipality of Bologna, Italy, were utilized, sourced from the Open Data Bologna Platform [38] under the CC BY 4.0 license. These datasets include diverse urban data such as parking availability, traffic flow, and environmental metrics, providing a comprehensive real-world test environment.

6.4.5.1 Dataset Sharing and Certification

This set of experiments was conducted to analyze the time required for data owners to certify the authenticity of a dataset and to grant access to consumers. Specifically,

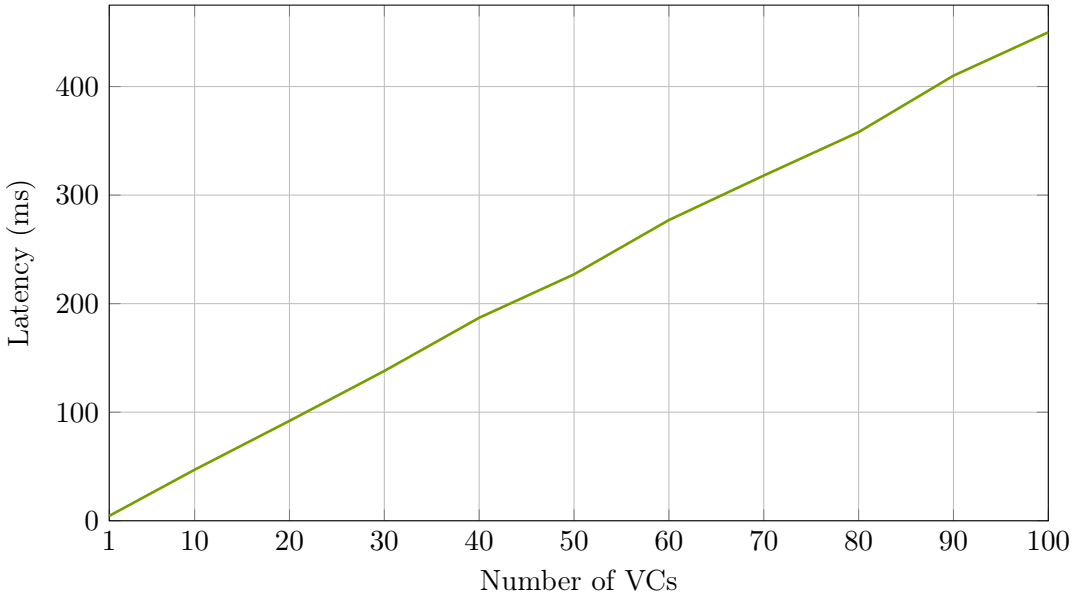


Figure 6.2: Latency (in milliseconds) for the issuance of Verifiable Credentials.

the latency associated with issuing vc_d^c and the corresponding status list contained in vb_t—which regulates access in VESPACE—was measured. As discussed in Sec. 6.4, each vc_d^c also contains an index i, referring to the i-th position in the status list, and the dataset topic t_D, thereby providing c with integrity and authenticity guarantees. The number of consumers was varied from 1 to 100. The time required to generate vc_D can be considered negligible, as each dataset is certified only once. No remarkable differences were observed when considering different datasets.

Figure 6.2 reports the collected results. As expected, the latency grows linearly with the number of credentials issued, ranging from 4,5 ms when issuing a single VC to 449 ms in scenarios with 100 consumers. According to the RAIL model, latency between 100 ms and 1000 ms is perceived as the natural and continuous progression of tasks, allowing users to maintain focus and flow without feeling interrupted or delayed. The results suggest that the issuance process has relatively low overhead as more credentials are generated. The overall latency remains within an acceptable range for real-world applications, ensuring that even a large batch of credentials can be generated rapidly.

6.4.5.2 Access Right Verification

This set of experiments evaluate the performance of the system in handling the authorization verification process for a consumer while varying the number of datasets

Table 6.1: Comparison of Authorization and Revocation Times

Number of Datasets	Authorization Time (ms)	Revocation Time (ms)
1	668	528
2	1337	1056
4	2648	2112
6	4100	3182
8	5448	4233
10	6690	5284

that belong to the same topic. In particular all the phases needed for varying access rights is considered, which include:

1. Resolving the DID Document associated with did_c to retrieve the consumer's public key and consequently verify the vp_D^c issued (line 4 of Alg. 2);
2. Querying acr to obtain tx_b and tx_D and retrieve the corresponding status list and credentials vc_D that certify the data from the blockchain (line 6-16 of Alg. 2);
3. Assessing whether the i-th position of the status list, which corresponds to the index included in the consumer credential is set to 0 (authorized) or 1 (revoked) (line 17-18 of Alg. 2).

The latency of the entire process is measured, reporting the results in the second column of Table 6.1 and in Fig. 6.3, which shows the authorization verification time across different dataset counts. Experimental results demonstrate that VESPACE can scale effectively while increasing the number of datasets. This underscores the system's validity in handling access to multiple datasets belonging to a single topic while maintaining an acceptable response time. It should be noted that the time required to download the datasets was not measured, as it depends on network connectivity and various cluster settings, which affect all platforms used for data sharing.

Access Right Revocation Finally, the efficacy of VESPACE in revoking access rights is evalueted. This is a critical aspect of the SSI paradigm, as it ensures that data owners maintain control over their data over time.It is observed that the time required to revoke access rights by generating the bitstring status list remains

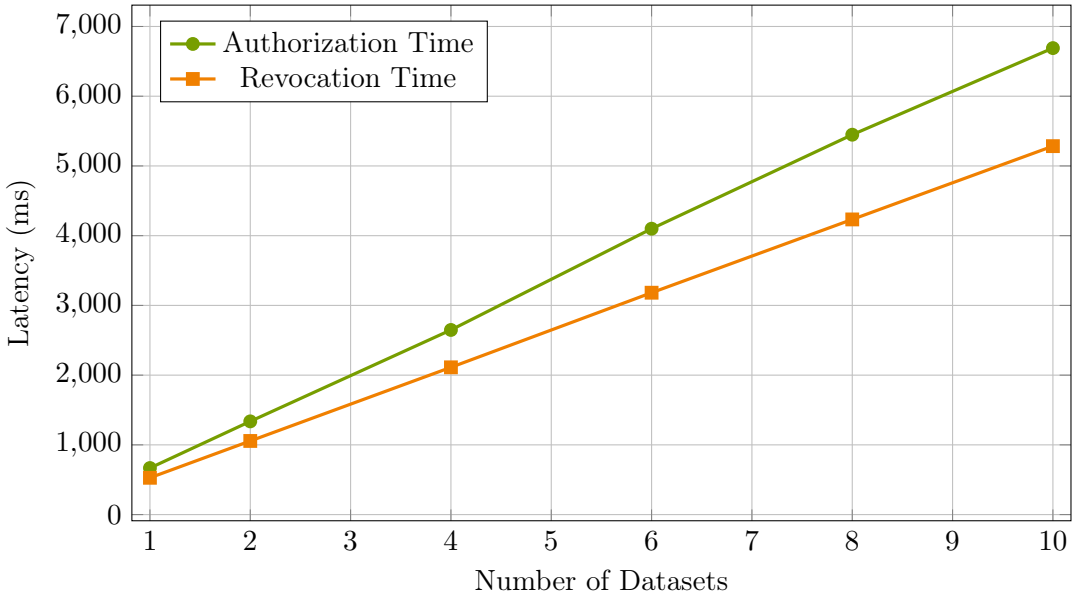


Figure 6.3: Comparison of Authorization and Revocation Times

constant when varying the number of consumers from 1 to 100. This is because revocation involves setting to 1 the bits corresponding to the indexes contained in the consumer credentials used to access the data. This introduces an average latency of approximately 27 ms, with no significant variations as the number of revoked users increases.

Moreover this experiments evaluate, the time for managing access revocation when the increasing number of datasets are issued by the same DID producer on the same topic. This involves the generation of an updated bitstring and the update of the access control registry on the blockchain. Table 6.1 presents the revocation times, which are illustrated in Fig. 6.3, showing the time required for revoking access while varying the number of datasets up to 10. For a single dataset, the entire process takes approximately 528 ms. As the number of datasets increases, the revocation time scales accordingly. This linear growth suggests that VESPACE exhibits predictable behavior, which is critical when a producer needs to revoke a consumer's access to multiple datasets simultaneously, ensuring a timely and secure update of credential status.

The results indicate that revocation is generally faster than authorization, with an average ratio of approximately 1.26, computed as the authorization time divided by the corresponding revocation time across different dataset sizes. Conversely, the inverse ratio represents the number of authorizations per revocation, highlighting

that revocation has a limited impact: its execution time is comparable to that of authorization. This ensures that the system can continue processing new access requests without significant delays.

The primary factor influencing revocation time is the upload of the bitstring to the blockchain. However, since revocations occur less frequently than authorizations, their overall impact on system performance remains minimal. Additionally, the system exhibits scalability as dataset size increases. As shown in Figure 6.3, authorization times scale proportionally with dataset size, while revocation times follow a similar trend but remain consistently lower. This demonstrates that the revocation mechanism remains efficient and does not introduce usability concerns.

Table 6.2: Analysis of Relationships and Timing between Authorization and Revocation

Number of Datasets	**Auth/Revoke Ratio**
1	1.26
2	1.26
4	1.25
6	1.28
8	1.28
10	1.26

6.5 Comparison with Previous Works

Existing solutions typically use DIDs and VCs for authentication, where VCs contain verifiable claims about an entity that are leveraged to grant access to data sharing platforms. However, these frameworks fail to fully embrace SSI principles, particularly when it comes to granting producers full control over their data. In most cases, credentials are issued by trusted third-party organizations, rather than the data owners themselves, thereby limiting the level of control producers have over their shared datasets. Moreover, many existing proposals overlook revocation entirely, and those that address it often rely on third-party components for this purpose.

In contrast, VESPACE directly addresses these gaps by enabling data owners to revoke access rights to their data in a fully decentralized manner. This is achieved through the W3C bitstring structure, which allows for efficient and direct access control updates by the producers themselves, ensuring full autonomy. To the best of current knowledge, VESPACE is the first solution to regulate access rights using this data structure, providing a groundbreaking approach to decentralized access control and revocation.

Table 6.3: Comparison of data spaces solutions based on function and non-functional requirements. The "✓" is used if the requirement is guaranteed, "∼" if it is partially met, "-" when it is not addressed, and "✗" if it is not guaranteed.

Requirement	[136]	[116]	[43]	[154]	[110]	[19]	[23]	[183]	**VESPACE**
User Management	∼	∼	✓	✓	✗	∼	✗	✓	✓
Data Sovereignty	✗	✓	✓	-	-	-	-	✓	✓
Discovery and Selection	∼	✓	✓	✓	∼	∼	∼	∼	✓
Data Access	✓	✓	∼	∼	∼	∼	✓	-	✓
Data Transfer and Exchange	✓	∼	✓	∼	✓	∼	∼	∼	✓
Data Interoperability and Portability	✓	✓	✓	✗	✗	✗	-	✓	✓
Auditing and Compliance	✗	✗	✓	✓	✓	✗	✓	✓	✓
Security and Privacy	✓	✓	✓	✓	-	∼	✓	✓	✓
Reliability and Usability	-	✓	-	✓	✓	-	✗	∼	✓
Verifiability and Trustworthiness	✗	✗	✓	✗	✗	-	✗	✓	✓
Scalability and Performance	✓	✓	✓	✓	✓	∼	∼	✓	✓
Auditability and Transparency	✓	✓	✓	✓	∼	✓	✓	✓	✓

Furthermore, while VCs are traditionally used by trusted issuers to certify claims about entities, VESPACE innovatively repurposes VCs to allow producers to self-certify the authenticity of their datasets in a decentralized manner by issuing self-signed credentials. This approach not only empowers producers with greater control over their data but also addresses data provenance, a critical concern that is often overlooked in many related works. By enabling producers to certify the authenticity of urban datasets, VESPACE enhances trust and transparency in data sharing within dataspaces, which is particularly valuable in contexts where the integrity and origin of data are crucial.

Finally, VESPACE overcomes a significant limitation in existing data stream platforms, which often fail to account for real-world scenarios where consumers may need access to both real-time data and historical datasets. Unlike traditional platforms that focus solely on live data streams, VESPACE ensures that previously shared data can be accessed in a secure, decentralized manner, offering a more flexible and scalable solution for data consumers and producers alike.

