From the 3D metric survey to a living digital shadow through IoT integration of real-time data. The case study of the Cavour Canal water bridge

Original
From the 3D metric survey to a living digital shadow through IoT integration of real-time data. The case study of the Cavour Canal water bridge / Matrone, F., Chinthakindi, S., Spadaro, A., Barbierato, L.. - (2026), pp. 303-310. (11th Intl. Workshop 3D-ARCH "3D Virtual Reconstruction and Visualization of Complex Architectures" Ancona (Ita) 10–12 February 2026) [10.5194/isprs-archives-XLVIII-2-W12-2026-303-2026].

Publisher:
Copernicus Publications

Published
DOI:10.5194/isprs-archives-XLVIII-2-W12-2026-303-2026

From the 3D metric survey to a living digital shadow through IoT integration of real-time data. The case study of the Cavour Canal water bridge.

Francesca Matrone [1], Saketh Chinthakindi [1], Alessandra Spadaro [1], Luca Barbierato [2]

[1] Geomatics Lab, Department of Environment, Land and Infrastructure Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy – (francesca.matrone, alessandra.spadaro)@polito.it; saketh.chinthakindi@studenti.polito.it
[2] Interdipartimental Center Ec-L - Energy Center Lab, Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy – luca.barbierato@polito.it

Keywords: scan-to-BIM**,** HBIM, digital shadow, IoT integration, monitoring

Abstract

The preservation of 19th-century architectural and civil engineering works, such as masonry bridges, is essential to ensure structural safety and the conservation of their historical value. Ageing and prolonged exposure to environmental actions progressively threaten these infrastructures, making planned maintenance a sustainable alternative to invasive restoration. In this context, digital replicas integrated with Internet of Things (IoT) technologies enable a proactive, data-driven approach to asset management. This paper presents a methodology for monitoring and preserving hydraulic historical structures, specifically the Cavour Canal bridge, a historic masonry water bridge within a wider network in northern Italy. A 3D metric integrated survey generated point clouds and orthophotos as the basis for the scan-to-HBIM workflow and for subsequent Historic Building Information Modelling (HBIM), supporting structural analyses and maintenance activities. The HBIM is enhanced through the integration of real-time data from continuous monitoring sensors. The resulting digital shadow supports an interactive, web-based management platform that can detect critical conditions and issue real-time alerts, improving the resilience and long-term management of historical infrastructures.

1. Introduction

The safeguarding and enhancement of 19th-century architectural and civil engineering works, such as masonry bridges, are crucial to preserve not only the engineering operas of their time, but also the historical significance embedded in their design and construction. As they age, exposure to environmental factors, such as weathering and natural wear, puts their structural integrity at risk, making preservation efforts crucial.

As emerged, planned maintenance plays a critical role in ensuring the long-term preservation of such infrastructures, offering a proactive alternative to costly and disruptive restorations. This approach, facilitated by the realization of a digital replica integrated with Internet of Things (IoT) data (Cui and Wu, 2025), not only ensures that the bridges remain functional and safe, but also protects their architectural stability, protecting them for future generations while minimizing the impact on the surrounding environment.

This contribution thus describes the methodology set up to preserve an example of this historical heritage. In particular, the preliminary results of the initial data acquisition campaign on the case study, the *ponte Canale Cavour* (Cavour Canal water bridge), which allowed for obtaining 3D integrated point clouds and orthophotos as the basis for the scan-to-HBIM (Historic Building Information Model) modelling and further analysis, with the definition of a digital shadow. In particular, attention was paid to the integration of real-time data into the HBIM model, which came from continuous sensors installed on the bridge for its structural monitoring via IoT devices. The core idea, in fact, is to create an interactive web-based platform for managing and maintaining the case study, achieved by using IoT sensors installed on the bridge.

1.1 Objective

The main objective is therefore to develop a model capable of detecting and alerting the management authority in the event of critical situations. In the past, similar infrastructures have suffered structural failures due to heavy rainfall and flooding, resulting in the accumulation of logs and debris around the bridge piers, which led to their collapse. The availability of a digital replica that can function as a “traffic light” system, indicating in real time significant variations or critical values detected by the sensors, is of vital importance for the effective management of this type of asset.

1.2 The case study

The Cavour canal is a hydraulic engineering project built between 1863 and 1866, primarily to support rice cultivation in northern Italy.

When it was inaugurated in 1869, the canal irrigated a surface area of approximately 200,000 hectares, with a flow rate of 110 m^3/sec for a total length of 82 km (Baratti, 1997). Among the infrastructure of this massive opera, water bridges allow the canal's water to override rivers. One example is the Ponte Cavour (Cavour bridge-canal). This bridge-canal, necessary to cross the Dora Baltea river, is the first (1864) and largest in Italy (Figure 1). Despite a general retrofit intervention carried out in 1998, some degradation phenomena undermining some piers is underway, in addition to water infiltrations that accelerate deterioration of the structure (Figure 2).

Figure 1. The Cavour canal water bridge over the Dora Baltea river.

https://doi.org/10.5194/isprs-archives-XLVIII-2-W12-2026-303-2026

Figure 2. Example of water infiltration and vegetation undermining the structure.

