2026 European Conference on Computing in Construction
Corfu, Greece
July 12-15, 2026

A COMPARATIVE ANALYSIS OF IoT DATA PIPELINES FOR REAL-TIME MIXED REALITY VISUALISATION IN UNITY-BASED DIGITAL TWINS

Muhammad Shahzad[1], Joseph H. M. Tah[1], Avar Almukhtar[1], and Muhammad Younas[1]
[1]Oxford Brookes University, Oxford, United Kingdom

Abstract

Digital Twins increasingly rely on real-time IoT data, yet the end-to-end architectural implications of alternative integration approaches remain underexplored. This paper presents a controlled comparative analysis of three IoT data pipelines connecting The Things Network (TTN) to a Unity-based Digital Twin for mixed reality visualisation: cloud-centric pipelines using Amazon Web Services and Microsoft Azure, and a lightweight publish–subscribe pipeline based on direct MQTT delivery. By isolating the data integration layer, a benchmarking approach evaluates data transmission behaviour, persistence, integration complexity, scalability, and cost. The findings provide practical architectural insight into how pipeline design choices shape Digital Twin behaviour.

Introduction

Digital Twins are increasingly adopted in the built environment to integrate physical assets, sensing systems, and digital representations in support of monitoring, operation, and maintenance activities. As originally articulated by Grieves and Vickers (2016) and subsequently extended to construction and infrastructure contexts by Boje et al. (2020), Digital Twins enable continuous coupling between physical systems and their digital counterparts. By linking real-world measurements with digital models, they provide situational awareness. They also support informed decision-making across the lifecycle of buildings and infrastructure (Boschert and Rosen, 2016; Lu et al., 2020; Shahzad et al., 2025).

A fundamental requirement for operational Digital Twins is the reliable and continuous integration of real-time data from distributed sensors. Internet of Things (IoT) and related sensor technologies have widely deployed in buildings to capture environmental and operational parameters relevant to performance assessment and maintenance (El-Ammari and Hammad, 2019; Al-Nasser et al., 2024). At the network layer, Low-Power Wide-Area Network (LPWAN) technologies such as LoRaWAN enable scalable sensor deployments with low energy consumption (Centenaro et al., 2016; LoRa Alliance, 2025). However, as emphasised by Wang et al. (2013) and Tao et al. (2018), sensor deployment alone does not constitute a functional Digital Twin. Real-time data must be transmitted, processed, and delivered through an integration pipeline that is sufficiently robust, scalable, and responsive for operational use.

Immersive Digital Twin applications are increasingly reported in construction and facility management research. However, existing studies largely foreground application-level outcomes such as training, inspection, or collaboration, while treating the underlying IoT data pipeline as an implementation detail (Palmarini et al., 2018). In practice, IoT-enabled Digital Twin systems are commonly realised through either cloud-centric architectures, which provide managed ingestion, storage, and dashboard services, or lightweight publish–subscribe approaches that prioritise low-latency data delivery. Cloud platforms such as Amazon Web Services (AWS) and Microsoft Azure are frequently adopted for Digital Twin deployments due to their scalability and managed services (Lu et al., 2020; Al-Nasser et al., 2024), while messaging protocols such as MQTT are widely used for real-time IoT streaming (Thangavel et al., 2014).

At the platform level, several studies have presented comparative analyses of major cloud providers, most notably AWS and Azure, focusing on pricing models, security mechanisms, quality-of-service attributes, and service capabilities (Kotas et al., 2018; Eisa et al., 2020; Palumbo et al., 2021; Hameed et al., 2025). While these contributions provide valuable insight into cloud platform characteristics, they are typically conducted from a general cloud computing or enterprise systems perspective. As noted by Palumbo et al. (2021) and Eisa et al. (2020), such comparisons often rely on abstract performance metrics or simulation-based evaluation and do not examine how alternative cloud architectures behave when embedded within end-to-end Digital Twin data pipelines. Moreover, controlled comparisons that trace IoT data flow from sensing infrastructure to immersive visualisation environments under identical conditions remain limited.

This paper addresses this gap through a controlled comparative analysis of three end-to-end IoT data pipeline architectures that deliver live sensor readings from The Things Network (TTN) to a Unity-based Digital Twin. TTN is an open, LoRaWAN-based IoT network providing device management and standardised data forwarding services (TTN, 2023). The evaluated architectures comprise: (i) cloud-centric pipelines implemented using AWS and Microsoft Azure, providing

managed ingestion, serverless processing, and persistent storage (AWS, 2024a, Microsoft, 2024); and (ii) a lightweight publish–subscribe integration path in which Unity subscribes directly to sensor topics via an MQTT broker, enabling low-latency data delivery.