6.6 Discussion

data are increasingly recognized as a critical component for the functioning of Urban Digital Twins. One of the main obstacles is the lack of governance and sovereignty mechanisms for data exchange. To address this gap, VESPACE was introduced as a verifiable blockchain-based data space. The framework leverages DID, VC, and blockchain to enable decentralized governance and validation of data authenticity, while also supporting the revocation of access rights directly by data providers. From this perspective, data spaces provide an appropriate architectural response. They integrate governance and sovereignty principles directly into the data lifecycle, enabling controlled sharing, traceability, and long-term reproducibility of urban data.

A prototype implementation was evaluated with urban datasets, demonstrating that the proposed approach can efficiently support secure and verifiable data sharing. This demonstration confirms that data spaces are not an optional add-on, but a fundamental element of Urban Digital Twins. They ensure that UDTs can exchange information across silos in a sovereign, verifiable, and reliable manner, essential for their long-term sustainability and effective implementation in real urban contexts.

7 Federated Process Governance for Trustworthy UDTs

After addressing interoperability and verifiable data exchange through federated and trustworthy infrastructures, the next step regards the governance of the processes that operate on top of these data layers. As UDTs evolve into decentralized ecosystems, not only data but also the workflows that depend on them such as simulations, predictive analytics, or federated learning, must remain transparent, traceable, and accountable across multiple autonomous domains. This final challenge shifts the focus from trust in data to trust in the processes that continuously transform, aggregate, and act upon them, requiring mechanisms capable of monitoring their evolution and mitigating bias or malicious behavior.

7.1 TrustFlow: A Traceable Federated Learning Framework for Federated Data-driven Urban Digital Twins

To address the last challenges, this dissertation introduces Trustflow, a framework designed to ensure end-to-end traceability, trust, and accountability in federated UDT environments.

Trustflow extends the principles of trustworthy data federation to the process layer, enabling decentralized workflows such as federated learning (FL) and predictive analytics to remain auditable, verifiable, and resilient to adversarial behavior. Trustflow extends the principles of trustworthy data federation to the process layer, enabling decentralized workflows such as FL and predictive analytics to remain auditable, verifiable, and resilient to adversarial behavior. The adoption of FL within the broader context of distributed and edge infrastructures,initially demonstrated in the Industrial Internet Of Things (IIoT) domain (FIg. 7.1) has shown how scalable, privacy-preserving intelligence can be achieved while maintaining the fidelity and adaptability required by complex systems [137, 74, 141]. Applying these principles

to Urban Digital Twins allows different urban domains to collaboratively train models without sharing sensitive raw data, while preserving both data sovereignty and analytical performance.

Consider a UDT designed to optimize citywide traffic flows by integrating data from multiple districts. To preserve data sovereignty, local authorities and transport agencies do not share raw data, but instead train local models and periodically contribute their updates to a shared federated learning framework. Malicious participants may inject adversarial or poisoned updates to distort predictions, while others may produce low-quality models due to sensor malfunctions, incomplete datasets, or improper calibration. Furthermore, free-riding behavior, where some nodes benefit from the aggregated global model without contributing meaningful updates, undermines fairness and overall model performance. Such conditions highlight the need for a verifiable governance layer capable of monitoring, validating, and correcting contributions throughout the federated process. At its core, Trustflow leverages DIDs and VCs, anchored on blockchain-based smart contracts, to link every actor, dataset, and model within a federated workflow to a verifiable digital identity. This enables complete provenance tracking, ensuring that every data contribution, model update, and aggregation event can be traced back to its origin. Trustflow also introduces mechanisms for influence estimation and automated revocation. Each local contribution to a global model is evaluated in terms of its quantitative impact, making it possible to identify and isolate updates that degrade model quality or introduce bias. When such cases are detected, policy-based revocation can be triggered through on-chain smart contracts, which remove or invalidate the affected updates and their derived global models. These operations ensure that federated learning and simulation processes remain transparent and self-correcting, even in the presence of faulty or malicious participants.

7.1.0.1 Federated Learning

FL is a distributed ml framework that enables multiple clients to collaboratively train a shared ml model without directly exchanging their data [22]. The paradigm ensures that the raw data remain local on each client, and only model updates or gradients are transmitted to a central server for aggregation. This approach addresses privacy concerns, reduces communication costs, and enables collaborative learning across geographically distributed data sources [78]. A simple but effective aggregation algorithm in FL is Federated Stochastic Gradient Descent (FedSGD) [107]. In this approach, the server initializes the global model parameters θ^t at the start of

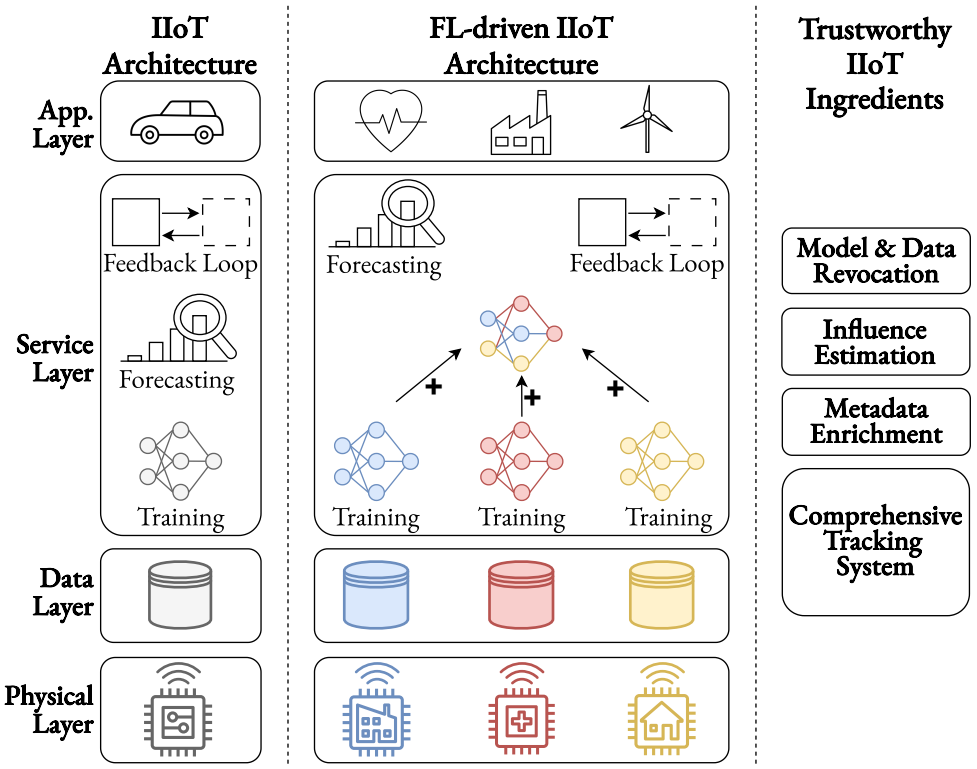


Figure 7.1: Bird's eye view of the FL-driven service architecture, showing both a (simplified) IIoT architecture and the key ingredients for building a trustworthy DT. At the top lies the Application Layer, which includes various verticals (e.g., industry, smart cities, healthcare, etc.) that leverage the functionalities provided by the Service Layer to obtain relevant insights about the monitored entity and potentially trigger adaptations to the process or system. The Service Layer, in turn, depends on the Data Layer to access real-time data streams that mirror the behavior of the process or system. These data streams are used to train data-driven models and pipelines.

round t. Each participating client k computes the gradient of its local loss function, $\nabla L(\mathcal{D}_k, \theta^t)$, based on its local dataset $\mathcal{D}_k$. The server aggregates the gradients from clients using a weighted average based on their dataset sizes, and the result is used to update the global model's weights using the standard stochastic gradient descent algorithm. This iterative process ensures that the global model updates reflect the contributions of all clients. However, a problematic update could negatively impact the global model, leading to poor performance or even degradation of the model's accuracy. Since updates from multiple clients are aggregated, a single biased or erroneous update can skew the global model parameters, especially when the update is from a client with faulty or malicious data. The misbehavior of the potential client highlights the importance of improving the traceability and observability of the FL training process. Tracking individual clients, estimating the influence of both clients in providing their local update models and of data exploited to train models, is crucial to understanding the contributions to the global model, thus increasing the quality, reliability, and explainability of the process. Despite various solutions that have been proposed to address the specific issue of client influence, many existing FL frameworks overlook and do not adequately incorporate influence estimation as a building block.

In the context of DT ecosystems, where accurate and reliable models are critical for real-time decision-making, the estimation of client influence on the process is considered paramount. For this reason, this framework integrates this capability as a core part of the training process, supporting the estimation of client-level and data-level influence.

7.1.1 DIDs for Trustworthy fl

Regarding the integration of FL with DID, various solutions take advantage of technologies to improve privacy, security, or decentralization. DID-eFed [57] proposes a FL as a Service system using DID and smart contracts for flexible access management. Similarly, Goh *et al.* [60] developed a blockchain-enabled FL architecture that incorporates DID for access control. The authors in [125] propose a decentralized privacy-preserving workflow for trustedfl, leveraging decentralized identity technologies from Hyperledger. The work focuses on a medical use case, requiring that only participants with VC issued by authorized entities can securely and verifiably participate in processesfl. Zeydan *et al.* [46] propose an identity management solution for vehicle users in FL. Using DID, this method ensures the confidentiality, authenticity, and integrity of user identities and data during FL processes. These approaches

aim to improve data privacy, security, and traceability in FL while addressing key challenges such as participant identity verification and data accountability. However, this body of work neglects important system features for model influence estimation and revocation. These features are essential to preserve the integrity of the global model and mitigate the impact of compromised updates.

7.1.1.1 Influence Estimation

Regarding the estimation of influence infl, different methods have been proposed for use in centralized or federated contexts. *Leave-One-Out* (LOO) and *Influence Functions* (IFs) [82] have been conceived for use in centralized settings. LOO evaluates the contribution of clients by retraining the model while excluding specific datasets and observing performance changes, but it is computationally impractical infl [82]. IFs approximate the effect of removing data points without retraining, using model parameters and second-order derivatives of the loss function. *TracIn* [130], another centralized approach, estimates the influence of a data point z (having input x and label y) by evaluating its impact on model loss throughout training. Unlike LOO, which requires retraining, or IFs, which depend on second-order derivatives computations, TracIn leverages gradient information from multiple model checkpoints to provide an efficient and scalable estimate. There is an ongoing research effort to estimate the influence of the client in a FL setting, and various alternatives have been proposed. Xue *et al.* [179] introduce the concept of *Fed-Influence*, a metric based on a leave-one-client-out approach to quantify the influence of individual clients. They propose an efficient and effective algorithm to estimate this metric. However, the algorithm relies on the Hessian approximation, which may still be impractical for devices with limited computational resources. More recent approaches like Wang *et al.* [171] use Shapley values [175], a well-known concept from cooperative game theory, to calculate the average marginal contribution of a client to the model, considering all possible combinations of clients. However, the high computational cost and the additional steps required make the approach impractical for large-scale or real-time applications in FL. Gill *et al.* [59] propose Trace FL, a method to improve transparency in FL by tracking individual neurons' contributions to the global model. This helps identify errors and biases, providing valuable insights for debugging. However, the approach introduces significant computational overhead as tracking neuron contributions requires additional resources and steps.

Although numerous studies focus on estimating a client's contribution, many adopt a coarse-grained approach, assessing the influence of a client as a whole rather than

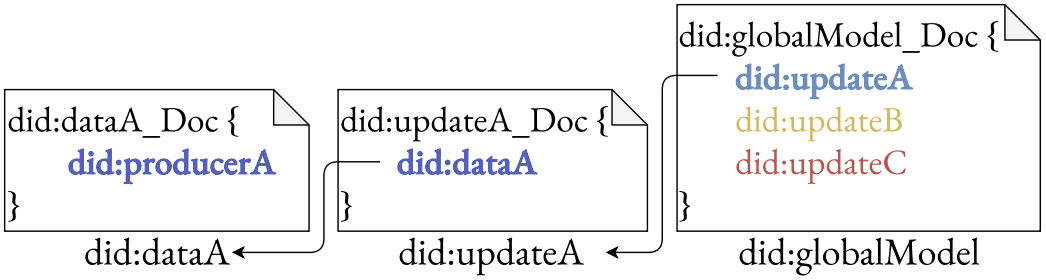


Figure 7.2: DID Documents and their associated metadata enable tracing back to the sources.

evaluating the influence of individual datasets, limiting the ability to pinpoint problematic data. Conversely, Trustflow integrates support for both client and individual data influence estimation and enables the revocation of both data or models, therefore helping to maintain the integrity of the global model and preventing the spread of faulty information.