2. Related works

The built environment and architecture-related literature increasingly adopts “digital twin” terminology to describe data-enriched digital–physical representations of assets, yet the underlying maturity levels of BIM-based representations vary substantially. A widely used conceptual taxonomy distinguishes digital model, digital shadow, and digital twin based on the existence and directionality of data exchange between the physical asset and its digital counterpart. In this view, a digital model is a standalone virtual representation without automated data links; a digital shadow features automated one-way data flow from the physical system to the digital representation; and a digital twin implies bidirectional coupling such that changes in one domain can trigger updates or actions in the other (Kritzinger et al., 2018). While originally introduced and consolidated mainly in manufacturing-oriented reviews, this taxonomy has been explicitly invoked in construction and smart built-environment research to clarify common misclassifications and to frame the incremental pathway from design models to operational digital–physical systems (Sepasgozar, 2021).
A **digital shadow** occupies, therefore, an intermediate position, where the physical asset continuously (or periodically) feeds data, often via IoT sensors, BMS streams, or monitoring systems, into a digital representation that mirrors the asset’s state, yet does not close the loop back to the physical system. Built-environment scholarship emphasizes that the hallmark of a digital shadow is *automatic ingestion* of operational data, enabling monitoring, diagnostics, and context-aware visualization without implying automated control.
Obtaining a reliable digital shadow of the asset was thus the main objective of this work, providing the management body with an early warning tool able to ensure continuous monitoring even in risky and hazardous situations.

In this framework, recent research increasingly combines IoT sensing (e.g., microclimate data or occupancy) with HBIM/BIM-centric digital representations to support preventive conservation, monitoring, and decision-making. A common architectural pattern is to use an HBIM as the structural backbone and to link it to time-series sensor streams via middleware services, dashboards, or model servers. (Cursi et al., 2025) demonstrated the feasibility of this approach by embedding live sensor data into HBIM environments to visualize and analyze real-time conditions in heritage buildings. They adopted a three-layer architecture for the IoT-HBIM platform, from the sensor data acquisition, processing and visualization thanks to communication protocols such as BLE, MQTT, REST, IFC, to their integration into the HBIM model for real-time visualisation and analytics.

Some of the main challenges when dealing with the integration of IoT data and heritage digital models are still the interoperability (BIM/IFC semantics vs. sensor ontologies), model updating strategies, lifecycle governance (Lucchi, 2023) and data sharing through open platforms (Martinelli et al., 2025). For this reason, the final proposed workflow of this project will attempt to exploit already existing interoperable platforms (Matrone et al., 2024) and open source solutions both for data integration and sharing. This will allow the application of this methodology on a wider base, expandable to different management scenarios of similar cases, such as historic bridges or hydraulic infrastructures.

Finally, recent applied studies (Cecere et al., 2024) report end-to-end pipelines where real-time IoT devices enrich HBIM models for monitoring and predictive maintenance in archaeological or historical assets, reinforcing the practical relevance of HBIM–IoT coupling for heritage operations and activities as those foreseen for the Cavour Canal water bridge.

3. Methodology

The overall methodology foresees a straightforward workflow which starts with the point cloud generation, moves to the scan-to-HBIM process and then integrates the data coming from the IoT sensors installed on the bridge into the object-oriented model (Figure 3).

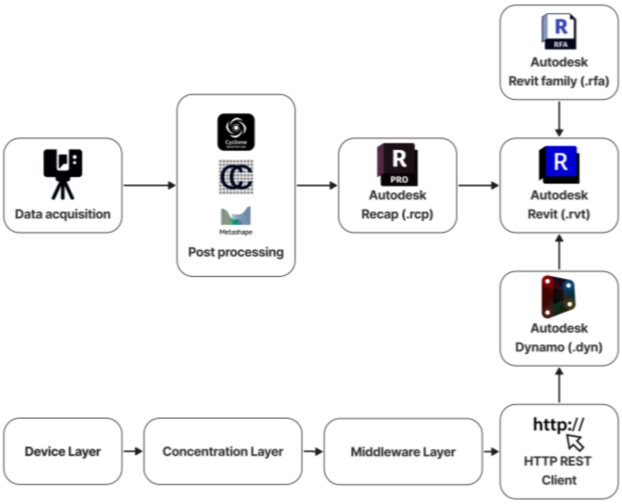


Figure 3. From 3D metric survey to IoT-HBIM workflow.

3.1 3D data acquisition

Data acquisition combined traditional topographic techniques (GPS/GNSS receivers and total stations), aerial photogrammetry from multispectral UAV (Uncrewed Aerial Vehicle), terrestrial laser scanning (TLS) and Mobile Mapping Systems (MMS) to ensure comprehensive coverage (Figure 4). UAV-based image acquisition was performed using DJI Mavic 3, capturing nadir and oblique imagery of canal facades and structural elements, particularly in areas exhibiting significant cracking and material degradation. Terrestrial laser scanning was conducted using a Leica RTC360 to capture high-resolution point clouds of the canal's masonry walls and internal hydraulic structures. MMSs, in particular a STONEX x70go, was used to acquire the surrounding built environment. The integration of UAV and TLS data enabled the documentation of both large scale geometry and fine surface details, while minimising occlusions and accessibility limitations, even under the arcades.

Figure 4. Data acquisition with Leica GS18, RTC360 and Leica MS50 (March 2025).

3.2 3D data processing and scan-to-HBIM

Photogrammetric processing of UAV imagery was carried out in Agisoft Metashape. The final mean RMSE on the Ground Control Points was 1.4 cm; while TLS data were processed and registered in Leica Cyclone Register software with a mean final error on the targets of 1.2 cm. Orthophotos of the facades and bridge riverbed were also created.

