The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-9-2025
30th CIPA Symposium “Heritage Conservation from Bits:
From Digital Documentation to Data-driven Heritage Conservation”, 25–29 August 2025, Seoul, Republic of Korea

# A Data-driven Information Modelling Approach for Cultural Heritage

Yuxin Lei [1], Nazarena Bruno[1], Riccardo Roncella [1]

[1] Dept. of Engineering and Architecture (DIA), University of Parma, Italy – (yuxin.lei, nazarena.bruno, riccardo.roncella)@unipr.it

**Keywords:** Cultural Heritage Documentation, HBIM, Spherical Photogrammetry, Data Co-registration, XR.

**Abstract**

The documentation of architectural heritage requires interdisciplinary collaboration and advanced information management. Historic Building Information Modelling (HBIM) and Geographic Information Systems (GIS) have enhanced cultural heritage documentation, while eXtended Reality (XR) technologies have improved accessibility and interactive engagement. However, challenges persist in data acquisition, modelling, and classification, limiting the practical implementation of comprehensive heritage information systems. This study introduces a data-driven information modelling framework optimized for cultural heritage, enabling the integration of 3D survey data and heterogeneous datasets. The framework supports cost-effective reuse of existing datasets by ensuring interoperability through a Broker Database and a Web Programming Interface (Web API). The system facilitates data structure abstraction, data description, geometric management, and spatial analysis, treating diverse datasets as informative layers. Allowing co-registration/coordinate conversion of 3D models, point clouds, and spherical images in/to a unified spatial reference system, it enhances data accessibility while reducing modelling efforts. The paper illustrates the structure and features of the proposed system, focusing on its brokering role and modular framework. Two case studies demonstrate the framework’s capability and scalability: the Parma (Italy) Cathedral and the Church of Santa Maria della Steccata in Parma (Italy). The system outputs are presented through a Virtual Reality application in Unity, providing dynamic geometry visualization, data querying, and immersive exploration.

## 1. Introduction

Architectural heritage documentation is a complex and multifaceted process that requires integrated knowledge, interdisciplinary collaboration, and advanced information management. The heterogeneous information characterizing historic buildings significantly influences all proposals and actions undertaken by the various specialists involved. However, the diverse qualitative and quantitative datasets representing this knowledge – such as geometric data, environment monitoring and Internet of Things (IoT) data, material and structural analysis, historical and archival documents, and conservation or restoration records – are often collected and analysed by different specialists using varied hardware and software tools (López et al., 2018; Quattrini et al., 2017). This results in scattered, fragmented datasets, complicating effective interdisciplinary collaboration (Penjor et al., 2024).

Historic Building Information Modelling (HBIM) and Digital Twin systems have significantly expanded the possibilities for documenting and managing cultural heritage (Boje et al., 2020; Botín-Sanabria et al., 2022; Lovell et al., 2023). In response to the challenges in the management of existing knowledge for historic buildings, researchers have extensively explored semantic enrichment and data management of HBIM model (Quattrini et al., 2017). A common strategy involves linking HBIM models to external databases specifically tailored to proprietary Building Information Modelling (BIM) software environments (Bruno and Roncella, 2019; Cursi et al., 2022; Rodrigues et al., 2019). These solutions facilitate the structured management of data, facility information, and archival documentation.

In parallel, Geographic Information Systems (GIS) have always been adopted for the spatial representation, analysis, and management of heritage data. Although typically used for territorial scales applications, GIS can be valuable tools for documenting the context of cultural heritage buildings. In some cases, they have also been adapted for building-level documentation (Bitelli et al., 2019). In 2020, (Sánchez-Aparicio et al., 2020) developed a Web-GIS platform that integrates 3D point clouds, 360-degree panoramic imagery, and IoT data into a geospatial database accessible to both expert and non-expert users. The integration of BIM and GIS has become a research focus in cultural heritage knowledge management, aiming to combine the advanced editing functionalities of semantic HBIM models with the data-handling strengths of GIS (Yang et al., 2020). Ongoing research continues to explore effective strategies for aligning these systems, particularly in addressing interoperability and minimizing information loss during data exchange (Xia et al., 2022).