The conceptual scope of the study is illustrated in Figure 1, which positions the data pipeline layer as an intermediary between the IoT sensing infrastructure and the Digital Twin visualisation environment. Within this framework, the case study environment, TTN configuration, and Unity-based Digital Twin are held constant, and only the data pipeline architecture between TTN and Unity is varied. Mixed Reality (MR) and Virtual Reality (VR) are employed as representative high-demand real-time visualisation contexts such as live walkthroughs, remote asset inspection, and collaborative review where low latency, continuous updates, and immediate user feedback are critical for effective decision-making rather than as the primary objects of evaluation.

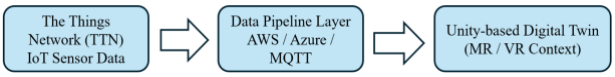

Figure 1: Conceptual Scope of the Study

The contribution of this work is threefold. First, it documents the implementation of three TTN-to-Unity data pipelines under identical case study conditions, enabling controlled architectural comparison. Second, it applies a structured benchmarking approach to evaluate IoT data pipelines across data transmission behaviour, platform capabilities, data persistence, Unity integration complexity, and architectural maintainability. Third, it synthesises the observed trade-offs to support evidence-informed selection of IoT data pipeline architectures for real-time Digital Twin visualisation in the built environment.

Case Study and Experimental Context

Case Study Overview

The study is based on an operational building used for teaching and research activities, instrumented with six IoT sensors for continuous environmental monitoring. The building provides a realistic yet controlled context for evaluating alternative IoT data pipeline architectures under normal operational conditions, without introducing domain-specific application bias.

Environmental monitoring was performed using Elsys CO_2 sensors deployed exclusively within the shared office space of the case study building. The sensors measure temperature, relative humidity, and CO_2 concentration. These are widely accepted indicators of indoor environmental quality in office environments and representative of sensing modalities commonly used in building monitoring and facilities management. The sensors operate on a low-power basis and are suitable for continuous, long-term deployment. Figure 2 illustrates the spatial distribution of the six deployed sensors within the shared office space, which provides the physical context for mapping measured conditions to corresponding locations in the Unity-based Digital Twin.

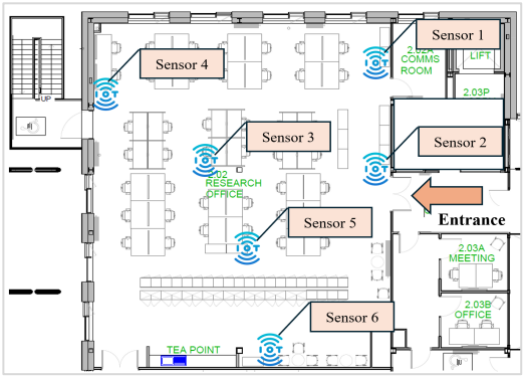

Figure 2 – Spatial deployment of IoT sensors within the case study building

All sensors communicated using a Low-Power Wide-Area Network (LPWAN) based on the LoRaWAN protocol. Sensor uplinks were received and managed by The Things Network (TTN), which served as the common IoT network layer across all experimental configurations. TTN was selected due to its widespread adoption, support for LoRaWAN devices, and ability to forward sensor data through standardised integration mechanisms. Importantly, TTN was already operational within the case study environment, ensuring a stable and continuously functioning network infrastructure and avoiding additional variability associated with network setup.

At the visualisation layer, sensor data were integrated into a Unity-based Digital Twin representing the monitored building. Sensor readings were mapped to their corresponding spatial locations within a three-dimensional model and updated dynamically in response to incoming data. MR and VR devices were used as representative high-demand visualisation contexts, where lag-free interaction and continuous state synchronisation are essential, while the visualisation logic and interaction mechanisms were kept constant throughout the study.

Controlled Comparison Strategy

To enable controlled architectural comparison, the physical sensing infrastructure, including sensor types, deployment locations, payload structure, and data transmission frequency, was held constant across all experiments. The sensor data transmission interval was configured at 10 minutes for all devices, ensuring consistent temporal resolution across all pipeline evaluations. The TTN configuration remained unchanged, ensuring identical data semantics and network behaviour for each pipeline.