The two point clouds were then integrated into a single final point cloud (Figure 5) for import into Autodesk Recap.

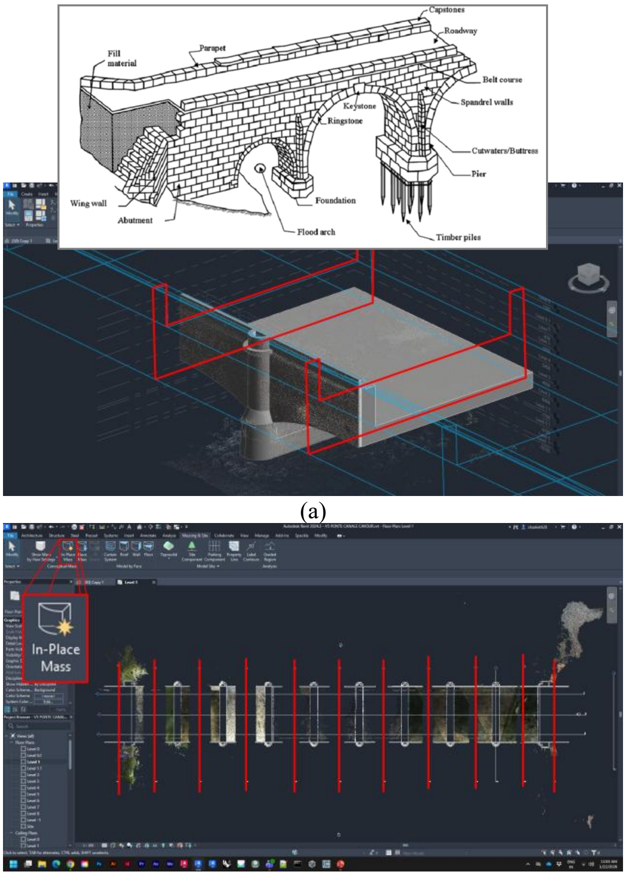
Figure 5. Final integrated point cloud used as a basis for the scan-to-HBIM.

The BIM modelling phase focused on the creation of an As-is parametric representation of the historical water bridge, characterised by deck, arches, piers, and retaining walls. Given the irregular geometry and heritage nature of the structure, the modelling strategy prioritised geometric fidelity to the surveyed point cloud over conventional parametric simplifications.

3.2.1 Structural decomposition and modelling strategy.

The bridge was decomposed into its main constructive components, namely the deck, arches, piers, and walls, to allow a structured and semantically meaningful BIM representation (Figure 6a). Additional elements, such as foundations and terrain context, were also modelled to provide a comprehensive understanding of the structure and its interaction with the surrounding environment.

The deck geometry was derived directly from the point cloud by generating accurate longitudinal and cross-sectional views. Section planes were extracted at regular intervals, allowing precise tracing of the deck profile (Figure 6a). Initially, the deck was modelled as a parametric family; however, this approach proved inadequate due to deviations between the family constraints and the actual geometry captured in the point cloud. Consequently, the deck was remodelled using *mass* modelling techniques (Figure 6b), which enabled closer adherence to the surveyed sections and ensured greater consistency.



(a)

(b)

Figure 6. Mass modelling of deck profile (a). Longitudinal and Cross section lines on point cloud (b).

3.2.2 Arches and pier modelling. The arches were modelled by segmenting the bridge geometry at the longitudinal section of the bridge and the north and south views in Autodesk Revit. This approach enabled precise control over the curvature of the arch profiles, which were reconstructed directly from sectional cuts through the point cloud. Each arch was modelled individually to capture local geometric variations resulting from construction methods and material ageing.

The piers were modelled using *wall* elements rather than mass objects. This decision was driven by both geometric and operational considerations: wall-based modelling allowed easier parameter control, better compatibility with rule-based operations, and improved integration with Dynamo workflows for future automation and data extraction.

The water cut portion of the pier was modelled separately as a family element, as it is a repeating element (Figure 7). The final HBIM was thus obtained (Figure 8) by adding the topographic surface derived from the UAV Digital Terrain Model (DTM).

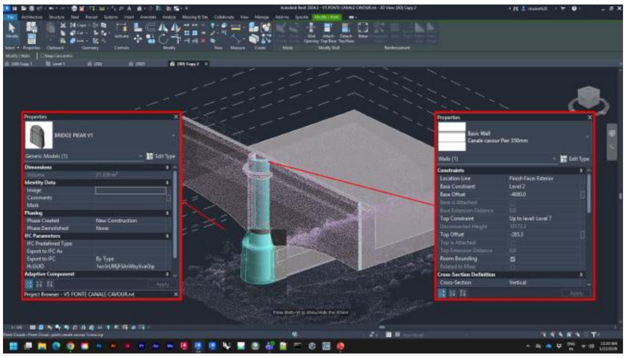


Figure 7. Pier (*wall* element) and riverbed part (*mass* model).

Figure 8. Integrated point cloud overlapped with the HBIM model.