7.2 Trustflow

In this section, a framework is presented that incorporates a comprehensive tracking system based on DIDs and VCs to uniquely identify process entities, datasets, updates, and global models within FL-driven DTs. Trusflow enriches these elements with specific metadata, such as data collection procedures, to ensure the quality and trustworthiness of the process. In addition, it incorporates an automated, policy-driven revocation mechanism for data and global models, enabling proactive retraining management in the case of flaws or bias in data contributions or local model updates. Trustflow can be seamlessly integrated into any FL-driven architecture as a governance component, placed primarily within the Service Layer as evidenced in Figure 7.1.

Without loss of generality, it is assumed that adversaries act as malicious clients within the federated learning system, attempting to poison the global model by submitting manipulated updates. These adversaries have access to their own local data and are aware of the model architecture and training protocol. However, they do not have access to the data or updates of other clients. Although adversaries can participate in training and submit arbitrary or malicious updates, the central server is assumed to be honest and secure. Attackers cannot alter, compromise, or manipulate the server, which reliably aggregates updates without leakage or alteration.

These assumptions are widely accepted and grounded in standard security models commonly used in FL research [24, 89, 100].

7.2.0.1 Enabling Traceability in Federated Learning

A key feature of this framework is the comprehensive traceability of every element of the system, including data, actors, and models. This is achieved through the use of DID technology, with each component of the system having a valid DID stored on the DLT. It is assumed that the system operates either under a permissionless autonomous registration phase or under a trusted authority managing participant selection in a permissioned manner. Each DID is also associated with a corresponding DID Document, which contains important information about the entity referenced by the DID. Figure 7.2 illustrates how DID Documents are used to link the various entities involved in the process.

Data Producers: Data producers generate data. They can be users of smartphones, smart city sensors, hospitals, industrial machinery, or other IoT devices. Producers can choose to remain completely anonymous, disclosing no information, or, conversely, provide crucial details to ensure clear identification. Any information about a producer is saved in its DID Document. For example, a public entity, like a hospital, might prefer to be easily recognizable to guarantee that users can trust the origin of the data.

Data: The data generated by a producer is linked to a unique DID, allowing unambiguous reference. The corresponding DID Document includes information about the producer, using the DID controller field, and other details regarding how the data was generated. As mentioned above, one of the most critical aspects of creating accurate modelsFLis to ensure that the data are produced and handled accordingly to minimize potential biases or errors. In the datasets DID Document, the producer can outline the procedures followed to ensure data integrity, such as those recommended by the EU Agency for Fundamental Rights [51]. Trainers use this information to guide their data selection. For example, a trainer might select only data from recognized or globally trusted sources. Additionally, a trainer might opt for data produced according to specific guidelines or data that has gone through particular quality checks performed by the producers. To be valid, each data must have an associated valid VC signed by the producer specified in its DID Document. The server retains part of the data, comprising the test dataset, used by the server to calculate the influence of individual clients on the global model. This dataset also has DID and VC; all used in the event of disputes.

Model Trainers: Model trainers use data to train a ml model. It is assumed that a trainer can be either the producer of the data it uses or a third-party entity that utilizes data from one or more producers. In the latter case, selecting the appropriate data for training is a crucial step to ensure that the resultingFLmodels are as accurate and reliable as possible. In the first case, the trainer and the producer coincide, so the DID and the DID Document of the trainer are the same as that of the producer. However, in the second case, trainers should have a unique DID stored on a DLT to be identifiable. The DID Document may also include information to identify the actual entity behind the DID.

Clients Updates: Local models are trained by the trainers participating in theFLprocess using selected data, producing an update. The latter is identified by a DID on the DLT published by the corresponding trainer. The associated DID Document contains information related to the training, such as the optimization algorithm used, the number of local epochs, and other metadata. In addition, the DID of the data used to train that model is also specified. This ensures that it is always possible to trace which data were used to train a local model and produce the specific update.

Server: The server performs the standard operations of a traditionalFLserver, along with additional specific tasks to manage the influences of the data and the revocation mechanism. It has a DID on the DLT and the corresponding DID Document. Typically, the server is a trusted third party that enables theFLprocess.

Global Model: The global model is generated by aggregating updates from various trainers. The server generates a DID and publishes it in the DLT along with its corresponding DID Document. This document includes metadata about the global model, such as the aggregation algorithm used and the did of the local models involved in the aggregation process. This ensures full traceability of which models and, by extension, which data were used to create the global model and influence its development. This process is carried out during each round, providing a complete view of how the global model evolves and which data have played the most significant role in shaping it. To be valid, each global model must have an associated VC issued by the server that created it.

DLT: The DLT stores the process information, guaranteeing its immutability and traceability during its lifecycle. It stores the DID and their respective DID Documents to provide full transparency of theFLprocess and accountability for each entity. In addition, a smart contract maintains a detailed record of how each dataset influences the global model. Table 7.1 illustrates an example of how influences can be recorded. For each global model generated at the end of a training round, the

Table 7.1: Influence table stored on the smart contract.

Data	**Global Models**		
	DID:globalModel^1	...	DID:globalModel^N
DID:dataA	-2	-1	-3
DID:dataB	+2		
DID:dataC	-1	+4	+2

DID of the data used by the local models are listed, along with their corresponding influence values. Since not all trainers participate in every round, influence data may be missing for trainers who did not contribute during a specific round. By systematically tracking these influences, it becomes possible to enforce policies for revoking data or models when necessary.

7.2.0.2 Influence Estimation

Trustflow framework uses the TraceIn [130] algorithm to assess the influence of a client, adapting it to a federated distributed context. This algorithm, originally designed for centralized learning, is efficient and allows fine-grained analysis of specific datasets, enabling the identification of problematic data when clients send multiple gradients. In a centralized context, the algorithm works as follows. Given a training dataset $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}\}$, let $\theta^{(t)}$ denote the model parameters in the training iteration t. The loss function for a data point z_i is represented as $L(z_i, \theta) = \ell(f_\theta(x_i), y_i)$, where f_θ is the model, and ℓ is the loss function.

The influence score for a data point z_i on a reference data point z' is given by:

$$Influence(z_i, z') = \sum_{t \in \mathcal{C}} \eta_t \nabla_\theta L(z_i, \theta^{(t)}) \nabla_\theta L(z', \theta^{(t)}) \tag{7.1}$$

where:

- $\mathcal{C}$ is the set of selected checkpoints during training,
- η_t is the learning rate at iteration t,
- $\nabla_\theta L(z, \theta^{(t)})$ is the gradient of the loss function with respect to the model parameters θ at checkpoint t.

The influence score aggregates the dot products of the gradients at each checkpoint, capturing the cumulative effect of z_i on z'. A positive influence score indicates that

z_i contributes to reducing the loss for z', meaning z_i is helpful in correctly learning z'. A negative influence score implies that z_i increases the loss for z', suggesting that z_i may introduce noise or conflict. To adapt TracIn to Trustflow framework, the server calculates the influence at the end of each round, using the global models produced at the end of the previous round as checkpoints. As in FedSGD, each client sends its update to the server after local training. This update is the gradient of the loss function with respect to the global model $\theta^{(t-1)}$ from the previous round, computed using the client's local data $\mathcal{D}_{Client}$, i.e., $\nabla_\theta L(\mathcal{D}_{Client}, \theta^{(t-1)})$. The server uses this value as the first term of Equation 1. A client with multiple datasets can send a single update or N updates, one for each dataset. In the first scenario, the influence is calculated collectively for all datasets and is equally attributed to each. In the second scenario, the server calculates the influence of each dataset.

Regarding the comparison data z', the server has a small dataset to compute the second term of Equation 1, i.e., $\nabla_\theta L(\mathcal{D}_{Test}, \theta^{(t-1)})$. This ensures consistency between all clients, since they are all evaluated using the same data, providing a more uniform measure of how each client's data influences the global model. The practice of retaining such a dataset is well established in theFLliterature [180, 152, 101, 99], where it is commonly used for evaluation, debugging, or supporting specific computations.

Next, the server applies the TracIn formula to calculate the influence of each client's update. The influence score I_{Client} for a given client is:

$$I_{Client} = \nabla_\theta L(\mathcal{D}_{Client}, \theta^{(t-1)}) \nabla_\theta L(\mathcal{D}_{Test}, \theta^{(t-1)}) \tag{7.2}$$

This process is done for each client, and the server then publishes the influences on the DLT through the deployed smart contract. This process introduces minimal overhead, as client updates are already part of theFLprocess. The only extra step is for the server to compute the gradients on a smaller test dataset, resulting in negligible additional cost.

7.2.0.3 Revocation Mechanism

The Trustflow decentralized revocation mechanism is tightly integrated with the tracking system, using DID, VC, and a smart contract to ensure complete traceability and effective model management. This mechanism is governed by a smart contract that periodically monitors the recorded influences in the system to detect potential issues, applying predefined policies to identify problematic data and models. Algorithm 4 provides the pseudocode for an example smart contract. The

Algorithm 4 Revocation Mechanism for VCs

Global:
  $T[r][g]$: influence score matrix
  B_r: bitstring for resource VC revocation
  B_g: bitstring for global VC revocation
Function Revoke($g, \{r_1, ..., r_n\}$)
Input:
  g: global model
  $\{r_1, \ldots, r_n\}$: DIDs of resources linked to g
Output:
  Revocation status of VC_g
for each r_i in $\{r_1, ..., r_n\}$ **do**
  if $T[r_i][g] < 0$ **then**
    Extract CID in DID Document of r_i
    Compute bit index idx_r from CID
    Set $B_r[idx_r] \leftarrow 1$
    Revoke VC_{r_i}
  end if
end for
return RevokeGlobalVC($g, \{r_1, ..., r_n\}$)
Function RevokeGlobalVC($g, \{r_1, ..., r_n\}$)
$T_g \leftarrow$ number of DIDs associated with g
$R_g \leftarrow$ number of resource bits set to 1 in B_r for $\{r_1, ..., r_n\}$
if $R_g/T_g > \phi$ **then**
  Extract CID from DID Document of g
  Compute bit index idx_g from CID
  Set $B_g[idx_g] \leftarrow 1$
  Revoke VC_g
end if
return revocation status of VC_g

revocation mechanism in Trustflow is based on an influence-aware trust assessment that operates throughout the federated learning rounds. During each round, the system updates an *influence matrix* $T[r][g]$, where each entry $T[r_i][g]$ quantifies the contribution (positive or negative) of a data resource r_i to a global model g. This matrix is computed continuously during the learning process and serves as a foundational structure to identify underperforming or malicious contributions, allowing fine-grained control over the trustworthiness of each component in the system. Once the model training round is complete, the smart contract executes the `Revoke` function (lines 5-10). This function takes as input the global model g and the set of associated resource DIDs $\{r_1, \ldots, r_n\}$, and performs a two-stage evaluation.

In the first stage, for each resource r_i, the system checks whether its influence score $T[r_i][g]$ is strictly negative (lines 11-12). If so, the system considers the resource untrustworthy and proceeds to revoke its associated VC_{r_i}. Technically, this is done by setting to 1 the corresponding bit in the global bitstring B_r, which maintains the revocation state of all resource credentials (lines 13-16). The position of the bit is computed by extracting the CID stored in the `serviceEndpoint` field of the DID Document associated with r_i. This approach ensures that revocation is transparent, tamper-proof, and efficient, as it avoids the need to remove the VC entirely and instead marks its invalidity verifiably.

In the second stage, the algorithm assesses whether the global model g should also be revoked. It counts how many of the associated resource VCs have been revoked (i.e., how many corresponding bits in B_r are set to 1). If the proportion of revoked resources exceeds the threshold ϕ (line 23), the system considers the global model compromised. Currently, the threshold is empirically set to 0.5, as higher values can prevent the model from converging [29]. The corresponding bit in the global model bitstring B_g is also set to 1 (lines 24-27). At this point, the bit index is computed from the CID in the `serviceEndpoint` field of the DID Document associated with g, which points to the VC of the global model saved on IPFS.