The Unity-based Digital Twin, including the 3D model, data mapping logic, and update routines, was also held constant across all experimental scenarios. As shown in Figure 3, this setup fixes both the upstream sensing and networking layer and the downstream visualisation component. The data integration layer is isolated as the sole variable element.

All pipelines were implemented and executed in the same computing environment, using identical hardware and software configuration, to ensure that observed differences were not influenced by variations in computational resources or execution conditions.

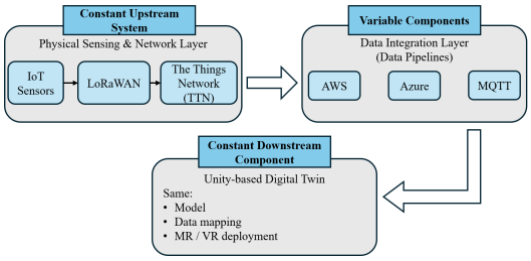

Figure 3 – Controlled Experimental Setup

Within this controlled configuration, three data pipeline architectures were implemented and evaluated: cloud-centric pipelines using AWS and Azure, and a lightweight publish–subscribe pipeline using an MQTT broker. This design ensures that observed differences in system behaviour, complexity, and capability can be attributed directly to the data pipeline architecture rather than to sensing, networking, or visualisation factors.

IoT Data Pipeline Architectures

This section describes the IoT data pipeline architectures implemented and evaluated in the study. All architectures connect the same upstream IoT network to the same Unity-based Digital Twin, enabling controlled comparison of alternative data integration approaches. A common baseline is provided by The Things Network (TTN), followed by three downstream pipelines that differ only in how sensor data are processed, stored, and delivered.

Common Baseline: TTN Data Flow

All sensor data originate from distributed IoT devices deployed within the case study building and are transmitted using the LoRaWAN protocol. Uplink messages are received and managed by The Things Network (TTN), which serves as the common IoT network layer across all experimental configurations.

Sensor uplinks are encoded as structured JSON payloads containing timestamped measurements and device identifiers. Payload structure and transmission frequency remain identical across all pipelines. TTN functions exclusively as a data forwarding layer, providing webhook- and MQTT-based interfaces to downstream systems without performing data processing or persistence. This baseline data flow configuration ensures that observed differences arise from downstream pipeline architectures rather than sensing or network variability.

Cloud-Centric Pipelines: TTN–AWS–Unity and TTN–Azure–Unity

The two TTN–AWS–Unity and TTN–Azure–Unity pipelines follow a cloud-centric architectural pattern designed to support persistence and structured data access. Sensor data are forwarded from TTN via webhooks to cloud ingestion endpoints, triggering serverless processing functions that perform lightweight payload parsing and validation.

Processed data are stored in managed cloud databases, enabling retention of both recent and historical sensor readings. An API layer exposes stored data to the Unity-based Digital Twin via HTTP-based requests. Unity retrieves updated sensor data at a fixed polling interval of 10 seconds, ensuring regular synchronisation between the cloud database and the visualisation environment. This pull-based access model decouples ingestion from consumption. However, it introduces multiple intermediary components between the IoT network and the visualisation layer.

Architecturally, the AWS- and Azure-based pipelines exhibit functional parity. Differences observed between them relate primarily to platform-specific tooling and configuration workflows rather than fundamental architectural design.

Streaming-Based Pipeline: TTN–MQTT Broker–Unity

The TTN–MQTT broker–Unity pipeline represents a lightweight, streaming-based architecture based on a publish–subscribe communication model. Sensor data are forwarded from TTN to an MQTT broker, from which the Unity application subscribes directly to relevant topics.

Data delivery is push-based and occurs as messages are published, enabling near real-time updates. Unlike the cloud-centric pipelines, this architecture introduces no intermediary processing or inherent data persistence. Sensor readings are therefore transient unless additional storage is explicitly introduced. This design prioritises immediacy and simplicity while shifting responsibility for data handling and temporal context to the consuming application.

Architectural Abstraction

Figure 4 presents an abstracted comparison of the two dominant pipeline paradigms evaluated: cloud-centric and streaming-based pipelines.