3.2.3 IoT Sensor modelling and integration. To support the following BIM-to-IoT integration, sensor elements were modelled directly within the BIM environment as *electrical family* components. Each sensor instance (see paragraph 3.3.1) was enriched with custom instance parameters corresponding to sensor variables (VAR4, VAR5, VAR6, and VAR7), enabling the association of real-time or time-series data with specific structural locations. The sensors were placed strategically on the bridge elements to reflect the real monitoring scenario (Figure 9) and to facilitate direct linkage between the BIM model and external data streams (Figure 10).

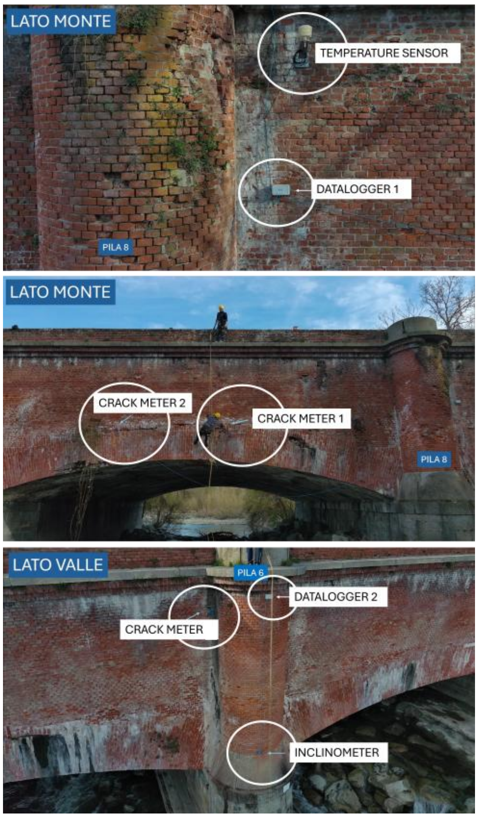


Figure 9. Location of some of the installed sensors.

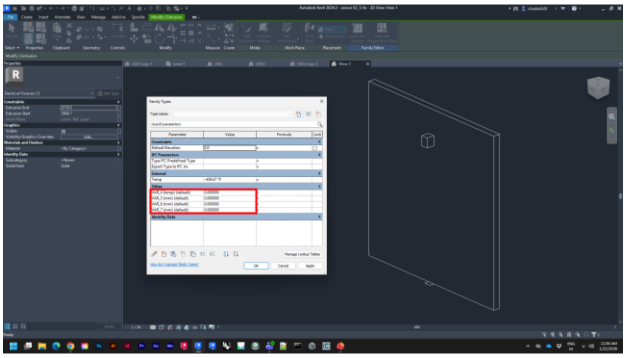


Figure 10. Autodesk Revit family *electrical fixture* and its instance parameter.

As shown in Figure 9, the two Data loggers are installed on the north and south facades. Data logger 1, named “north/mountain façade” collects data from temperature sensor and crack meter (1), crack meter (2), crack meter (3). Data logger 2, named “south/valley façade”, collects data from another crack meter and an inclinometer, reading the x-axis and y-axis.

In particular, there are two Data loggers:

- Datalogger 1- Lato Monte - “north/mountain façade”
- Datalogger 2- Lato Valle - “south/valley façade”

Datalogger 1 contains Temperature [\textdegree C] data (VAR4), Crack meter 1 [mm] (VAR5), Crack meter 2 (mm) (VAR6), Crack meter 3 [mm] (VAR7).

Datalogger 2 contains Inclinometer in x-axis [\textdegree] (VAR4), Inclinometer in y-axis [\textdegree] (VAR5) and Crack meter [mm] (VAR6).

3.3 IoT-HBIM integration

For the sensors data handling, we defined a four-layer IoT-HBIM integration framework designed to support real-time environmental and structural monitoring within the HBIM model, enabling the implementation of a digital shadow of the monitored asset. The architecture is conceived as a modular, layered system-engineering approach, ensuring robust data acquisition, resilient communication, scalable data management, and seamless integration with BIM-based visualisation environments.
As illustrated in Figure 11, the framework is structured into four functional layers:

- device layer,
- concentrator layer,
- middleware layer,
- visualisation layer.

Each layer encapsulates a well-defined responsibility and exposes explicit interfaces to adjacent layers, thereby promoting scalability, fault tolerance, and technology independence.

3.3.1 Device layer. The *Device Layer* constitutes the physical sensing infrastructure deployed directly on the bridge structure. Its primary function is to acquire in-situ measurements describing the asset's structural and environmental state. Sensors are installed at strategically selected locations identified as critical for monitoring deformation, displacement, and thermal effects.
In the proposed implementation, three sensor typologies are employed: i) crack meters, ii) inclinometers, and iii) temperature sensors. All devices are connected to the *Concentrator Layer* through wired communication links, a

design choice driven by requirements for high measurement reliability, deterministic latency, and immunity to electromagnetic interference, which are essential in long-term structural monitoring applications.