To ensure interoperability and verifiability, bitstrings B_r and B_g are managed on-chain and follow the W3C Status List 2021 specification [167]. This design enables efficient revocation proofs and ensures consistency across decentralized components. By combining continuous influence tracking, structured trust evaluation, and decentralized bitstring-based revocation, Trustflow enables granular, transparent, and scalable trust management of the underlying DT model. It allows for selective deactivation of harmful resources or outputs while preserving valid contributions, ensuring both security and continuity in decentralized AI workflows.

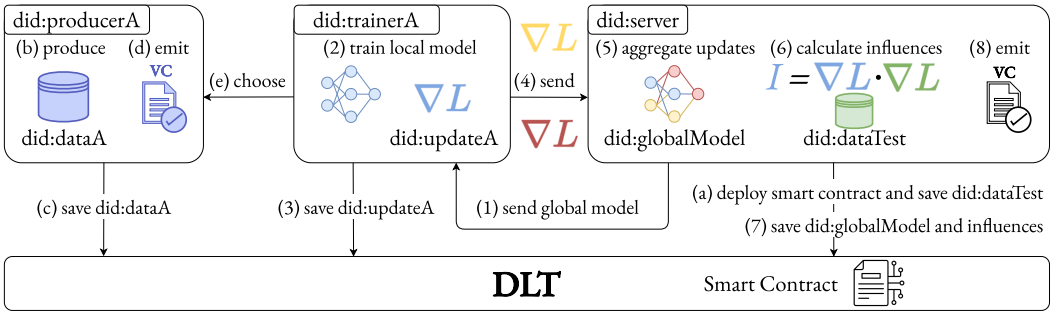


Figure 7.3: Trustflow during the Preparation phase, (a)-(e) steps, and the traditionalFLProcess, (1)-(8) steps.

7.2.1 Trustflow at Work

A predictive maintenance scenario is considered, in which sensors, machines, and servers within a manufacturing environment collaboratively participate in creating an FL-driven DT capable of predicting failures and optimizing operations. The main execution flow can be divided into three main phases: (I) Preparation, (II)FLProcess, (III) Revocation. Figure 7.3 depicts phases (I) and (II).

7.2.1.1 Preparation

The first phase, steps (a)-(e) of Figure 7.3, focuses on preparing all the essential elements required for the process. In step (a), the server, which could be a smart manufacturing server or an industrial IoT platform, deploys a smart contract on the DLT that allows recording the influence of the data on global models. The contract also implements the policies for the revocation of the data and the model. In addition, the server saves the DID related to its testing data. A producer, such as a smart sensor or an industrial machine, collects the data and makes it available (b). The producer publishes a DID to identify the dataset (c) and issues a unique VC (d) certifying the validity of the dataset. In step (e), a trainer, which could be an edge device or a local server, selects one or more datasets based on specific requirements. For example, a trainer might prefer datasets from specific sensors or machines. These steps could also be performed before or simultaneously with the steps executed by the server.

7.2.1.2 FL Process

The second phase, steps (1)-(8) of Figure 7.3, integrates the standardFLworkflow with additional operations to track models and their associated metadata. The-

FLprocess begins with the server sending the initial global model (1). The trainer uses the selected datasets to train a local model (2) and produce an update. The trainer then publishes (3) a DID for the newly generated update and sends (4) it to the server. The server collects and aggregates all client updates (5) to create the global model and uses them to calculate (6) the influences on the global model. The server accepts only updates produced by models trained using data with valid VC. It then publishes (7) a DID to identify the global model, records the influences using the smart contract, and issues (8) a corresponding VC to certify the validity of the global model. The server subsequently sends the global model back to the clients to continue theFLprocess.

7.2.1.3 Revocation

The revocation flow, described in Section **??**, is executed when it is discovered that the data used in the training negatively impact the model. Every time the server publishes new influences on the smart contract, the contract applies a policy, like the one presented in Algorithm 4, to determine whether any data or models should be revoked. If necessary, the server can either reinitialize the training from scratch, rejecting updates from revoked sources, or restore the model weights to a state before the faulty data was aggregated if it had saved them. Overall, the proposed framework introduces minimal overhead to theFLprocess, as shown also in the experimental part.

7.3 Experimental Evaluation

This section presents the experiments conducted to validate the proposed approach. The implementation setup is first described, followed by an overview of the conducted experiments. Finally, the performance indicators are analyzed.

7.3.1 Experimental Strategy

The following analysis focuses on evaluating the accuracy of the influence estimation algorithm, the latency of key tracking operations, and the overhead introduced by the revocation mechanism. Since Trustflow does not modify the FL protocol, the accuracy of the global model remains unaffected and is therefore not reported here. Instead, the focus is placed on metrics directly relevant to the objectives of the framework. Detecting detrimental clients and their influence is crucial to ensuring

the reliability of the global model while minimizing tracking latency. Consequently, accurate estimation of both the data and the influence of the client is fundamental.

Moreover, enabling fast revocation of data and models when flawed contributions are detected is necessary to maintain the integrity of the training process. In particular, the experimental evaluation is conducted on operations that include the creation of DIDs and DID Documents, the storage of these DIDs and DID Documents on the blockchain, and the issuance of VCs. Moreover, the time required to store VCs on the IPFS decentralized storage is evaluated. The objective is to assess how effectively the proposed method detects detrimental contributions, rapidly revokes problematic data and associated models, and ensures that the overhead introduced by influence calculations and interactions with the DLT does not significantly affect the overall FL system performance.

7.3.2 Experimental Setup

Table 7.2: Influence score estimation in different settings.

Setting	**Clients**	**IID**				**non-IID**			
		Last Checkpoints			**Total Sum**	**Last Checkpoints**			**Total Sum**
5 Clients	4 Correct	1292.83	867.68	622.05	2782.57	1198.90	770.56	890.24	18381.10
	1 Faulty	-233.70	-863.92	14.39	-1083.23	118.07	-912.61	-738.47	3518.47
10 Clients	9 Correct	4144.13	213.64	67.34	4867.81	835.50	1142.98	1088.56	15551.82
	1 Faulty	2016.46	0.55	-164.02	2142.62	-981.03	-397.65	-721.70	-6206.53
	8 Correct	2074.12	284.41	88.34	2639.06	1737.32	1568.03	1238.61	13201.61
	2 Faulty	-359.04	-144.63	-354.06	-2114.39	-951.57	-958.97	-418.34	-3467.97
20 Clients	19 Correct	3605.32	663.47	455.97	6024.43	374.74	398.19	402.11	7160.18
	1 Faulty	-370.24	-449.91	398.66	1429.65	-292.61	-605.11	-1242.62	-4698.83
	18 Correct	3726.20	915.30	818.87	5809.99	287.25	214.60	156.58	8600.07
	2 Faulty	-283.83	384.12	-243.95	1038.18	-1818.28	-1467.94	-1222.87	-8632.68
	16 Correct	6271.37	803.36	275.62	7350.36	411.20	232.23	365.07	15584.95
	4 Faulty	613.95	-1282.07	-726.26	-1394.39	-1473.78	-1002.70	-1661.37	-8467.47

The Trustflow mechanisms are evaluated under different configurations and with a varying number of clients (5, 10, 20) participating in the FL process. Furthermore, different percentages of faulty clients are considered: 5% (with 20 clients), 10% (with 10 and 20 clients), and 20% (across all client numbers).

These configurations are selected to reflect realistic scenarios, such as smart industry ecosystems relying on predictive maintenance models or hospitals implementing advanced smart healthcare solutions, where the chosen number of participating clients represents a plausible scale.

To simulate the effect of biased or incorrect data, client datasets are modified through label flipping [77], a common approach to introduce noise in classification

tasks. For the ML model, MobileNetV2 [143] is employed as the base architecture, optimized using Adam with a learning rate of 1e-3.

Local training is conducted over 5 epochs with a batch size of 32. The evaluation uses the CIFAR-10 [83] dataset, a well-established benchmark in the ml community, with both iid and non-iid data distributions. The non-iid setting is simulated using a Dirichlet distribution with $\alpha = 0.5$. Regarding the DLT, the work is based on Ethereum due to its robust support for decentralized, programmable smart contracts through Solidity [176], as well as its maturity, ease of programmability, and strong ecosystem support. This choice was driven by the need for a secure, transparent, and immutable infrastructure capable of automating the complex interactions required for revocation while ensuring accountability. The flexibility of Solidity enables the design of custom logic for managing updates to credential status. For local development, the Truffle framework is employed in conjunction with a simulated Ethereum blockchain, such as Ganache, which has also been adopted in similar works within the FL domain [37, 149]. This setup enables the efficient testing and deployment of smart contracts within a controlled environment. Specifically, Web3.js was employed to facilitate interaction with the blockchain, enabling both the deployment and management of smart contracts. To create DIDs, DID Documents, VCs, and bitstrings, the Digital Bazaar library is employed. Finally, IPFS is employed to store the VC.

7.3.3 Performance Results

7.3.3.1 Influence Score Estimation

Table 7.2 shows the trend of the calculated influence in different settings and clients. For simplicity, the table shows only the averages computed for the correct clients, i.e., those without erroneous data, and faulty clients, i.e., those characterized by flipped labels. Since each setting required a different number of rounds and more clients needed additional rounds to converge, only the influences from the last three checkpoints are explicitly reported (*Last Checkpoints* columns). The *Total Sum* column shows the sum of the influences, including those from the non-displayed checkpoints. The data indicates that faulty clients tend to contribute negatively to the global model's quality of the global model in both iid and non-iid settings. Their influence is generally negative or lower than that of the correct clients. However, in some cases, the influence score of faulty clients can be positive. This occurs because these clients still possess a portion of correct data, i.e., data without label flipping, which contributes positively to the model's classification ability. For example, in a scenario with 20 clients, where only one is faulty, splitting CIFAR-10 into 20 parts results in

the faulty client holding a relatively small amount of corrupted data. Consequently, the data remain insufficient to significantly degrade the model's overall performance. An interesting observation in the iid setting is that as the proportion of faulty clients increases, their influence score decreases and eventually becomes negative. This indicates that the faulty data contribute very little to improving the model's performance. If the global model is generated from several faulty updates, the gradient computed by the server on the correct test data attempts to compensate for the errors introduced by the faulty data during training. As a result, this gradient differs significantly from the ones provided by faulty clients, showing how their contributions are less useful and eventually harm the performance of the global model. The non-iid case exhibits greater variability, as some clients may have a smaller or even negative influence despite not containing faulty data. This is not surprising, as the contribution of each client depends on the proportion and distribution of their data. For example, clients with smaller datasets will contribute significantly less than those with larger datasets. In addition, the results consistently show that faulty clients have lower influence scores, often falling into negative values, compared to correct clients. In these experiments, each client is assigned a single dataset for simplicity, so the influence score is assigned to the entire client. If a client were to possess multiple datasets, it would send multiple gradients and the influence score would be computed separately for each dataset. In any case, the shown results remain valid as the influence trends remain the same.

7.3.3.2 Latency Analysis

Table 7.3 shows the latency distribution for the indicated operations, i.e., DID Generation, Save DID on DLT, VC Generation and Upload VC to IPFS. It reveals that all operations exhibit stable latency across different client groups. In general, system performance remains consistent regardless of the number of clients, indicating that operations are not significantly affected. This result suggests that the system can effectively handle an increasing number of clients without introducing significant delays in operations. Figure 7.4 compares the average execution time required to perform local training with the time taken for operations related to the creation and storage of DID and VC in different settings. The data indicates that the time required to execute the DID and VC operations is constant, as seen in Table 7.3, and negligible compared to the neural network training, demonstrating the minimal impact of Trustflow framework on client performance. It should be noted that training time decreases as the number of clients increases, as each client receives a smaller

Table 7.3: Average Latency Times for Different Operations and Client Groups in Milliseconds

Operation	**5 Clients**	**10 Clients**	**20 Clients**
DID Generation	1.43	1.44	1.37
Save DID on DLT	198.80	197.67	197.68
VC Generation	864.74	876.23	882.98
Upload VC to IPFS	22.03	24.28	21.29

portion of the dataset, leading to faster training. However, this result is specific to this controlled setup with a fixed-size dataset. This trend may not be observed in real-world scenarios where client data are not partitioned artificially.