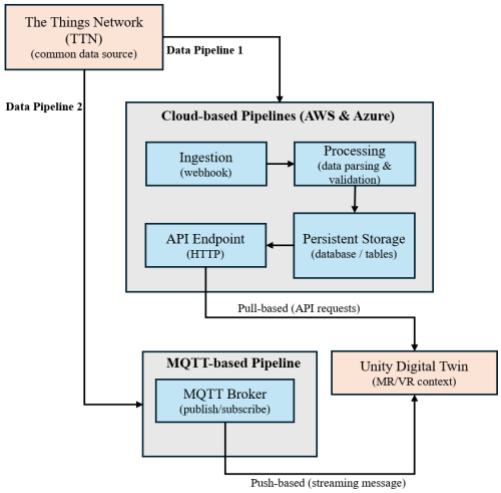

Figure 4 - Abstracted IoT data pipeline architectures connecting TTN to a Unity-based Digital Twin

The figure highlights differences in component composition, data flow, and data persistence while omitting platform-specific services. This abstraction provides architectural context for the benchmark evaluation presented in the following section.

Benchmarking Approach

The comparative evaluation in this paper adopts a structured benchmarking approach to assess alternative IoT data pipeline architectures supporting real-time Digital Twin visualisation. Consistent with established cloud and systems benchmarking practice, complex platforms are evaluated using common architectural and operational criteria rather than vendor-specific performance ranking (Kotas et al., 2018; Eisa et al., 2020). Accordingly, this study focuses on architectural behaviour and practical integration characteristics observed during implementation and deployment.

Benchmark Design Rationale

The benchmarking criteria were developed according to three principles: platform agnosticism, architectural relevance, and implementation grounding. First, the criteria are platform-agnostic, focusing on architectural and functional properties that can be compared across cloud-managed and publish–subscribe approaches rather than vendor-specific services. This aligns with established cloud benchmarking literature, where heterogeneous platforms are normalised into comparable capability and quality dimensions (Kotas et al., 2018; Eisa et al., 2020), and with more recent work emphasising architectural capability over provider-specific implementation detail (Hameed et al., 2025). Palumbo et al. (2021) further demonstrate the importance of vendor-neutral analysis when assessing end-to-end latency behaviour.

Second, the criteria are architecture-oriented, emphasising data flow, component responsibilities, and coupling between ingestion, persistence, and application access layers. This reflects evidence that system-level properties such as latency, responsiveness, and integration overhead emerge primarily from architectural composition rather than from individual services in isolation (Palumbo et al., 2021).

Third, the criteria are deployment-informed, grounded in deploying and troubleshooting multiple TTN-to-Unity pipelines in a real building context. While protocol-level studies quantify MQTT performance under controlled conditions (Thangavel et al., 2014; Ansyah et al., 2023), they do not capture the integration and operational challenges encountered in end-to-end Digital Twin deployments. Incorporating factors such as configuration effort, tooling complexity, and operational usability therefore extends existing benchmarking work toward application-facing Digital Twin systems.

Each pipeline was implemented and tested repeatedly under identical conditions. Benchmark assessments were based on direct observation of system behaviour during deployment, configuration, and runtime operation. Comparative evaluation was conducted using consistent criteria across all pipelines, ensuring that observed differences reflect architectural characteristics rather than experimental variability.

In this study, latency is defined as the elapsed time between sensor data transmission from TTN and its availability within the Unity-based Digital Twin. Based on repeated implementation observations, the MQTT based pipeline exhibited near real-time updates (typically within 1-2 seconds), whereas cloud-based pipelines introduced additional delay (approximately 5-8 seconds) due to ingestion, storage, and API retrieval stages.

Benchmark Categories

Based on these principles, including platform agnosticism, architectural relevance, and implementation grounding, six benchmark categories were defined.

Data Transmission Characteristics assess setup effort, communication mechanisms, and latency, including the distinction between push-based streaming and pull-based API access.

Platform Capabilities evaluate native support for data inspection, monitoring, and debugging.

Data Storage and Representation address persistence, ease of querying, and built-in visualisation or analytics.

Unity Integration Complexity captures software development kit (SDK) availability, programming effort, and runtime debugging effort.

Architectural Complexity and Maintainability consider component count, scalability, and suitability for sustained deployment.

Cost and Resource Implications capture licensing constraints, tooling requirements, and access to platform features relevant to academic and early-stage deployments.

Together, these benchmarks provide a concise basis for comparing cloud-based and streaming-based IoT data pipeline architectures under controlled conditions. Their application to the AWS-, Azure-, and MQTT broker–based pipelines is presented in the following section.