At the same time, advances in extended reality (XR) technologies are transforming how users interact with cultural heritage, improving accessibility and supporting on-site maintenance. Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) play an increasing role in the interpretation and presentation of cultural heritage sites, adding immersive experiences and interactive engagement to platforms dedicated to heritage data sharing (Casini, 2022). The topic of how to leverage existing documentation as content to streamline the creation of virtual experiences for cultural heritage has drawn considerable attention among scholars (Banfi et al., 2019; Pybus et al., 2019). Nonetheless, achieving a balance between historical accuracy, data richness, and user interactivity continues to present significant technical challenges and requires specialized expertise (Ferdani et al., 2020).

In all these domains, research is still ongoing in several key areas to meet the specific requirements of Cultural Heritage (CH) documentation, particularly in 3D reconstruction, semantic classification, and advanced information management.

However, developing comprehensive information systems for CH remains costly and complex from many points of view: i) acquiring accurate geometric data is often resource-intensive and time-consuming, requiring high-cost instruments and prolonged acquisition sessions; ii) modelling the acquired/surveyed information is not easily automatable and typically demands significant manual effort, which can hinder the actual implementation of such systems; iii) to make 3D models truly informative, they generally need to be segmented and classified, linking them to relevant information: a process that is often long, scarcely automated and labour-intensive; iv) finally, especially for real-time visualization purposes, it is often necessary to simplify 3D objects or parametrize models, which typically leads to a significant loss in geometric accuracy and completeness.

Building on these premises, this paper presents the outcomes of an ongoing research project aimed at developing a data-driven information system framework optimized for CH management. The framework addresses the identified challenges by enabling the dynamic integration of heterogeneous datasets – including pre-existing ones, not originally acquired for the specific application under development – thereby reducing acquisition time and cost and optimizing development efforts. The integration of diverse data sources leverages the complementary strengths of different data types and makes them truly interoperable, without altering their original structure (Lei et al., 2024).

# 2. Materials and Methods

## 2.1 System Overview

The developed framework is based on a data-driven architecture that enables the integration of heterogeneous datasets through a brokering approach. This design preserves the original characteristics of each dataset while promoting their interoperability. Dedicated brokering components manage data access and provide the necessary mediation and harmonization functionalities, ensuring consistency, integrity, and reusability. Acting as an intermediary between fragmented data sources and end-user applications, the broker offers a unified access point through standardized protocols and interfaces. Rather than relying on client-side operations, all transformation and integration processes are centralized within the brokering system, thereby avoiding any direct manipulation of the source data.

The brokering architecture is based on a Web Application Programming Interface (Web API) coupled with a Broker Database (BrokerDB). Together, these components streamline the data flow between existing fragmented data sources and diverse application environments. The Web API handles all operations related to data import, transformation, harmonization, enrichment, and access, while the BrokerDB serves as the core repository and coordination layer. Although PostgreSQL was used for all the presented case studies (mainly to take advantage of the PostGIS extensions, which allow more efficient processing of geometric and spatial data), the Web API is capable of interacting with most of the RDBMS (Relational DataBase Management Systems) currently available.

As illustrated in Figure 1, from an information flow perspective, the Web API supports a set of core functionalities that enable seamless data integration, management, and access:

**2.1.1 Connect/Import:** This functionality allows both the importation of newly acquired datasets (*Input Data*) and the connection to existing *External Databases*. In the first case, the system supports the ingestion of 3D survey data, such as point clouds, Triangular Irregular Network (TIN) models, meshes, and equirectangular imagery. These datasets are processed through dedicated control modules within the Web API and, if some transformations occur, they are stored directly in specific tables within the BrokerDB. In the second case, the system can establish connections to databases managed by different Database Management Systems (DBMSs). In this case, the data remain physically stored in their source systems, which can be configured as read-only to preserve their integrity. The database connection configurations are maintained in dedicated management tables within the BrokerDB. In these cases, if additional attributes or tables need to be associated with externally connected databases, these are stored separately in the BrokerDB to avoid altering the original data sources.

**2.1.2 Semantic Management:** This module allows the enrichment of both imported and externally referenced data with semantic attributes and auxiliary tables. All additional semantic layers are stored in the BrokerDB, ensuring that enhancements are performed non-invasively, without modifying the source data.