Figure 7.5 shows the latency in seconds for the various settings in a challenging scenario where all the VC of data and global models must be revoked due to non-compliance. The results show that as the number of clients increases, the latency of the revocation process increases significantly. This increase in latency is due to the need to manage a growing number of data and global models to revoke, which implies greater computational complexity and more time required to complete the operation. Nevertheless, it is important to highlight that the evaluation considers the worst-case scenario, which rarely occurs in practice, as revocation is typically triggered only in specific situations such as the detection of non-compliant or malicious behavior. As a result, its infrequent execution ensures that the system remains efficient during standard operations.

7.4 Discussion

In UDT ecosystems, where data are continuously generated by distributed sources,FL plays a key role in enabling collaborative and privacy-preserving model training. This approach enables UDT to adapt and evolve in real-time without compromising data privacy. As a result, ensuring the trustworthiness, integrity, and reliability of FL processes becomes essential, demanding robust observability to monitor, analyze, and optimize system performance. In this direction, Trustflow is introduced as a robust tracking framework that leverages DLT, DID, and VC to achieve comprehensive traceability in federated learning processes. Trustflow enables one to gain up-to-date information on client participation and model performance, fostering quality-aware, adaptive, and trustworthy learning processes. Trustflow operates independently of the specific FL algorithm, ensuring broad applicability across various implementa-

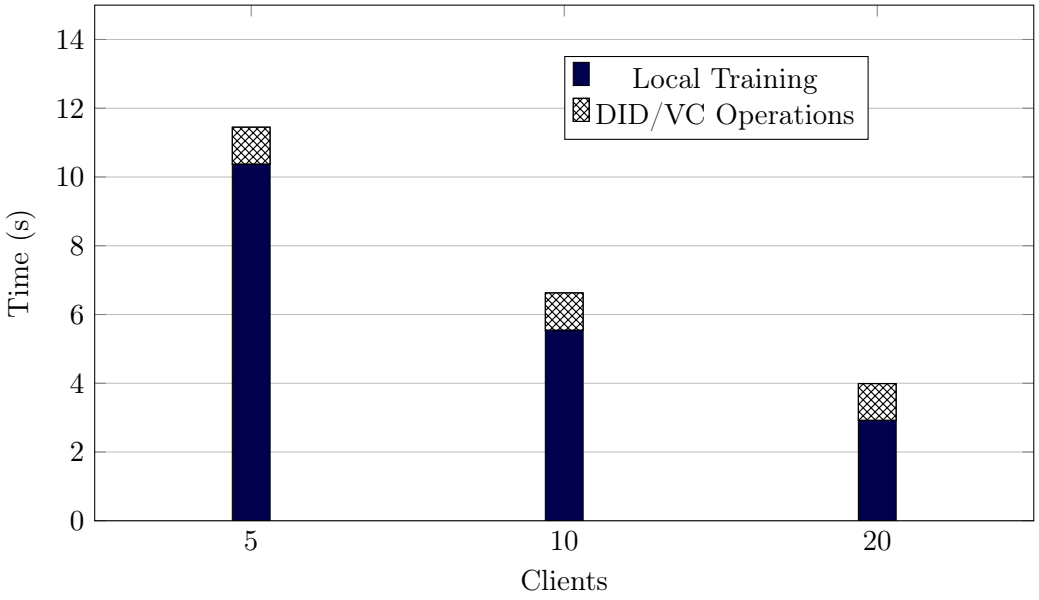


Figure 7.4: Average time (in seconds) required for local training and DID/VC operations, with varying numbers of clients.

tions. An original feature of Trustflow is its integration of automated revocation mechanisms, which enable the invalidation of erroneous or malicious contributions, thereby safeguarding the integrity of the global model. In addition, it introduces minimal overhead to the FL process, preserving efficiency. Additionally, future developments will evolve Trustflow towards a more decentralized architecture that minimizes or eliminates reliance on central servers, thereby aligning more closely with the principles of distributed FL. Another direction involves relaxing some of the assumptions made in the current threat model to better reflect real-world adversarial conditions and improve the robustness of the framework in more practical deployments. These advancements will further strengthen Trustflow's role as a comprehensive and practical solution for trustworthy and transparent fl-driven DT systems.

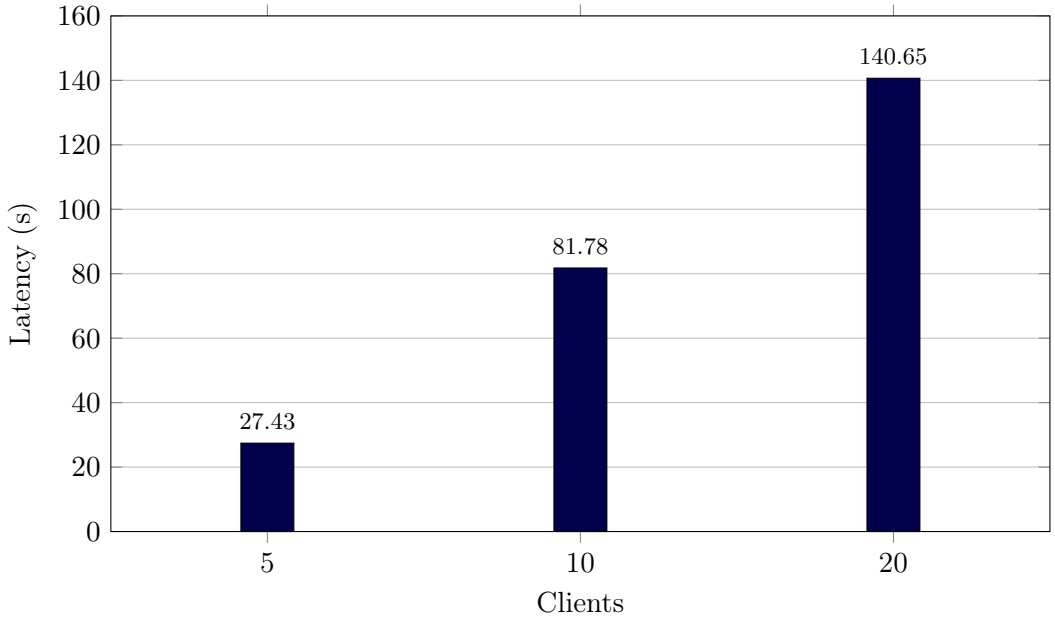


Figure 7.5: Latency (in seconds) of the revocation mechanism for 5, 10, and 20 clients.

8 Conclusion and Features works

This thesis examined how UDTs can evolve into coherent, interoperable, and reliable digital infrastructures through a progressive sequence of five interconnected studies. These studies address some of the obstacles to the development of reliable and sovereign urban ecosystems, covering semantic interoperability, temporal reasoning, verifiable data federation, and reliable process governance, collectively strengthening the conceptual and technical foundations of UDT ecosystems.

The first contribution introduced a multi-faceted interoperability model based on MIMs and serverless computing. This work proposed a lightweight, vendor-neutral architecture that enables elastic data processing and cross-domain integration through minimal, shared mechanisms. The adoption of MIMs allows heterogeneous systems, developed independently across different urban sectors, to communicate seamlessly while maintaining autonomy and compliance with domain-specific rules. By combining these mechanisms with the serverless paradigm, the proposed architecture ensured the scalability and composability of urban data pipelines in distributed environments. This first study established the technical foundation for interoperable UDT infrastructures, directly aligning with European initiatives that promote open, reusable, and standard-based digital ecosystems.

Building on the foundational structure of GlassBox, the second contribution repositioned the model as a semantic backbone for Urban Digital Twins, extending its original simulation-oriented logic into a modular and ontologically interpretable framework. While syntactic interoperability enables data exchange, semantic alignment establishes a shared understanding across heterogeneous urban subsystems. Within this perspective, the GlassBox entities were reconceptualized as components that can be semantically described, annotated with temporal metrics, and linked to heterogeneous data flows and decision logic.

Rather than treating interoperability as a matter of format compatibility, this contribution conceptualised it as a structural property of the digital twin infrastructure, where traceability, consistency, and interpretability emerge from the explicit association of metadata, rules, and metrics to each model component. This shift enabled a level of semantic observability, where urban data is not only transmitted but can

be interrogated, validated, and reused coherently across domains such as mobility, energy, and environmental monitoring.

Through this reformulation, the GlassBox model evolves from a simulation engine into a semantic mediation layer capable of aligning simulation logic with real-world urban data streams. This provides the groundwork for verifiable and interpretable UDT processes, supporting both cross-domain reasoning and human-in-the-loop oversight.

The third contribution extended this conceptual backbone into the temporal dimension through the introduction of TDCs. While GlassBox focused on structural and semantic modularity, TDCs addressed the problem of diachronic data management. This issue concerns the representation of entities and datasets as they evolve over time. Cities are inherently dynamic systems, administrative boundaries change, infrastructures expand, and datasets are collected under different methodologies and resolutions. Traditional models fail to capture these transformations, relying on static or snapshot-based representations. The TDC element introduced rule-driven mechanisms to encode entity transformations such as splits, mergers, and reclassifications, together with multidimensional quality metrics evaluating spatial precision, temporal granularity, and semantic completeness. TDCs provided a practical environment for reconstructing historical urban configurations and assessing dataset reliability over time. This contribution established the temporal interoperability layer of the dissertation's framework, enabling consistent longitudinal analysis and quality-aware decision-making in evolving urban contexts.

After establishing the technical, semantic, and temporal foundations, the dissertation turned to the challenge of data federation and trustworthiness in distributed UDT ecosystems. As cities increasingly rely on data from autonomous organizations, traditional centralized models become infeasible and incompatible with principles of data sovereignty. The VESPACE framework was introduced as a verifiable blockchain-based data space solution that integrate the concepts of SSI, DIDs, and VCs. VESPACE ensures that every dataset, transaction, and actor within a federated environment can be authenticated, traced, and audited without relying on intermediaries. Data providers retain full control over access policies, revocation, and provenance, while all certification events are immutably recorded on blockchain. Through its prototype implementation with open data from the Municipality of Bologna, the framework demonstrated how verifiable data exchange can be achieved across domains without compromising sovereignty or privacy. VESPACE thus established the federation and verifiability layer of the overall architecture, bridging

the principles of interoperability and trust within the paradigm of European Data Spaces.

Finally, the dissertation addressed the governance of federated and data-driven processes, where trust must extend beyond data to the workflows that depend on them. In this context, the Trustflow framework was developed to ensure end-to-end traceability and accountability in federated learning and analytical processes within UDTs. While existing blockchain-based approaches primarily record transactions or model updates, Trustflow integrates DIDs, VCs, and smart contracts to create a verifiable link between datasets, models, and participants. Every contribution, whether data, model update, or aggregation, is associated with a unique verifiable identity, allowing the complete provenance of results to be reconstructed. The framework also introduces quantitative influence estimation to measure the impact of individual contributions and an automated revocation mechanism that removes corrupted or malicious updates through policy-based on-chain enforcement. This ensures that federated models remain auditable and self-correcting, even in adversarial scenarios. Trustflow thus represents the governance and accountability layer of the proposed vision, extending verifiability from static data to dynamic processes and providing the missing link between decentralized learning, transparency, and trustworthiness in UDTs.

Collectively, these contributions form a coherent continuum that spans from interoperability and semantics to trust and governance.

The MIMs ensure the syntactic and operational connection among distributed systems; the extended GlassBox model and TDCs provide semantic and temporal coherence; VESPACE ensures verifiable and sovereign data exchange and Trustflow enforces process-level transparency and accountability. Together, they can be seen as building blocks that contribute to the construction of secure, interoperable, and trustworthy UDT ecosystems.

Looking ahead, several directions emerge for future work across the proposed models and frameworks. For the GlassBox model, the next step will be to conduct a wide set of experiments to assess the effectiveness of this original model in real-world contexts. Validating the results obtained from simulations and case studies coming from cities, as well as the application of the model to datasets regarding the past, will provide valuable data to further refine it. Regarding the TDC framework, future activities will explore the integration of additional data sources and the application of the framework to new domains, including smart city services and linked environmental datasets. Further developments will focus on enhancing interoperability through

the adoption of established standards, such as GeoSPARQL and Linked Open Data vocabularies, to ensure full compatibility with existing semantic web infrastructures. An additional objective is to incorporate spatial and temporal uncertainty modeling into the reasoning layer, enabling confidence-weighted reconstructions and enhancing robustness in the presence of ambiguous or incomplete records.