Comparative Analysis and Results

This section presents a comparative analysis of the three IoT data pipeline architectures evaluated in the study: TTN–AWS–Unity, TTN–Azure–Unity, and TTN–MQTT broker–Unity. The analysis applies the benchmarking approach defined in the preceding section and focuses on architectural behaviour observed during implementation and deployment. Results are presented using high-level comparison tables supported by restrained qualitative interpretation. The objective is not to rank platforms, but to identify architectural trade-offs relevant to real-time Digital Twin visualisation.

Comparison of Cloud-Based Pipelines: AWS vs Azure

Both the TTN–AWS–Unity and TTN–Azure–Unity pipelines follow a cloud-centric architectural pattern, employing webhook-based ingestion, serverless processing, managed storage, and API-mediated data

access. Both architectures support persistent data storage, structured querying, and controlled access from the Unity-based Digital Twin.

Despite the architectural parity, differences were observed in integration effort, workflow transparency, tooling requirements, and practical usability, particularly under student-tier account constraints. Table 1 summarises these differences across key architectural and implementation-oriented criteria.

Table 1. High-level comparison of AWS and Azure cloud-based IoT pipelines for digital twin data integration

Criterion	AWS	Azure
Overall architectural pattern	Cloud-centric, serverless	Cloud-centric, serverless
Data ingestion from TTN	Webhook-based	Webhook-based
Serverless processing	AWS Lambda	Azure Functions
Data persistence	Managed database (table-based)	Managed storage / tables
Querying capability (ease of data access and retrieval)	Strong, straightforward querying	Strong but more workflow-dependent
Data export (CSV/Excel)	Direct export from tables	Indirect; requires queries or external tools
Built-in dashboards (student tier)	Limited without paid services	Available for basic charts
Unity integration effort	Moderate	High
Workflow transparency	Relatively clear and linear	Fragmented, multi-step
Tooling requirements	Mostly web-based	Web + desktop tools (e.g. Storage Explorer)
Maintainability (ease of modifications)	High	Medium
Learning curve	Moderate	Steep
User Interface Usability	Relatively Intuitive	Complex

Table 1 compares the AWS- and Azure-based pipelines for digital twin data integration, focusing on how each cloud platform supports real-time sensor data delivery, persistence, and access within a Unity-based Digital Twin. The AWS-based pipeline exhibited a clearer and more linear workflow, enabling easier data access, modification, and maintenance during development. The Azure-based pipeline achieved comparable functional outcomes but required more extensive configuration, additional tooling, and multi-step workflows, resulting in higher development and debugging overhead. These differences reflect platform-level implementation characteristics rather than fundamental architectural distinctions.

Cloud-Based Pipelines versus MQTT broker-based Streaming

While AWS and Azure are architecturally comparable, the TTN–MQTT broker–Unity pipeline represents a fundamentally different integration paradigm. Rather than relying on cloud-managed processing and storage, this pipeline adopts a lightweight publish–subscribe model in which sensor data are streamed directly from TTN to the Unity application via an MQTT broker.

From a data transmission perspective, the streaming-based pipeline demonstrated lower end-to-end latency and simpler configuration, as it does not introduce serverless processing, databases, or API gateways. However, this architectural simplicity comes at the cost of data persistence, platform-level analytics, and historical access, which are inherent features of cloud-centric pipelines.

Table 2 – Comparison of cloud-based pipelines and MQTT broker-based streaming for Digital Twin data integration

Criterion	AWS/Azure	MQTT
Architectural paradigm	Centralised, cloud-centric	Decentralised, streaming
Data delivery model	Pull-based (API requests)	Push-based (publish–subscribe)
End-to-end latency	Medium (5-8 seconds)	Low (1-2 seconds)
Data persistence	Native and integral	Not provided
Historical data access	Supported	Not supported
Built-in analytics and dashboards	Available (tier-dependent)	Not available
Unity integration complexity	Medium to High	Low
Number of system components	High	Low
Setup and configuration effort	High	Low
Cost sensitivity (student tier)	High	Low
Suitability for live visualisation	Appropriate	Appropriate
Suitability for long-term monitoring	Appropriate	Limited
Data latency	Medium	Low

Table 2 summarises the key architectural differences as they affect real-time Digital Twin data integration, contrasting cloud-based pipelines with the MQTT broker–based streaming approach. The analysis indicates that streaming-based integration is fairly effective for immediate, real-time data delivery in live Digital Twin visualisation. In contrast, cloud-based pipelines provide stronger support for long-term monitoring, historical analysis, and system observability, but with higher integration overhead.

Architectural Trade-off Observations

The comparative analysis reveals several recurring architectural trade-offs across the evaluated pipelines.