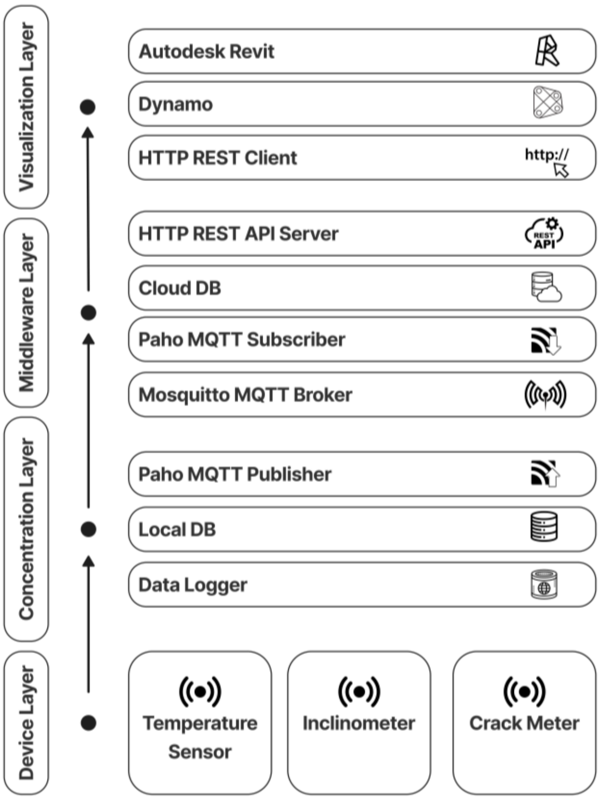


Figure 11. IoT–BIM integration architecture.

3.3.2 Concentration layer. The *Concentration Layer* operates as an edge gateway between the distributed sensor network and the cloud-based *Middleware Layer*. Its primary role is to aggregate sensor data, perform local preprocessing, and ensure reliable data delivery under variable network conditions.

As depicted in Figure 11, this layer comprises three core components:

- a Data Logger,
- a Local Database (DB),
- a Paho-based MQTT Publisher.

Together, these elements implement an edge-computing paradigm that enhances system resilience, reduces communication overhead, and preserves data integrity.

Generally speaking, the Data Logger acquires and manages data from multiple sensors via wired interfaces. Such interfaces provide deterministic communication, low latency, and robustness in harsh environmental conditions. It supports heterogeneous signal types (e.g., analog, digital, or serial) corresponding to the different sensors deployed in the *Device Layer*. Incoming measurements are conditioned, synchronised, and time-stamped before being forwarded either to the Local DB for buffering or to the MQTT publisher for transmission to the Middleware Layer.

The inclusion of a Local Database within the *Concentrator Layer* implements an edge-computing strategy to improve system resilience and efficiency. Two main motivations underpin this design choice: fault tolerance under network disruptions and local data reduction. First, monitoring sites frequently experience intermittent or unreliable internet connectivity. In such conditions, the Local DB functions as a persistent buffer, ensuring that sensor data generated during outages is retained locally and forwarded once connectivity to the *Middleware Layer* is restored (Liu et al., 2025). Second, high-frequency sensors can generate large volumes of data that are impractical to transmit continuously. The *Concentrator Layer* can therefore perform local preprocessing, storing raw measurements while publishing only aggregated values, detected anomalies, or event-based summaries. This approach significantly reduces bandwidth consumption and cloud storage requirements, which are critical considerations in scalable IoT deployments (Panchal, 2025).

Then, data transmission from the edge to the cloud is implemented using the Message Queuing Telemetry Transport (MQTT) protocol via the Paho open-source client. MQTT follows a publish--subscribe communication model and is specifically designed for constrained devices and unreliable networks (Gebremeskel, 2025).

In AEC monitoring scenarios, where network availability may be intermittent, and devices may operate under limited resources, MQTT offers minimal protocol overhead, with headers as small as 2 bytes. This makes it significantly more efficient than request-based protocols such as HTTP (ThingDash, 2025).

Within the proposed architecture, the concentrator normalises sensor data into structured payloads (e.g. JSON) and publishes them to dedicated topics on a central broker, enabling an asynchronous, scalable, and decoupled data flow to the *Middleware Layer*.

3.3.3 Middleware Layer. The *Middleware Layer* constitutes the logical integration core of the architecture, bridging the gap between telemetry-oriented IoT systems and data-centric BIM and enterprise applications. Its primary functions are data ingestion, persistent storage, semantic enrichment, and controlled data exposure through standardised interfaces.

As illustrated in Figure 11, this layer includes four main components:

- a Mosquitto MQTT Broker,
- a Paho MQTT Subscriber,
- a Cloud Database,
- an HTTP REST API Server.

The Mosquitto MQTT Broker is a widely used open-source implementation of the MQTT standard and serves as the central message-routing component of the system (Gebremeskel, 2025). It receives sensor data published by multiple edge concentrators and distributes them to subscribed middleware services.

Deploying the broker in the cloud or on a central server enables scalable handling of concurrent data streams and ensures loose coupling between data producers and consumers, thereby improving extensibility and fault tolerance.

The Paho MQTT Subscriber consumes messages from the broker and transforms transient telemetry into persistent, queryable data records. Upon message reception, the subscriber validates payload structure, performs preprocessing operations (e.g., unit conversion or timestamp normalisation), and forwards the processed data to the Cloud DB.

Since MQTT does not inherently support long-term data storage, the subscriber plays a critical role in bridging real-time messaging and historical data management, enabling longitudinal analysis and condition assessment.

The Cloud Database provides persistent storage and acts as a data historian for the monitoring system. It stores time-stamped sensor measurements, derived indicators, and contextual metadata required for structural analysis and BIM integration.
At this stage, schema mapping and semantic enrichment are performed. Sensor data are associated with BIM object identifiers, spatial locations, and asset hierarchies, ensuring traceability between physical measurements and their digital counterparts within the BIM environment.