**2.1.3 Transform:** This functionality includes format conversions, geometric transformations, harmonization of Coordinate Reference Systems (CRS), and the co-registration of datasets. All transformation operations are executed by dedicated Web API modules, and their outputs are stored in the BrokerDB to maintain traceability and consistency across datasets.

**2.1.4 Metadata Management:** To support interoperability and data discovery by external applications, the system enforces consistent documentation of all datasets through standardized metadata structures. A centralized *Metadata Table* in the BrokerDB ensures that both internally stored and externally connected datasets are described using a common schema. This enables external applications to efficiently query, visualize, and filter data while maintaining semantic clarity.

**2.1.5 Data Query:** The Web API provides structured access to both integrated and externally referenced data. It functions as an interface between end-user applications and the available datasets – whether stored in the BrokerDB or accessed via external connections – allowing users to perform queries and retrieve relevant information in an application-ready format.

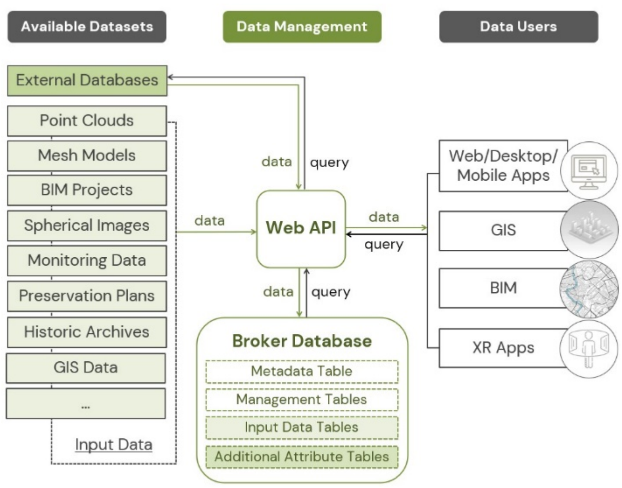


Figure 1. Graphic scheme of the proposed system.

The Web API's functionalities are organized into a set of specialized controllers, each responsible for a specific subset of data management operations described above. These controllers implement the core logic of the system in a modular and scalable way, allowing new functionalities to be added as needed.

To guarantee data security and appropriate access control, the system defines different user roles with specific levels of access to the available functionalities. Currently, in all the developed case studies, a set of just three different roles (Admin, Developer, Client), provided an appropriate data-access and manipulation policy for the system: Client can only access or query the data through specific Web API functionalities and insert new data only in specific BrokerDB tables; Developer, on the contrary, can define the structure of the whole application, can add, transform or inject data in the BrokerDB, specify new connections and manipulate metadata and resources; finally, the Admin role, allows full control over the whole BrokerDB (including roles and users administration). This role-based model ensures that users can only access and interact with the system components relevant to their responsibilities and expertise.

To further support and simplify user interaction across different stages of use, from system deployment to operational use, a graphical user interface (GUI) has been developed (Figure 2). This GUI serves as an additional client application that provides data access features, making it easier and more user-friendly to configure the BrokerDB through the Web API functionalities. Following user authentication, access is granted to functionalities according to role-based privileges. The GUI is organized into function-specific tabs, each corresponding to a Web API controller. This layout reflects the modularity of the system and enables progressive scalability, allowing the interface to evolve in parallel with the underlying functionalities. Alternatively, different GUIs (e.g., a web application) can be developed for specific use cases. By abstracting the complexity of the brokering operations, the GUI facilitates intuitive interaction with the system for users with varying technical backgrounds.

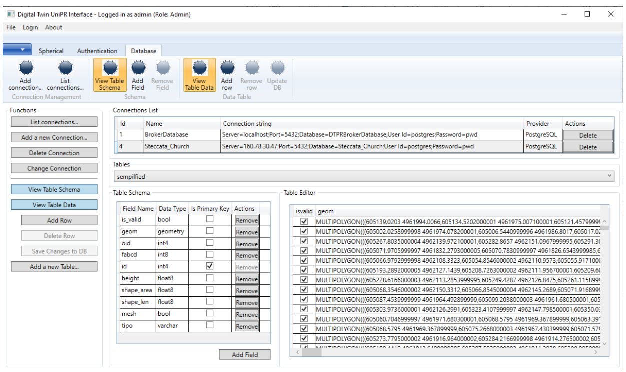


Figure 2. Example tab of the GUI developed to facilitate system management and interaction with the Web API functionalities.