Cloud richness versus simplicity – Cloud-centric pipelines provide a rich ecosystem of managed services, including ingestion, storage, monitoring, and analytics, supporting structured data management and operational visibility (AWS, 2024b). However, these capabilities require coordination of multiple components. In contrast, the MQTT broker–based pipeline minimises architectural footprint and setup effort but offers limited platform-level tooling.

Persistence versus immediacy – Persistent storage is a defining characteristic of cloud-based pipelines, enabling historical access and retrospective analysis (Palmarini et al., 2018; Qiu et al., 2025). By contrast, streaming-based pipelines prioritise immediacy by delivering data directly to the consuming application, supporting near real-time updates but offering no inherent data retention (Thangavel et al., 2014).

Integration overhead versus architectural responsibility – Cloud-based pipelines introduce higher integration overhead due to serverless processing, databases, and API layers, but centralise responsibilities such as data validation, persistence, and access control. Streaming-based pipelines reduce backend complexity but shift responsibility for data handling, temporal context, and robustness to the consuming application (Al-Nasser et al., 2024).

Summary of Findings

The comparative analysis demonstrates that no single IoT data pipeline architecture is universally optimal. Cloud-centric pipelines offer robustness, persistence, and extensibility, making them suitable for long-term, data-rich Digital Twins. Streaming-based integration prioritises low latency and simplicity, supporting responsive real-time visualisation with minimal backend overhead.

These findings reinforce that data pipeline selection should be guided by architectural alignment with application requirements, resource constraints, and system lifecycle considerations, rather than by platform capability alone. The implications of these architectural trade-offs for real-time Digital Twin visualisation are discussed in the following section.

Implications and Discussion

The comparative analysis demonstrates that architectural decisions at the IoT data pipeline level have direct and measurable implications for how effectively Digital Twins support real-time visualisation. While prior studies have examined cloud platform selection using quality-of-service metrics, simulation models, or abstract optimisation criteria (e.g. Kotas et al., 2018; Eisa et al., 2020; Hameed et al., 2025), such analyses are typically decoupled from operational Digital Twin deployments. By contrast, this study evaluates data pipeline behaviour under controlled but realistic conditions. It reveals how architectural composition shapes update continuity, data availability, and development and maintenance effort in real-time Digital Twin systems.

Latency, Update Continuity, and Immediacy

Latency is a critical determinant of real-time Digital Twin fidelity, as temporal misalignment between physical processes and their digital representations can undermine user confidence and operational reliability. As Wang et al. (2013) and Palmarini et al. (2018) have argued, delayed or inconsistent updates reduce the effectiveness of immersive and monitoring-oriented Digital Twins. In MR and VR environments, such latency can manifest as perceptible delays between user interaction and system response, disrupting the sense of immersion and reducing the reliability of real-time feedback. This is particularly critical in applications such as live walkthroughs and remote monitoring, where users rely on immediate visual updates to interpret environmental conditions and support timely decision-making.

The results indicate that publish–subscribe streaming architectures are architecturally conducive to lower end-to-end latency than cloud-centric pipelines, primarily due to the absence of intermediary storage, querying, and API retrieval stages. This design supports more continuous and immediate visual updates within Digital Twin. In contrast, cloud-based pipelines introduce additional processing layers associated with ingestion, persistence, and pull-based access, which can lead to discretised update behaviour. These observations suggest that low-latency streaming architectures are particularly well suited to live monitoring and event-driven Digital Twin applications, while recognising that such latency advantages arise from architectural simplification rather than from quantified performance measurements.

Data Persistence and Long-Term Use

Persistent data storage is a defining characteristic of cloud-centric IoT data pipelines and is widely recognised as essential for diagnostics, performance assessment, and maintenance planning in the built environment. Palmarini et al. (2018) and Qiu et al. (2025) emphasise that historical data enables Digital Twins to move beyond real-time state representation toward analytical and decision-support functions.

Consistent with this view, the analysis shows that cloud-based pipelines inherently support structured data retention and querying, enabling historical inspection and

aggregated representations within or alongside the Digital Twin. Streaming-based architectures, by contrast, prioritise transient data delivery and do not retain historical information unless supplemented by additional storage components. While this does not affect real-time update continuity, it constrains retrospective analysis, highlighting persistence and immediacy as complementary but not always compatible architectural objectives.