The HTTP REST API Server provides controlled and interoperable access to monitoring data. While MQTT is optimised for machine-to-machine communication, most BIM tools and user-facing applications rely on request-response paradigms over HTTP.
The API exposes structured endpoints that allow visualisation and analytics tools to retrieve specific data subsets (e.g., by sensor, time window, or BIM element) without maintaining persistent broker connections. This decoupling is essential for integration with legacy AEC software ecosystems and supports scalable and secure data access (Singh, 2025).

3.3.4 Visualization Layer. The *Visualisation Layer* represents the final stage of the architecture, where monitoring data are contextualised and interpreted by human stakeholders. This layer operationalises the digital shadow by embedding live and historical sensor data within the BIM environment, enabling spatially informed analysis of structural and environmental conditions.
As shown in Figure 11, this layer consists of three tightly integrated components:

- an HTTP REST Client,
- Dynamo,
- Autodesk Revit.

The HTTP REST Client implemented within Dynamo is responsible for interfacing with the Middleware Layer. It periodically issues HTTP *GET* requests to the REST API Server and retrieves the latest sensor data stored in the Cloud DB.

Dynamo is a visual programming environment embedded within Autodesk Revit that enables custom data-processing and automation workflows. In the proposed architecture, Dynamo serves as the orchestration layer, retrieving external data, performing logical transformations, and updating BIM parameters programmatically.
Its capability to access and modify Autodesk Revit elements without altering the core application makes Dynamo the key enabler for implementing a live digital shadow within a traditionally static BIM environment.
Autodesk Revit serves as the primary BIM authoring and visualisation platform. It contains the authoritative geometric and semantic model of the asset; however, it is inherently a static, design-oriented system and does not natively support continuous data streams or direct IoT integration.
To overcome this limitation, the proposed architecture introduces an intermediary data-injection mechanism that enables dynamic parameter updates while preserving Revit's internal data structures and workflows.

The operational workflow is thus defined as follows:

1. Dynamo executes at a predefined temporal interval,
2. an HTTP request is sent to the Middleware Layer's API endpoint,
3. the API server returns structured sensor data (e.g., JSON),
4. Dynamo parses the response and extracts sensor identifiers, timestamps, and values,
5. sensor IDs are mapped to corresponding Revit element IDs,
6. Autodesk Revit parameters are updated with the latest measurements,
7. view filters and graphical rules automatically reflect updated values through colour-coded visualisation.

Through this mechanism, Autodesk Revit evolves from a static design model into an interactive monitoring interface, enabling intuitive assessment of structural conditions directly within the BIM environment and supporting informed decision-making across the asset lifecycle.

4. Results

Once the HBIM model with the geometries to be linked to the sensor data was obtained, the effective IoT integration was completed. In this contribution, as an initial stage of the research, the temperature data have been used to validate the pipeline, as they provide a wider range of values for the "traffic light" approach.

4.1 Device Layer

In the *Device Layer*, the sensors described in paragraph 3.2.3 were connected via wired connections to a data logger in the *Concentration Layer,* which acts as the primary data acquisition unit on site.

4.2 Concentration Layer

At the *Concentration Layer*, the data logger aggregates sensor readings and forwards them to a local data storage environment before transmission, ensuring continuous data handling and buffering.
Sensor data is structured and streamed through the *Middleware Layer* designed to simulate continuous data acquisition and communication.

Time series datasets were processed and transmitted using the Paho MQTT protocol, enabling lightweight and efficient message exchange suitable for infrastructure monitoring applications.

4.3 Middleware Layer

At the *Middleware Layer,* a Mosquitto MQTT broker was employed to manage the publish-subscribe communication model, acting as an intermediary that decouples sensor data producers from data consumers.

The MQTT publisher continuously transmitted sensor data streams, while a subscriber service listened to predefined topics and forwarded the incoming data to a cloud-based database, which was updated dynamically. An HTTP REST API server was implemented to expose the most recent sensor values in JSON format, enabling standardised access to live monitoring data by external applications and services under the webhost.

4.4 Visualization Layer

At the *Visualization Layer*, the HBIM model served as the central interface for data representation. A Dynamo script was created to retrieve live sensor data via HTTP requests using a web request node (Figure 12).

The returned JSON data were parsed and mapped to corresponding BIM sensor elements using instance parameters embedded within the sensor families. VAR4, VAR5, VAR6, VAR7 of sensor information from the real world are mapped to the sensor Revit family in the virtual world.
As previously mentioned, for visualisation reasons, only VAR4 (the temperature sensor) is chosen, to show the variation and change of colour in the model element in Autodesk Revit.

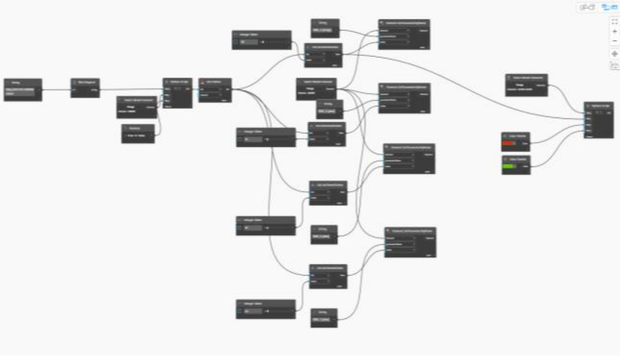


Figure 12. Dynamo script to retrieve data from a web request and set the data as parameters.