## 2.2 Data Import

Once the BrokerDB is initialized, a set of controllers handles the importation, registration, and standardization of heterogeneous data originating from various sources used in cultural heritage documentation. These include both structured data from existing databases and unstructured data such as 3D models, images, and point clouds.

**2.2.1 Integration of Existing Databases:** The system allows for the integration of pre-existing databases by establishing connections to various RDBMSs, including PostgreSQL, SQL Server, MySQL, Oracle, and SQLite. This integration is handled transparently by the Web API, which manages the communication protocols and compatibility requirements between different systems. To connect a database, users are only required to provide a connection string, specify the type of RDBMS, and indicate whether the database should be accessed in read-only mode or with editing permissions. Once connected, the database becomes accessible through the Web API, which abstracts the underlying complexity of each RDBMS. Basic management functions – such as table creation and deletion, field and record editing, and data insertion – are available directly through the GUI, allowing users (developer) to interact with all connected databases from a single, unified access point. This avoids the need to use multiple native RDBMS interfaces, which are often less intuitive and more technically demanding.

**2.2.2 Point Clouds:** The system allows the integration of point cloud datasets, which can be directly imported into the BrokerDB through a dedicated Web API function. Point clouds are stored in binary format within specific tables. While full-resolution import is the default behaviour, certain use cases or application constraints may require a more lightweight representation of the data. To address this, the system includes optional preprocessing features that allow users to either subsample the point cloud or partition it into spatial tiles of user-defined dimensions. These operations, fully managed server-side, enable users to optimize the dataset according to specific visualization or performance requirements, without the need for external software.

**2.2.3 Spherical Images:** A specific Web API function has been developed to process and import georeferenced equirectangular panoramic images, which serve as highly detailed 2D representations of real-world 3D scenes. These 360° images are valuable for immersive visualization and can significantly enhance spatial understanding in cultural heritage contexts. To ensure correct spatial registration, panoramic images must be both georeferenced and rotated according to their acquisition geometry. The system can also generate equirectangular images from photogrammetric blocks acquired with spherical cameras. One of the currently available Web API modules reads data from Agisoft Metashape projects (with planned extensibility to other photogrammetric software), extracting raw image frames (usually fisheye images), exterior orientation parameters (camera position and rotation), and interior orientation parameters (camera calibration and distortion model). Based on these parameters, an equirectangular image is generated and stored in the BrokerDB. Another functionality (at this stage of writing still under development) allows the developer to download spherical images, along with their georeferencing information, directly from external services such as Google Street View. Each image is saved in a dedicated table as a multiresolution pyramid, along with its associated positional and orientation metadata. This ensures that the panoramic image can be correctly co-registered with the 3D scene during visualization, supporting immersive experiences or augmented spatial interfaces (see Section 3.2 for examples of application).

**2.2.4 BIM Models:** The system supports integration with BIM models. A dedicated controller has been developed to interface with Autodesk Revit projects (with planned extensibility to other BIM platforms). The Web API connects via an add-in to Revit software, open a user-defined project file and make Revit extract the geometric entities, which are converted into mesh models and stored in the BrokerDB. Non-geometric attributes associated with the BIM elements – such as materials, classifications, and functional metadata – are likewise imported and maintained in dedicated relational tables.

**2.2.5 Additional 3D and Multimedia Data:** In addition, the


https://doi.org/10.5194/isprs-archives-XLVIII-M-9-2025-813-2025  

system supports the import of 3D geometric models, raster images, PDF documents and any other kind of digital documents, which are stored as binary objects within dedicated tables in the BrokerDB.