Integration Complexity and Architectural Responsibility

Integration complexity directly affects the feasibility and sustainability of Digital Twin implementations. Recent studies by Al-Nasser et al. (2024) and Hadjidemetriou et al. (2023) emphasise that Digital Twin viability depends as much on integration effort as on conceptual design.

The benchmark results indicate that cloud-centric pipelines introduce higher integration overhead due to the configuration of serverless processing, databases, and API layers. This complexity provides flexibility, access control, and extensibility, but increases development and maintenance effort. Streaming-based pipelines reduce backend complexity by enabling direct data delivery to the visualisation layer, but shift responsibility for data validation, temporal context, and robustness to the consuming application. This redistribution of responsibility clarifies why MQTT broker–based pipelines are not inferior, but rather represent a different architectural choice aligned with immediacy and simplicity.

Design Guidance and Context-Aware Architectures

The observed trade-offs confirm that no single data pipeline architecture is universally optimal. Cloud-based pipelines are most appropriate for Digital Twins intended as long-term, data-rich systems requiring persistence, analytics, and enterprise integration. MQTT broker–based streaming is advantageous where low latency, simplicity, and rapid deployment are prioritised.

Consistent with architectural Digital Twin literature (Wang et al., 2013; Palmarini et al., 2018), these findings also point toward the value of hybrid architectures. By combining streaming mechanisms for immediate data delivery with cloud-based services for persistence and analytics, hybrid pipelines can balance responsiveness with long-term data value. Rather than competing alternatives, cloud-centric and streaming-based pipelines should therefore be understood as complementary components within a broader Digital Twin ecosystem.

Overall, the results reinforce that IoT data pipeline selection should be guided by intended Digital Twin functionality, lifecycle requirements, and operational context. By treating immersive visualisation as a demanding application context rather than the primary object of evaluation, this study highlights how backend architectural choices shape the robustness, deployability, and real-time behaviour of Digital Twin systems in the built environment.

Conclusion

This paper presented a comparative architectural analysis of IoT data pipeline approaches supporting real-time Digital Twin visualisation in the built environment. Using a controlled case study, three end-to-end pipelines; TTN–AWS–Unity, TTN–Azure–Unity, and TTN–MQTT broker–Unity were implemented and evaluated under identical sensing, visualisation, and mixed reality deployment conditions. The data integration layer was isolated as the primary variable.

The findings demonstrate that data pipeline architecture significantly influences Digital Twin responsiveness, complexity, and functional scope, particularly in high-demand visualisation contexts such as mixed reality. Cloud-centric pipelines support persistence, scalability, and long-term data management, making them suitable for sustained and data-rich Digital Twin deployments, albeit with higher integration and operational overhead. In contrast, streaming-based pipelines prioritise architectural simplicity and immediacy, supporting responsive real-time visualisation but offering limited inherent support for historical data handling.

Beyond platform-specific performance claims, the study provides architectural insight into how data integration design choices shape the practical behaviour of real-time Digital Twins when deployed in immersive and interactive environments. Future research may extend this work through hybrid pipeline architectures that combine low-latency streaming with cloud-based persistence and analytics, as well as through quantitative latency evaluation and deployment at larger or multi-building scales.

Overall, the findings confirm that no single data pipeline architecture is universally optimal. Instead, effective pipeline selection should be guided by the intended role, lifecycle stage, and interaction demands of the Digital Twin, underscoring the importance of aligning backend data architectures with application requirements rather than adopting a one-size-fits-all integration strategy.

References

Al-Nasser, H., Ahmad, M.E., Peralta, A., Patricia, C.G.C., Al-Zuriqat, T. Dragos, K. and Smarsly, K. (2024). Digital twin architectures in civil engineering: A systematic literature review, conference, Fachtagung Baustatik – Baupraxis: Hamburg, Germany.

Ansyah, A.S.S., Arifin, M., Alfan, M.B., Suriawan, M.V., Farhansyah, N.H., Shiddiqi, A.M. and Studiawan. H. (2023) MQTT Broker Performance Comparison between AWS, Microsoft Azure and Google Cloud Platform, 2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC), Mysore, India, 2023, pp. 1-6, doi: 10.1109/ICRTEC56977.2023.10111870.

AWS (2024a) Guidance for Digital Twin Framework on AWS. Available at: https://aws.amazon.com/solutions/guidance/digital-twin-framework-on-aws/ (Accessed: 13 January 2026).

AWS (2024b) Building your digital twin solution using the Digital Twin Framework on AWS. Available at: https://aws.amazon.com/blogs/hpc/building-your-digital-twin-solution-using-the-digital-twin-framework-on-aws/ (Accessed: 16 January 2026).