In particular, the parameter VAR4 was used to store temperature-related values. A Python node within Dynamo was employed to evaluate the incoming data and apply rule-based visualisation logic. Sensor elements were colour-coded based on temperature thresholds, with values below 6 °C displayed in green and values exceeding 6 °C displayed in red. This visual feedback enabled immediate identification of potential thermal anomalies directly within the BIM environment (Figure 13).

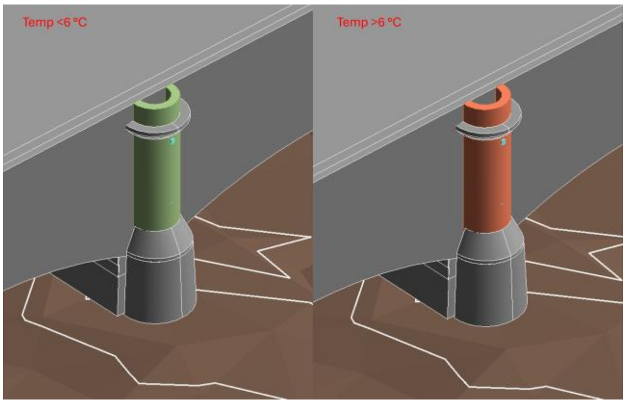


Figure 13. Automatic change of element colour based on the temperature parameter in the Autodesk Revit family.

The Dynamo workflow was configured to execute periodically with 1000ms, allowing the BIM model to update automatically as new data became available. This approach transforms the static As-is BIM into a dynamic, data-driven representation, supporting the conceptual implementation of a digital shadow for the Cavour Canal water bridge.

4.4.1 Dynamo nodes. With respect to Figure 12, this section provides a detailed description of the main nodes. Figure 14 shows the nodes from 1 to 4.

1. Node 1. *String*: text-based node input, the HTTP link used to publish the JSON live data.
2. Node 2. *Web Request*: retrieve data from the JSON link.
3. Node 3. *Python script*: A custom Python script was developed to read the JSON file from the web request and parse the selected element information from the select model element node. Three inputs are necessary for the Python script: the JSON data from the web request, the selected model element, and a Boolean value (true or false). The final output consists of two lists: the first is the name of the data container, and the second is the data to be set to the element.
4. Node 4. *List Flatten*: it is used to flatten two lists into a single list to set the parameters for the following nodes.

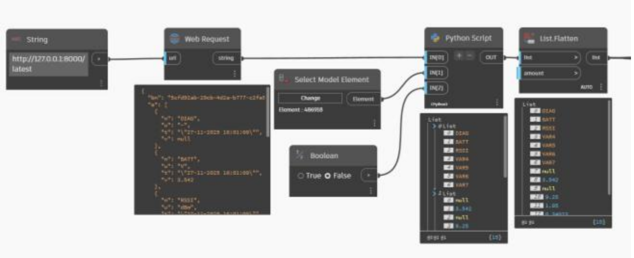


Figure 14. Dynamo script showing the connection of *String*, *Web request*, *Python script* and *List flatten* nodes.

Figure 15 shows nodes 5 and 6.

5. Node 5. *List get item at index*: this node allows for getting specific information from the given list (Figure 15).
6. Node 6. *Element set parameter by name*: this node sets the given value to the selected element parameter on the bases of the parameter name. It has three inputs: the "element" input is given with the select model element node, the "parameter name" is given with the string node, which is case sensitive and should match the parameter name created in the Revit model, and the "value" is the data requested to set to the parameter. In this case, it is from the web request temperature sensor value.

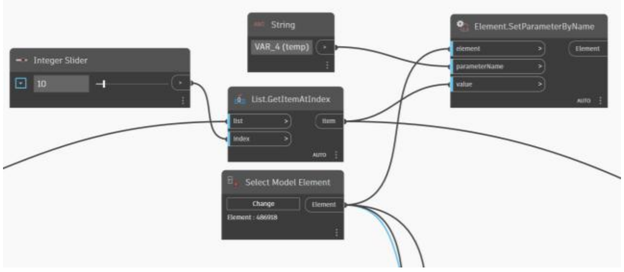


Figure 15. Dynamo script showing the connection of *List get item at index* and *Set parameter by name* nodes.

7. Node 7. *Python script node* (Figure 16): this custom Python script node triggers the temperature value, which is retrieved from the flattened list, and if this value is less than a certain threshold (6 in this specific case), the colour-code assigned to the selected element is green; if higher than 6, the colour-code will be red. This node has four inputs: "input 0" to select the model element pier, "input 1" to get the temperature value that triggers the colour change, "input 2" for the red colour palette, and "input 3" for the green colour palette.

The colours and thresholds could be arbitrarily set and constitute a future work of this research. In fact, in collaboration with structural monitoring experts, the most appropriate value ranges for the crack meters and inclinometers will be defined, and the visualisation in the HBIM model will be updated accordingly.