### 2.3 Data Description and Documentation

Building upon the metadata infrastructure introduced in the system overview, the system enables semantic enrichment, discoverability, and contextual understanding of datasets through a centralized metadata structure. Every dataset integrated into the system must be described through an associated metadata entry to ensure its discoverability by end-user applications. This approach reflects the principles outlined in the INSPIRE Technical Guidance for the implementation of dataset and service metadata based on ISO/TS 19139:2007, which emphasizes the importance of enabling users to locate spatial datasets and related services and assess their suitability for specific purposes. In alignment with this guidance, the framework ensures that metadata records are accessible to client applications, which may filter, visualize, or query the available datasets based on metadata-driven search and categorization mechanisms.

A dedicated table has been implemented within the BrokerDB to store metadata entries for each dataset, including both internal tables and those residing in externally connected databases. The metadata schema implemented in the system adheres to the INSPIRE metadata specification, so each dataset is described using a set of standardized fields, including resource title, resource type, resource locator, temporal content, data type, and reference date. Beyond these core metadata fields, each entry is associated with a corresponding JavaScript Object Notation (JSON) file that stores the detailed table schema and its relationships with other tables. This additional layer of structural description enhances transparency and supports automated data handling and validation within the system.

### 2.4 Data Transformation and Co-registration

When datasets are imported into the system, any that include geometric or geographic data trigger the addition of a corresponding entry in the metadata BrokerDB table, recording the dataset's CRS. While some RDBMS solutions (e.g., PostgreSQL with the PostGIS extension) offer built-in support for CRS storage and transformation, it is essential that the Web API handles all CRS-related operations. This approach ensures consistent interoperability across datasets originating from different database systems. During the data ingestion or preparation phase, developers have the option to request a permanent transformation of the dataset's CRS. In such cases, the data is stored in BrokerDB as a transformed, static copy of the original, now with updated coordinates. This is especially beneficial for datasets where on-the-fly CRS conversion would be computationally expensive, such as meshes or point clouds containing millions of points, thereby improving data access performance for client applications. At the same time, the Web API also provides runtime CRS conversion capabilities, enabling client applications to implement custom conversion logic when needed. This dual approach – supporting both pre-converted storage and dynamic transformation – ensures maximum flexibility, performance, and consistency in the handling of spatial data within the system.

Beyond co-registration, the system supports additional transformation operations: i) 2D-to-3D extrusion: 2D GIS datasets can be extruded into 3D geometries and stored as binary objects in the BrokerDB; ii) Bounding box computation: Bounding boxes are calculated for all geometric datasets to enable spatial filtering and selection. These are particularly useful for limiting the display or querying of datasets to specific spatial extents; iii) Localization of non-geometric data: Datasets without inherent geometry can be geolocated and visualized as interactive hotspots within 3D environments.

These transformation tools ensure that all integrated datasets are spatially consistent and interoperable. Information is organized in layers, with each dataset treated as a distinct yet co-registered information layer. This layered structure is inspired by GIS and supports multi-source spatial analysis. A key advantage of this architecture is the ability to query across heterogeneous datasets, combining their complementary strengths. For example, point clouds provide the most detailed and accurate representation of architectural elements because they are not subject to subsequent modelling operations such as simplification or parameterization. However, they are generally not semantically classified and therefore do not directly convey information. Similarly, 360-degree panoramic images are faithful 2D representations of 3D scenes, allowing high detail object detection, condition mapping, but do not provide semantic information. On the other hand, HBIM models offer comprehensive semantic information, but their level of accuracy can vary considerably depending on modelling processes such as simplification or parameterization. GIS provide spatially organized and easily queryable data, but are typically used for 2D representations, which can reduce their effectiveness in 3D applications.

This layered and co-registered architecture allows datasets to be queried transversally. As illustrated in the case studies, users can interact with a point cloud to examine fine geometric details without the need to segment or annotate it directly. Instead, semantic information can be retrieved from an underlying simplified model (such as a 3D GIS layer or an HBIM), where attributes and classifications are already structured.

A similar approach applies to equirectangular panoramic images. Since they are georeferenced and spatially aligned with the 3D scene, they offer a visually rich interface that retains high detail. Users can interact with the panorama to access the information stored in the underlying model. This setup combines the visual fidelity of raw survey data with the semantic depth of modelled or structured datasets, allowing users to navigate across multiple data types while maintaining a unified and coherent spatial context.