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y. (2020) 'Towards a semantic Construction Digital Twin: Directions for future research', Automation in Construction, 114, 103179.

Boschert, S. and Rosen, R. (2016) 'Digital Twin—The simulation aspect', in Mechatronic Futures. Cham: Springer, pp. 59–74.

Centenaro, M., Vangelista, L., Zanella, A. and Zorzi, M. (2016) 'Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios', IEEE Wireless Communications, 23(5), pp. 60–67.

El Ammari, K. and Hammad, A. (2019) Remote interactive collaboration in facilities management using BIM-based mixed reality, Automation in Construction ,107(2019), 102940,

Eisa, M., Younas, M., Basu, K. and Awan, I. (2020) Modelling and Simulation of QoS-Aware Service Selection in Cloud Computing, Simulation Modelling Practice and Theory, 103, 102108.

Grieves, M. and Vickers, J. (2016) 'Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems', in Transdisciplinary Perspectives on Complex Systems. Cham: Springer, pp. 85–113.

Hadjidemetriou, L., Stylianidis, N., Englezos, D., Papadopoulos, P., Eliades, D., Timotheou, S., Polycarpou, M.M. and Panayiotou, C. (2023) A digital twin architecture for real-time and offline high granularity analysis in smart buildings, Sustainable Cities and Society, 98, 104795,

Hameed, S., Bukhari, S.H.R., Rizvi, S.U.E.A., Tahir, M. and Aadil, F. (2025) A Comparative Analysis on the Services Offered by Amazon Web Services and Microsoft Azure, Concurrency and Computation: Practice and Experience, WILEY, pp. 1-19.

Kotas, C., Naughton, T. and Imam, N. (2018) "A Comparison of Amazon Web Services andMicrosoft Azure Cloud Platforms for High Performance Computing," in 2018 IEEE International Conference on Consumer Electronics (ICCE) (IEEE, 2018), pp. 1–4.

LoRa Alliance (2025) What is LoRaWAN? Available at: https://lora-alliance.org/about-lorawan/ (Accessed: 12 January 2026).

Lu, Q., Parlikad, A., Woodall, P., Ranasinghe, G. D., Xie, X., Liang, Z., Konstantinou, E., Heaton, J., & Schooling, J. (2020). Developing a dynamic digital twin at building and city levels: A case study of the West Cambridge campus.

Microsoft (2024) Microsoft Azure – Cloud Computing Services. Available at: https://azure.microsoft.com (Accessed: 13 January 2026).

Palmarini, R., Erkoyuncu, J.A., Roy, R. and Torabmostaedi, H. (2018) 'A systematic review of augmented reality applications in maintenance', Robotics and Computer-Integrated Manufacturing, 49, pp. 215–228.

Palumbo, F., Aceto, G., Botta, A., Ciuonzo, D., Persico, V. and Pescape, A. (2021) 'Characterization and Analysis of Cloud-To-User Latency: The Case ofAzure and Aws', Computer Networks, 184 (2021): 107693.

Qiu, S., Zaheer, Q., Ali, F., Wajid, S., Chen, H. Ai, C. and Wang, J. (2025). Exploring the impact of digital twin technology in infrastructure management: a comprehensive review. Journal of Civil Engineering and Management, 31, pp. 395-417.

Shahzad, M., Joe, H.M. Tah, Younas, M. and AlMukhtar, A. (2025) Technologies and techniques in digital twins for real-time data visualisation in building maintenance: A state-of-the-art review', Journal of Infrastructure Intelligence and Resilience, 4(4), 100185.

Tao, F., Zhang, H., Liu, A. and Nee, A.Y.C. (2018) 'Digital twin in industry: State-of-the-art', IEEE Transactions on Industrial Informatics, 15(4), pp. 2405–2415.

Thangavel, D., Ma, X., Valera, A., Tan, H.-X. and Tan, C.K.-Y. (2014) 'Performance evaluation of MQTT and CoAP via a common middleware', 21-24 April 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). IEEE, pp. 1–6.

The Things Network (TTN) (2023) The Things Network Documentation. Available at: https://www.thethingsnetwork.org/docs/ (Accessed: 15 January 2026).

Wang, X., Love, P. E. D., Kim, M. J., Park, C.-S., Sing, C.-P., & Hou, L. (2013). A conceptual framework for integrating building information modeling with augmented reality. Automation in Construction, 34, pp.37- 44.