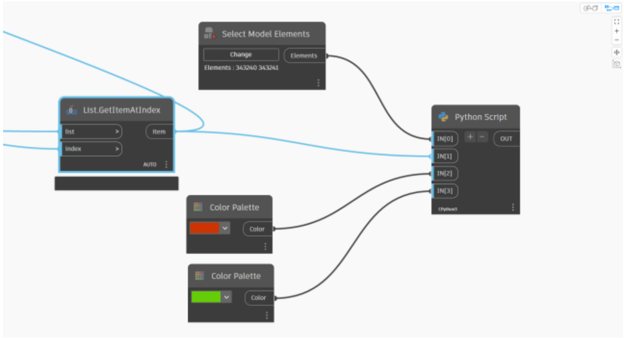


Figure 16. Dynamo script showing the connection of *the Python script* and its inputs to change the colours of the selected element in the model.

5. Discussions and conclusions

This paper presented a methodology for integrating IoT-based monitoring data into an HBIM model to create a digital shadow for the preventive conservation of historical hydraulic infrastructure.
The proposed workflow demonstrates how an integrated 3D metric survey, scan-to-HBIM process, and IoT-based monitoring architecture can support the evolution of a static digital replica into a living digital shadow for historical hydraulic infrastructures. The Cavour Canal water bridge case study demonstrates the feasibility of embedding real-time sensor data within an HBIM environment to enable continuous condition monitoring and early warning capabilities for ageing masonry structures.

A key strength of the approach lies in the modular four-layer IoT-HBIM architecture, which ensures scalability, resilience, and interoperability by separating sensing, data aggregation, middleware, and visualisation components. This design facilitates the reuse of open protocols and standards and avoids tightly coupled solutions, which are often difficult to maintain in long-term heritage monitoring scenarios. The *As-is* HBIM, derived from integrated UAV, TLS, and MMS surveys, provides a geometrically accurate and semantically structured backbone for linking sensor data to specific structural elements.

At the current stage, the system aligns with the concept of a digital shadow, as data flows remain unidirectional from the physical asset to the digital model. This configuration supports monitoring and diagnostics without automated feedback or actuation, which is consistent with conservation-oriented constraints typical of heritage assets. The rule-based “traffic light” visualisation implemented in Dynamo provides an intuitive interface even for non-expert users, though the definition of reliable thresholds requires further validation in collaboration with structural monitoring specialists.

Future work will focus on integrating additional data (thermal images and multispectral orthophotos), refining alert thresholds based on long-term data analysis, extending web-based visualisation tools jointly with Unity, improving an open framework for the IoT integration and testing the methodology on other historic bridges and hydraulic assets to assess its generalisability.

Acknowledgements

The authors would like to thank the *Associazione di Irrigazione Ovest Sesia* (AIOS) and, in particular, Livio Bourbon, who supported the data acquisition and sensor placement.

References

Baratti, C., 1997. *I Fontanili del Novarese.* Novara: Associazione Irrigazione Est Sesia, Stampa.

Cecere, L., Colace, F., Lorusso, A., & Santaniello, D., 2024.. Predictive Maintenance of an Archeological Park: An IoT and Digital Twin Based Approach. In *International Conference on Human-Computer Interaction*. Cham: Springer Nature Switzerland, 323-341.

Cui, H., & Wu, J., 2025. How Architectural Heritage Is Moving to Smart: A Systematic Review of HBIM. *Buildings*, 15(15), 2664.

Cursi, S., Martinelli, L., Calcerano, F., Calvano, M., & Gigliarelli, E., 2025. Digital twin and HBIM-to-IoT for smart museum management–the case of the DART Museum Chiostro del Bramante. SCIRES-IT-SCIentific RESearch and Information Technology, 15(2), 71-84.

Gebremeskel, H., 2025. Understand the protocols behind the IoT messaging. *CloudAMQP.* https://www.cloudamqp.com/blog/understand-the-protocols-behind-iot-messaging.html. Last access. Jan. 2026.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W., 2018. Digital Twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline,* 51(11), 1016-1022.

Liu, X., Antwi-Afari, M. F., Li, J., Zhang, Y., & Manu, P. (2025). BIM, IoT, and GIS integration in construction resource monitoring. *Automation in Construction*, 174, 106149.

Lucchi, E., 2023. Digital twins for the automation of the heritage construction sector. *Automation in Construction*, 156, 105073.

Matrone, F., Colucci, E., Iacono, E., & Ventura, G. M., 2023. The HBIM-GIS Main10ance platform to enhance the maintenance and conservation of historical built heritage. *Sensors*, 23(19), 8112.

Martinelli, L., Calcerano, F., Adinolfi, F., Chianetta, D., & Gigliarelli, E., 2025. Open HBIM-IoT Monitoring Platform for the Management of Historical Sites and Museums. An Application to the Bourbon Royal Site of Carditello. International Journal of Architectural Heritage, 19(2), 153-170.

Panchal, S., 2025. How BIM and Big Data are Transforming the AEC Industry, *Techture*. https://www.techture.global/blog/bim-and-big-data-integration. Last access: Jan. 2026.

Sepasgozar, S. M., 2021. Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. *Buildings*, 11(4), 151.

Singh, V. 2025. What Is BIM in Construction? Exploring Its Integration with AI & IoT. *Capstone Engineering Simplified*. https://www.capstonees.com/blog/what-is-bim-in-construction-exploring-its-integration-with-ai-iot. Last access: Jan 2026.

ThingDash, 2025. MQTT vs AMQP Deciding on Your IoT Protocol. https://www.thingdash.io/blog/mqtt-vs-amqp-deciding-on-your-iot-protocol. Last access: Jan 2026.