The next development step for VESPACE will integrate semantic alignment mechanisms with existing European Data Space frameworks, such as IDS and Gaia-X, enabling cross-domain data federation and interoperability. Ongoing research will also explore the inclusion of privacy-preserving computation techniques, such as zero-knowledge proofs and secure multiparty computation, to strengthen data confidentiality while preserving verifiability. For TrustFlow, the focus will be on further testing the framework in different settings, such as with various datasets, to assess its adaptability and effectiveness in diverse federated learning scenarios. Moreover, it will be extended to incorporate recent unlearning techniques [140], enabling the efficient removal of revoked data influence from the global model without compromising the learning process.

Beyond these individual advancements, the convergence of the proposed components into a single pipeline emerges as a key direction for future research, where interoperability, semantic modeling, data federation, and trust management operate as integrated layers of a unified framework. The integration of VESPACE and TrustFlow, in particular, represents a crucial next step, enabling a continuous chain of verifiability from data origin to model outcome and ensuring that every stage of the urban data lifecycle is transparent, accountable, and auditable. Future work will further evolve these mechanisms to enable semantic compliance verification and dynamic trust policy evaluation across federated data spaces. Additionally, the adoption of adaptive and probabilistic trust models within TrustFlow could introduce dynamic trust scoring, continuously refined by runtime metrics such as reliability, latency, or feedback consistency. Another research direction concerns the federation of UDTs across cities, enabling interoperability among local digital twins to form regional or transnational federations aligned with the principles of Gaia-X and the European Data Spaces initiative. Beyond the urban domain, the proposed models could be adapted to other critical sectors, such as healthcare, industry, and environmental monitoring, where verifiable, decentralized, and ethical data management is essential. Finally, integrating explainable AI and participatory governance mechanisms will be key to bridging human and machine trust, ensuring that the increasing automation of digital infrastructures remains transparent and socially accountable.

In conclusion, this dissertation has proposed an integrated approach for building the trustworthy foundations of UDTs. Through the combination of interoperability mechanisms, semantic modeling, temporal reasoning, verifiable data federation, and process governance, it has demonstrated how cities can move toward digital infrastructures that are not only intelligent and efficient, but also sovereign, verifiable, and ethically grounded. These frameworks lay the groundwork for UDTs that can evolve into self-regulating ecosystems capable of supporting democratic, explainable, and sustainable decision-making in the data-driven cities of the future.

Bibliography

[1] SAREF - Smart Appliances REFerence ontology. [Online]. Available: https://saref.etsi.org/.

[2] Verifiable Credentials Data Model v1.1. W3C Recommendation. [Online]. Available: https://www.w3.org/TR/vc-data-model/.

[3] *MEDES: Proceedings of the 8th International Conference on Management of Digital EcoSystems*, New York, NY, USA, 2016. Association for Computing Machinery.

[4] Decentralized identifiers (dids) v1.0: Core architecture, data model, and representations, July 2022. Editors and authors refer to the W3C editors list.

[5] A. Sabbioni *et al.* DIFFUSE: A DIstributed and decentralized platForm enabling Function composition in Serverless Environments. *Computer Networks*, 210, 2022.

[6] Abdelrahman Abdallah, Youssef Aboulyousr, Milad Ghantous, and Hassan Soubra. Digital twin of cairo: Opportunities and challenges. pages 413–417, 10 2023.

[7] Ahmed Abid, Jieun Lee, Franck Le Gall, and JaeSeung Song. Toward mapping an ngsi-ld context model on rdf graph approaches: A comparison study. *Sensors*, 22(13), 2022.

[8] Ajuntament de Barcelona. Barcelona cityos: Urban platform for smart city data. https://ajuntament.barcelona.cat/digital/en/technology-accessible-everyone/accessible-and-participatory/accessible-and-participatory-0, 2017.

[9] Ajuntament de Barcelona. Cityos documentation. https://cityos.ae/, 2020. Urban platform integrating IoT, open data and analytics.

[10] Adeyinka K. Akanbi and Muthoni Masinde. Semantic Interoperability Middleware Architecture for Heterogeneous Environmental Data Sources. In *Proc. of IST-Africa Week Conference (IST-Africa)*, pages 1–10, 2018.

[11] Alberto Aleta, Sandro Meloni, and Yamir Moreno. A multilayer perspective for the analysis of urban transportation systems. *Scientific Reports*, 7(1):44359, 2017.

[12] Moayad Aloqaily, Ismaeel Al Ridhawi, and Salil Kanhere. Reinforcing industry 4.0 with digital twins and blockchain-assisted federated learning. *IEEE Journal on Selected Areas in Communications*, 41(11):3504–3516, 2023.

[13] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, Giancarlo Pellegrino, and Alessandro Sorniotti. From multiple credentials to browser-based single sign-on: Are we more secure? In Jan Camenisch, Simone Fischer-Hübner, Yuko Murayama, Armand Portmann, and Carlos Rieder, editors, *Future Challenges in Security and Privacy for Academia and Industry*, pages 68–79, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[14] R. A. Atkinson, P. Zaborowski, F. Noardo, and I. Simonis. Smart cities – systems of systems interoperability and ogc enablers. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W3-2022:19–26, 2022.

[15] Sahin Aydin and Mehmet Nafiz Aydin. Semantic and syntactic interoperability for agricultural open-data platforms in the context of iot using crop-specific trait ontologies. *Applied Sciences*, 10(13), 2020.

[16] Abdullah Aziz, Olov Schelén, and Ulf Bodin. Data Integration Models for Heterogeneous Industrial Systems: A Conceptual Analysis. In *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2021.

[17] Chiara Bachechi. Digital twins for urban mobility. In *Symposium on Advances in Databases and Information Systems*, 2022.

[18] Hawazin Faiz Badawi, Fedwa Laamarti, and Abdulmotaleb El Saddik. Devising Digital Twins DNA Paradigm for Modeling ISO-Based City Services. *Sensors*, 21(4):1047, Feb 2021.

[19] Shaimaa Bajoudah, Changyu Dong, and Paolo Missier. Toward a Decentralized, Trust-Less Marketplace for Brokered IoT Data Trading Using Blockchain. In *Proc. of IEEE International Conference on Blockchain*, pages 339–346, 2019.

[20] Michael Batty. Digital twins. *Environment and Planning B: Urban Analytics and City Science*, 45(5):817–820, 2018.

[21] Martin Bauer. Fiware: Standard-based open source components for cross-domain iot platforms. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–6, 2022.

[22] Paolo Bellavista, Luca Foschini, and Alessio Mora. Decentralised Learning in Federated Deployment Environments: A System-Level Survey. *ACM Comput. Surv.*, 54(1):1–38, feb 2021.

[23] Julen Bernabé-Rodríguez, Albert Garreta, and Oscar Lage. A Decentralized Private Data Marketplace using Blockchain and Secure Multi-Party Computation. *ACM Trans. Priv. Secur.*, mar 2024.

[24] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International conference on machine learning*, pages 634–643. PMLR, 2019.

[25] Bold Cities / City of Rotterdam. Rotterdam urban digital twin. https://smart-cities-marketplace.ec.europa.eu/news-and-events/news/2019/rotterdams-digital-twin-redefines-our-physical-digital-social-worlds, 2020.

[26] Arianna Brutti, Angelo Frascella, Nicola Gessa, Piero de sabbata, and Cristiano Novelli. Interoperability in the smart city: A semantic approach for merging flexibility with strictness. pages 434–439, 06 2018.

[27] I. Buyuksalih, S. Bayburt, G. Buyuksalih, A. P. Baskaraca, H. Karim, and A. A. Rahman. 3d modelling and visualization based on the unity game engine – advantages and challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W4:161–166, 2017.

[28] Harry Cai, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2cp: Decentralized protocols to transparently evaluate contributivity in blockchain federated learning environments. *ArXiv*, abs/2011.07516, 2020.

[29] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6885–6893, 2021.

[30] Fran Casino, Thomas K. Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36:55–81, 2019.

[31] Giordana Castelli, Gabriella Tognola, Emilio Fortunato Campana, Amedeo Cesta, Matteo Diez, Marco Padula, Paolo Ravazzani, Giovanni Rinaldi, Stefano Savazzi, Michela Spagnuolo, and Lucanos M. Strambini. Urban intelligence: a modular, fully integrated, and evolving model for cities digital twinning. *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)*, pages 033–037, 2019.

[32] Cesium. 3d tiles: Open specification for streaming massive heterogeneous 3d geospatial datasets. https://cesium.com/why-cesium/3d-tiles/. Accessed: June 2025.

[33] K. Chaturvedi and T. H. Kolbe. A requirement analysis on extending semantic 3d city models for supporting time-dependent properties. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W9:19–26, 2019.

[34] F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD'16)*, pages 1011–1025, San Francisco, CA, USA, June 2016.

[35] City of Helsinki. Helsinki 3d+ project: City information model. https://www.hel.fi/en/decision-making/information-on-helsinki/maps-and-geospatial-data/helsinki-3d, 2016.

[36] CKAN. Ckan - open source data portal, 2025. Accessed: 2025-02-21.

[37] Daniel Commey, Sena Hounsinou, and Garth V. Crosby. Securing Health Data on the Blockchain: A Differential Privacy and Federated Learning Framework. *arXiv preprint arXiv:2405.11580*, 2024.

[38] Comune di Bologna. Open Data - Comune di Bologna, 2025. Accessed: 2025-02-16.

[39] Costagliola A.R. *et al.* Verifiable Blockchain-based Data Space Solution to Empower the Data Economy: Source Code. URL https://github.com/AndreaCostagliola/VSPACE. Accessed on February 2025.

[40] I. F. Cruz, V. R. Ganesh, C. Caletti, and P. Reddy. Giva: A semantic framework for geospatial and temporal data integration, visualization, and analytics. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*, pages 544–547, Orlando, FL, USA, 2013.

[41] Li Deren, Yu Wenbo, and Shao Zhenfeng. Smart city based on digital twins. *Computational Urban Science*, 1(1):4, 2021.

[42] Digital Bazaar. Digital Bazaar GitHub Repository, 2025. Accessed: 2025-02-16.

[43] Akanksha Dixit, Arjun Singh, Yogachandran Rahulamathavan, and Muttukrishnan Rajarajan. FAST DATA: A Fair, Secure, and Trusted Decentralized IIoT Data Marketplace Enabled by Blockchain. *IEEE Internet of Things Journal*, 10(4):2934–2944, 2023.

[44] Daniel Ricardo dos Santos. Access control vulnerabilities in network protocol implementations: How attackers exploit them and what to do about it. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies*, SACMAT '23, page 5–6, New York, NY, USA, 2023. Association for Computing Machinery.

[45] E. Jonas *et al.* Cloud Programming Simplified: A Berkeley View on Serverless Computing. Technical Report UCB/EECS-2019-3, University of California, Berkeley, 01 2019.

[46] E. Zeydan *et al.* Blockchain-Based Self-Sovereign Identity: Taking Control of Identity in Federated Learning. *IEEE Open Journal of the Communications Society*, 2024.

[47] Mohammed El-Hajj and Pim Beune. Decentralized zone-based pki: A lightweight security framework for iot ecosystems. *Information*, 15(6), 2024.

[48] EOSC. EOSC Future Results, 2024. Online; accessed June 2024.

[49] Esri / ArcGIS StoryMaps. Digital twin of downtown dubai. ArcGIS StoryMap, 2025. Accessed: 2025-09-06.

[50] EUR-Lex. Eur-lex. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401183. Accessed on July 2024.

[51] European Union. Data Quality and Artificial Intelligence - Mitigating Bias and Error to Protect Fundamental Rights. https://fra.europa.eu/en/publication/2019/data-quality-and-artificial-intelligence-mitigating-bias-and-error-protect, 2019. Last Accessed: December 2024.

[52] F. Ullah *et al.* Semantic Interoperability for Big-Data in Heterogeneous IoT Infrastructure for Healthcare. *Sustainable Cities and Society*, 34:90–96, 2017.

[53] Jaume Ferré-Bigorra, Miquel Casals, and Marta Gangolells. The adoption of urban digital twins. *Cities*, 131:103905, 2022.

[54] FIWARE. Orion Context Broker.

[55] FIWARE Foundation. Fiware documentation: Generic enablers. https://www.fiware.org/, 2021. Includes Context Broker, IoT integration tools, visualization and monetization components.

[56] Hyperledger Foundation. Hyperledger Foundation Project ARIES, Apr 2023.

[57] Jiahui Geng, Neel Kanwal, Martin Gilje Jaatun, and Chunming Rong. DID-EFed: Facilitating Federated Learning as a Service with Decentralized Identities. In *Proc. of Evaluation and Assessment in Software Engineering*, EASE 2021, page 329–335, New York, NY, USA, 2021. Association for Computing Machinery.

[58] Jorge Gil, Dessislava Petrova-Antonova, and Graham JL Kemp. Redefining urban digital twins for the federated data spaces ecosystem: A perspective. *Environment and Planning B: Urban Analytics and City Science*, 2024.

[59] Waris Gill, Ali Anwar, and Muhammad Ali Gulzar. TraceFL: Interpretability-Driven Debugging in Federated Learning via Neuron Provenance. *arXiv preprint arXiv:2312.13632*, 2024.

[60] Eunsu Goh, Daeyeol Kim, Do-Yup Kim, and Kwangkee Lee. Blockchain-enabled Federated Learning: A Reference Architecture Incorporating a DiD Access System. *arXiv preprint arXiv:2306.10841*, 2023.

[61] Gerhard Gröger and Lutz Plümer. Citygml – interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012.

[62] GSMA, May 2023.

[63] M. La Guardia, M. Koeva, L. Díaz-Vilariño, and P. Nourian. Open-source solutions for real-time 3d geospatial web integration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4-2024:289–295, 2024.

[64] Rudi Hardi, Achmad Nurmandi, Titin Purwaningsih, and Halimah Binti Abdul Manaf. Exploration organizational interoperability in smart governance in indonesia and malaysia. In Constantine Stephanidis, Margherita Antona, Stavroula Ntoa, and Gavriel Salvendy, editors, *HCI International 2023 Posters*, pages 203–210, Cham, 2023. Springer Nature Switzerland.

[65] Austin Harris and Mina Sartipi. Data integration platform for smart and connected cities. *Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering*, 2019.

[66] George Hatzivasilis, Ioannis G. Askoxylakis, George Alexandris, Darko Anicic, Arne Bröring, Vivek Kulkarni, Konstantinos Fysarakis, and George Spanoudakis. The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0. *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–7, 2018.

[67] Manoj Herath, Maira Alvi, Roberto Minerva, Hrishikesh Dutta, Noël Crespi, and Syed Mohsan Raza. Smart city digital twins: A modular and adaptive architecture for real-time data-driven urban management. *2024 20th International Conference on Network and Service Management (CNSM)*, pages 1–7, 2024.

[68] Manoj Herath, Maira Alvi, Roberto Minerva, Hrishikesh Dutta, Noel Crespi, and Syed Mohsan Raza. Smart city digital twins: A modular and adaptive architecture for real-time data-driven urban management. In *2024 20th International Conference on Network and Service Management (CNSM)*, pages 1–7, 2024.

[69] Manoj Herath, Hrishikesh Dutta, Roberto Minerva, Noel Crespi, Maira Alvi, and Syed Mohsan Raza. An integrated digital twin architecture for real-time urban air quality management. In *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 135–138, 2025.

[70] Patrik Hummel, Matthias Braun, Max Tretter, and Peter Dabrock. Data sovereignty: A review. *Big Data & Society*, 8(1):2053951720982012, 2021.

[71] IBM. Ibm manta data lineage. https://www.ibm.com/products/manta-data-lineage. Accessed June 2025.

[72] IDS. International Data Spaces, 2024. Online; accessed June 2024.

[73] IPFS Community. IPFS Cluster, 2024. Online; accessed June 2024.

[74] Sonain Jamil, MuhibUr Rahman, and Fawad. A comprehensive survey of digital twins and federated learning for industrial internet of things (iiot), internet of vehicles (iov) and internet of drones (iod). *Applied System Innovation*, 5(3):56, 2022.

[75] Blake Janes, Heather Crawford, and Tj Oconnor. Never ending story: Authentication and access control design flaws in shared iot devices. pages 104–109, 05 2020.

[76] Li Jiang, Hao Zheng, Hui Tian, Shengli Xie, and Yan Zhang. Cooperative federated learning and model update verification in blockchain-empowered digital twin edge networks. *IEEE Internet of Things Journal*, 9(13):11154–11167, 2021.

[77] Yifeng Jiang, Weiwen Zhang, and Yanxi Chen. Data Quality Detection Mechanism Against Label Flipping Attacks in Federated Learning. *IEEE Transactions on Information Forensics and Security*, 18:1625–1637, 2023.

[78] P. Kairouz et al. Advances and Open Problems in Federated Learning. *Foundations and trends in machine learning*, 14(1–2):1–210, 2021.

[79] Erkan Karabulut, Salvatore F. Pileggi, Paul Groth, and Victoria Degeler. Ontologies in digital twins: A systematic literature review. *Future Generation Computer Systems*, 153:442–456, April 2024.

[80] Maninder Jeet Kaur, Ved P. Mishra, and Piyush Maheshwari. *The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action*, pages 3–17. Springer International Publishing, Cham, 2020.

[81] Latif U. Khan, Ehzaz Mustafa, Junaid Shuja, Faisal Rehman, Kashif Bilal, Zhu Han, and Choong Seon Hong. Federated Learning for Digital Twin-Based Vehicular Networks: Architecture and Challenges. *IEEE Wireless Communications*, 31(2):156–162, 2024.

[82] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proc. of International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[83] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009. CIFAR-10 dataset, available at https://www.cs.toronto.edu/~kriz/cifar.html.

[84] John Krogstie. Modeling of digital ecosystems: Challenges and opportunities. In Luis M. Camarinha-Matos, Lai Xu, and Hamideh Afsarmanesh, editors, *Collaborative Networks in the Internet of Services*, pages 137–145, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[85] M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pages 935–942, Athens, GA, USA, November 2006.

[86] Sunil Kumar, SeungMyeong Jeong, Il Yeup Ahn, and Muhammad Aslam Jarwar. Things data interoperability through annotating onem2m resources for ngsi-ld entities. In *2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 119–124, 2022.

[87] L. Raes *et al.* DUET: A Framework for Building Interoperable and Trusted Digital Twins of Smart Cities. *IEEE Internet Computing*, 26(3):43–50, 2022.

[88] Lucía Latorre, Eduardo Rego, Lorenzo De Leo, and Mariana Gutierrez. Tech report: Digital twins. 2024.

[89] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020.

[90] Tian Li, Huaqun Wang, Debiao He, and Jia Yu. Synchronized provable data possession based on blockchain for digital twin. *IEEE Transactions on Information Forensics and Security*, 17:472–485, 2022.

[91] Xiaoyong Li, Feng Zhou, and Xudong Yang. A multi-dimensional trust evaluation model for large-scale p2p computing. *Journal of Parallel and Distributed Computing*, 71(6):837–847, 2011. Special Issue on Cloud Computing.

[92] Yuejin Li, Shengpeng Chen, Kai Hwang, Xiaoqiang Ji, Zhen Lei, Yi Zhu, Feng Ye, and Mengjun Liu. Spatio-temporal data fusion techniques for modeling digital twin city. *Geo-spatial Information Science*, 28(2):541–564, 2025.

[93] Linux Foundation. Edgex foundry, 2017.

[94] Y. Liu, J. Ding, Y. Fu, and Y. Li. UrbanKG: An urban knowledge graph system. *ACM Transactions on Intelligent Systems and Technology*, 14(4):1–25, August 2023.

[95] Qiuchen Lu, Ajith Kumar Parlikad, Philip Woodall, Gishan Don Ranasinghe, Xiang Xie, Zhenglin Liang, Eirini Konstantinou, James Heaton, and Jennifer Schooling. Developing a digital twin at building and city levels: Case study of west cambridge campus. *Journal of Management in Engineering*, 36(3):05020004, 2020.

[96] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet of Things Journal*, 8(4):2276–2288, 2020.

[97] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Communication-efficient federated learning for digital twin edge networks in industrial iot. *IEEE Transactions on Industrial Informatics*, 17(8):5709–5718, 2020.

[98] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Low-latency federated learning and blockchain for edge association in digital

twin empowered 6g networks. *IEEE Transactions on Industrial Informatics*, 17(7):5098–5107, 2020.

[99] Xujiang Luo and Bin Tang. Byzantine fault-tolerant federated learning based on trustworthy data and historical information. *Electronics*, 13(8):1540, 2024.

[100] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[101] Van Sy Mai, Richard J La, and Tao Zhang. A study of enhancing federated learning on non-iid data with server learning. *IEEE transactions on artificial intelligence*, 2024.

[102] Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security and Privacy*, 6(2):16–23, March 2008.

[103] Nicola Giuseppe Marchioro, Yannis Velegrakis, Valentine Anantharaj, Ian Foster, and Sandro Fiore. Trustworthy provenance for big data science: a modular architecture leveraging blockchain in federated settings. *ArXiv*, abs/2505.24675, 2025.

[104] Angelo Martella, Amro Issam Hamed Attia Ramadan, Cristian Martella, Mauro Patano, and Antonella Longo. State of the art of urban digital twin platforms. In Lucio Tommaso De Paolis, Pasquale Arpaia, and Marco Sacco, editors, *Extended Reality*, pages 299–317, Cham, 2023. Springer Nature Switzerland.

[105] Maxis. Inside the Glassbox (GDC 2012), 2012. Accessed: 2025-01-24.

[106] Carlo Mazzocca, Abbas Acar, Selcuk Uluagac, Rebecca Montanari, Paolo Bellavista, and Mauro Conti. A Survey on Decentralized Identifiers and Verifiable Credentials. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2025.

[107] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient Learning of Deep Networks from Decentralized Data. In *Proc. of Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[108] Stefan Mihai, Mahnoor Yaqoob, Dang V. Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V. Prasad, Hrishikesh Venkataraman, Ramona Trestian, and Huan X. Nguyen. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022.

[109] Fabio Miranda, Harish Doraiswamy, Marcos Lage, Kai Zhao, Bruno Goncalves, Luc Wilson, Mondrian Hsieh, and Claudio T. Silva. Urban pulse: Capturing the rhythm of cities. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):791–800, January 2017.

[110] Paolo Missier, Shaimaa Bajoudah, Angelo Capossele, Andrea Gaglione, and Michele Nati. Mind my value: a decentralized infrastructure for fair and trusted IoT data trading. In *Proc. of the Seventh International Conference on the Internet of Things*, IoT '17, New York, NY, USA, 2017. Association for Computing Machinery.

[111] M. Montanari, L. Marsicano, R. Trojanis, S. Bernardoni, and L. Gigli. Open history map – status of the project. *Archeomatica*, 13(2), December 2022.

[112] Vaikkunth Mugunthan, Ravi Rahman, and Lalana Kagal. Blockflow: An accountable and privacy-preserving solution for federated learning. *ArXiv*, abs/2007.03856, 2020.

[113] Tridib Mukherjee, Deepthi Chander, Anirban Mondal, Koustuv Dasgupta, Amit Kumar, and Ashwin Venkat. Cityzen: A cost-effective city management system with incentive-driven resident engagement. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 1, pages 289–296, 2014.

[114] P. Rajitha Nair and D. Ramya Dorai. Evaluation of performance and security of proof of work and proof of stake using blockchain. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 279–283, 2021.

[115] National Research Foundation Singapore. Virtual singapore project. https://www.sla.gov.sg/, 2014.

[116] Harris Niavis, Nikolaos Papadis, Venu Reddy, Hanumantha Rao, and Leandros Tassiulas. A Blockchain-based Decentralized Data Sharing Infrastructure for

Off-grid Networking. In *Proc. of IEEE International Conference on Blockchain and Cryptocurrency*, pages 1–5, 2020.

[117] Samuel D Okegbile, Jun Cai, Hao Zheng, Jiayuan Chen, and Changyan Yi. Differentially private federated multi-task learning framework for enhancing human-to-virtual connectivity in human digital twin. *IEEE Journal on Selected Areas in Communications*, 41(11):3533–3547, 2023.

[118] Samuel D Okegbile, Haoran Gao, Oluwasegun Talabi, Jun Cai, Dusit Niyato, and Xuemin Shen. Optimizing federated semantic learning in distributed aigc-enabled human digital twins: A multi-criteria and multi-shard user selection framework. *IEEE Transactions on Mobile Computing*, 2025.

[119] OneM2M. The Global Community that Develops Standards for IoT.

[120] Open & Agile Smart Cities. Minimal Interoperability Mechanisms. [Online]. Available: https://oascities.org/minimal-interoperability-mechanisms/.

[121] Open Geospatial Consortium. Geosparql – a geographic query language for rdf data. https://www.ogc.org/standards/geosparql, 2022.

[122] OpenAIRE. Openaire guidelines. https://guidelines.openaire.eu/en/latest. Accessed on July 2024.

[123] OpenMetadata. The open standard for metadata and data governance. https://open-metadata.org/. Accessed June 2025.

[124] Alessandro Ossola, Mary Cadenasso, and Emily Meineke. Valuing the role of time in urban ecology. *Frontiers in Ecology and Evolution*, 9, 03 2021.

[125] Pavlos Papadopoulos, Will Abramson, Adam J Hall, Nikolaos Pitropakis, and William J Buchanan. Privacy and trust redefined in federated machine learning. *Machine Learning and Knowledge Extraction*, 3(2):333–356, 2021.

[126] Dechen Peldon, Saeed Banihashemi, Khuong LeNguyen, and Sybil Derrible. Navigating urban complexity: The transformative role of digital twins in smart city development. *Sustainable Cities and Society*, 111:105583, 2024.

[127] Pedro Henrique Morgan Pereira, Gustavo Cainelli, Carlos Eduardo Pereira, Joao Paulo J. Da Costa, and Edison Pignaton De Freitas. An interoperability

middleware for iiot. In *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, pages 1–6, 2023.

[128] Antonios Pliatsios, Konstantinos Kotis, and Christos Goumopoulos. A systematic review on semantic interoperability in the ioe-enabled smart cities. *Internet of Things*, 22:100754, 2023.

[129] Ilung Pranata, Geoff Skinner, and Rukshan Athauda. Tide: measuring and evaluating trustworthiness and credibility of enterprises in digital ecosystem. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '11, page 9–16, New York, NY, USA, 2011. Association for Computing Machinery.

[130] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

[131] Deepak Puthal, Nisha Malik, Saraju P. Mohanty, Elias Kougianos, and Gautam Das. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consumer Electronics Magazine*, 7:6–14, 2018.

[132] Youyang Qu, Longxiang Gao, Yong Xiang, Shigen Shen, and Shui Yu. Fedtwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks. *IEEE Network*, 36(6):183–190, 2022.

[133] Hou Yee Quek, Franziska Sielker, Markus Kraft, Jethro Akroyd, Amit Bhave, Aurel Von Richthofen, Pieter Herthogs, Claudia Yamu, Li Wan, Timea Nochta, Gemma Burgess, Mei Qi Lim, Sebastian Mosbach, and V. S. K. Murthy Balijepalli. The conundrum in smart city governance: Interoperability and compatibility in an ever-growing digital ecosystem. 2021.

[134] Hafizur Rahman and Md. Iftekhar Hussain. A comprehensive survey on semantic interoperability for internet of things: State-of-the-art and research challenges. *Transactions on Emerging Telecommunications Technologies*, 31(12):e3902, 2020.

[135] RAIL. Rail. URL https://developer.mozilla.org/en-US/docs/Glossary/RAIL. Accessed on July 2024.

[136] Gowri Sankar Ramachandran, Rahul Radhakrishnan, and Bhaskar Krishnamachari. Towards a Decentralized Data Marketplace for Smart Cities. In *Proc. of IEEE International Smart Cities Conference*, pages 1–8, 2018.

[137] Swarna Priya Ramu, Parimala Boopalan, Quoc-Viet Pham, Praveen Kumar Reddy Maddikunta, Thien Huynh-The, Mamoun Alazab, Thanh Thi Nguyen, and Thippa Reddy Gadekallu. Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustainable Cities and Society*, 79:103663, 2022.

[138] M. Mazhar Rathore, Syed Attique Shah, Dhirendra Shukla, Elmahdi Bentafat, and Spiridon Bakiras. The Role of AI, Machine Learning, and Big Data in Digital Twinning: A Systematic Literature Review, Challenges, and Opportunities. *IEEE Access*, 9:32030–32052, 2021.

[139] Jane Reichel. Allocation of regulatory responsibilities: Who will balance individual rights, the public interest and biobank research under the gdpr? *GDPR and Biobanking*, 2021.

[140] Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2024.

[141] Mikail Mohammed Salim, David Camacho, and Jong Hyuk Park. Digital twin and federated learning enabled cyberthreat detection system for iot networks. *Future Generation Computer Systems*, 161:701–713, 2024.

[142] Hugo Lloreda Sanchez, Sophie Tysebaert, Annanda Thavymony Rath, and Etienne Riviere. Audittrust: Blockchain-based audit trail for sharing data in a distributed environment. In *EDCC Workshops*, 2022.

[143] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, June 2018.

[144] Gerhard Schrotter and Christian Hürzeler. The digital twin of the city of zurich for urban planning. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1):99–112, 2020.

[145] Self-Sovereign Identity. Home – self-sovereign identity. https://www.selfsovereignidentity.it/home-eng/, 2022. Accessed: 2025-08-24.

[146] Ehab Shahat, Chang T. Hyun, and Chunho Yeom. City digital twin potentials: A review and research agenda. *Sustainability*, 13(6), 2021.

[147] Junxin Shen, Shuilan Zhou, and Fanghao Xiao. Research on data quality governance for federated cooperation scenarios. *Electronics*, 13(18), 2024.

[148] Chi-Sheng Daniel Shih, Chan-Ming Yang, and Yen-Chien Cheng. Data alignment for multiple temporal data streams without synchronized clocks on iot fusion gateway. *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 667–674, 2015.

[149] Gagandeep Shubham, Vidushi Agarwal, and Sujata Pal. IoT Data Security: An Integration of Blockchain and Federated Learning. In *Proc. of IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 434–439, 2023.

[150] Dr-Shadab Siddiqui. X. 509 and pgp public key infrastructure methods: A critical review. 01 2014.

[151] Stefano Silvestri, Giuseppe Tricomi, Salvatore Rosario Bassolillo, Riccardo De Benedictis, and Mario Ciampi. An urban intelligence architecture for heterogeneous data and application integration, deployment and orchestration. *Sensors*, 24(7), 2024.

[152] Durga Sivasubramanian, Lokesh Nagalapatti, Rishabh Iyer, and Ganesh Ramakrishnan. Gradient coreset for federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2648–2657, 2024.

[153] Jaap-Willem Sjoukema, Arnold K. Bregt, and Joep Crompvoets. Evolving spatial data infrastructures and the role of adaptive governance. *ISPRS International Journal of Geo-Information*, 6:254, 2017.

[154] Michael Sober, Giulia Scaffino, Stefan Schulte, and Salil S Kanhere. A blockchain-based IoT data marketplace. *Cluster computing*, 26(6):3523–3545, 2023.

[155] Wen Sun, Ning Xu, Lu Wang, Haibin Zhang, and Yan Zhang. Dynamic digital twin and federated learning with incentives for air-ground networks. *IEEE Transactions on Network Science and Engineering*, 9(1):321–333, 2020.

[156] SynchroniCity. SynchroniCity: Delivering an IoT enabled Digital Single Market for Europe and Beyond.

[157] Hamed Taherdoost. Smart contracts in blockchain technology: A critical review. *Information*, 14(2), 2023.

[158] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, 2019.

[159] Hubert Tardieu. Role of gaia-x in the european data space ecosystem. In *Designing Data Spaces*, 2022.

[160] The Frictionless Data Project. Frictionless data specifications. https://specs.frictionlessdata.io/, 2024.

[161] Reza Tourani, Travis Mick, Satyajayant Misra, and Gaurav Panwar. Security, privacy, and access control in information-centric networking: A survey. *IEEE Communications Surveys & Tutorials (in submission, available at arXiv)*, PP, 03 2016.

[162] [ui!] UrbanPulse. Urbanpulse documentation. https://resilientcitiesnetwork.org/urban-pulse/, 2021. Platform for real-time smart city monitoring.

[163] University of Cambridge. Cambridge centre for digital built britain: Digital twin programme. https://www.cdbb.cam.ac.uk/research/digital-twins, 2019.

[164] Eric VanDerHorn and Sankaran Mahadevan. Digital twin: Generalization, characterization and implementation. *Decision Support Systems*, 145:113524, 2021.

[165] Fabio Viola, Francesco Antoniazzi, Cristiano Aguzzi, Carlos Kamienski, and Luca Roffia. Mapping the ngsi-ld context model on top of a sparql event processing architecture: Implementation guidelines. In *2019 24th Conference of Open Innovations Association (FRUCT)*, pages 493–501, 2019.

[166] QL Vivi, AK Parlikad, P Woodall, GD Ranasinghe, and J Heaton. Developing a dynamic digital twin at a building level: Using cambridge campus as case study. 2019.

[167] W3 Consortium. Bitstring status list v1.0. [Online]. URL: https://www.w3.org/TR/vc-bitstring-status-list/, 12 2024. Last Accessed: December 2024.

[168] W3C. JSON LD 1.1 - A JSON-based Serialization for Linked Data, 2020.

[169] W3C. Verifiable Credentials Bitstring Status List, 2025. Accessed: 2025-02-16.

[170] Paul Waddell, Ignacio Garcia-Dorado, Samuel M. Maurer, Geoff Boeing, Max Gardner, Emily Porter, and Daniel G. Aliaga. Architecture for modular microsimulation of real estate markets and transportation. *ArXiv*, abs/1807.01148, 2018.

[171] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A Principled Approach to Data Valuation for Federated Learning. *Federated Learning: Privacy and Incentive*, pages 153–167, 2020.

[172] Yuntao Wang, Zhou Su, Shaolong Guo, Minghui Dai, Tom H. Luan, and Yiliang Liu. A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects. *IEEE Internet of Things Journal*, 10(17):14965–14987, September 2023.

[173] Yuntao Wang, Zhou Su, Ning Zhang, Jianfei Chen, Xin Sun, Zhiyuan Ye, and Zhenyu Zhou. Spds: A secure and auditable private data sharing scheme for smart grid based on blockchain and smart contract. *IEEE Transactions on Industrial Informatics*, PP:1–1, 11 2020.

[174] Wilkinson, M.D. *et al.* The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 03 2016.

[175] Eyal Winter. The Shapley Value. *Handbook of Game Theory with Economic Applications*, 3:2025–2054, 2002.

[176] Maximilian Wöhrer and Uwe Zdun. Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity. *Proc. of International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 2–8, 2018.

[177] Kaidong Wu, Yun Ma, Gang Huang, and Xuanzhe Liu. A first look at blockchain-based decentralized applications. *CoRR*, abs/1909.00939, 2019.

[178] Qichao Xu, Zhou Su, Kuan Zhang, and Peng Li. Intelligent cache pollution attacks detection for edge computing enabled mobile social networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(3):241–252, 2020.

[179] Y. Xue *et al.* Toward Understanding the Influence of Individual Clients in Federated Learning. In *Proc. of AAAI Conference on Artificial Intelligence*, 2020.

[180] Miao Yang, Hua Qian, Ximin Wang, Yong Zhou, and Hongbin Zhu. Client selection for federated learning with label noise. *IEEE Transactions on Vehicular Technology*, 71(2):2193–2197, 2021.

[181] XiaoHui Yang and TianChang Li. A blockchain-based federated learning framework for secure aggregation and fair incentives. *Connection Science*, 36(1):2316018, 2024.

[182] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, and Kari Smolander. Where is current research on blockchain technology?—a systematic review. *PLOS ONE*, 11, 10 2016.

[183] DaeGeun Yoon, SungJin Moon, KiSung Park, and SungKee Noh. Blockchain-based Personal Data Trading System using Decentralized Identifiers and Verifiable Credentials. In *Proc. of International Conference on Information and Communication Technology Convergence*, pages 150–154, 2021.

[184] X. Zhao, Y. Cao, J. Wang, X. Fan, and M. Chen. A hierarchical spatio-temporal object knowledge graph model for dynamic scene representation. *Transactions in GIS*, 27:1992–2016, October 2023.