### 2.5 Data Accessibility via Unity Application

To validate the system's capability to manage and deliver heterogeneous cultural heritage datasets for end applications, a Unity-based testing environment has been implemented. Unity was selected as the initial deployment platform due to its flexibility in rendering complex 3D environments, its compatibility with web-based data pipelines, and its capability to create customized immersive and interactive experiences. To ensure a generalized approach and avoid overly specific solutions, a standard template in Unity has been designed and implemented. This template includes essential functionalities while allowing for the addition of specialized features tailored to the unique requirements of each cultural heritage site. The key functions include dynamic geometry loading for efficient visualization, diverse 3D representation options with varying levels of detail, including simplified geometry, textured meshes, point clouds, and spherical images, data querying and interactive display for historical and preservation information, and immersive exploration in VR.

A central component of the Unity template is the *ProjectSettings* class, implemented via Unity’s editor scripting framework. This class enables developers to configure the connection to the Web API and define dataset-specific behaviours directly within the Unity Inspector. Upon entering the Web API URL, the system connects to the Web API to filter and retrieve information for all geometric datasets from the metadata table, allowing developers to select specific datasets for import. Selected datasets are then displayed in a configuration table within the Inspector, where developers can adjust visibility, assign colliders (used both for enabling user interaction – e.g., displaying information or triggering actions upon selection – and for supporting realistic navigation by representing physical elements such as floors or walls), and bind predefined prefab templates to each dataset (Figure 3). A search radius parameter is defined by the developer to control dynamic loading based on user location within the 3D environment. Once the configuration is complete and saved, the system can dynamically construct the 3D scene. During runtime, the player’s position is continuously tracked and translated into coordinates consistent with the project CRS. The Unity client then requests, via the Web API, the identifiers of all objects whose bounding boxes fall within the defined search radius. These identifiers are used to fetch corresponding binary 3D model data from the BrokerDB, which are subsequently rendered in the Unity environment. Each visualized object stores its table name and unique identifier in an associated script, enabling efficient querying for additional information.

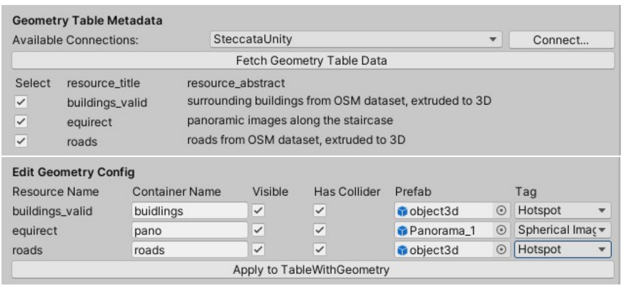


Figure 3. Geometric datasets selection on Unity Editor.

To further extend accessibility to heterogeneous database content, the system includes a customized query builder integrated into the Unity Editor (Figure 4). This visual interface allows developers – particularly non-experts – to construct SQL queries without writing code. The query builder interface allows developers to select relational tables across databases, configure field inclusion, specify aggregation and grouping options, and define table joins via an interactive graphical interface. The interface automatically generates the corresponding SQL query string based on these configurations and supports real-time query string and result preview, directly within the editor environment. For each selected field, the user may specify parameters fields such as alias, visibility status, and whether the field should be included in a results table or be presented as standalone content.
The system enables the user to save each constructed query along with its associated parameters to a JSON file. Each saved query includes the query string, primary table name, primary key column, join conditions, and per-field parameter settings. These saved queries are registered within the system’s configuration layer to allow efficient reuse without the need to manually reconstruct queries at runtime. Each saved query is associate with a geometric dataset selected in the previous operation.

During runtime, when a user selects an object in the 3D scene, the system identifies the associated table and unique identifier. It then locates the corresponding pre-saved query, dynamically injects a filtering condition into the WHERE clause using the object's primary key and executes the query. The returned results are structured according to the predefined configuration and rendered in the virtual environment. Depending on the parameter settings, each field in the result may be displayed either as standalone content – such as a 3D model, descriptive text, image gallery, or embedded PDF – or as part of a structured results table, as shown in the user interface. This flexible rendering strategy allows the system to present diverse data types in a coherent, modular, and context-appropriate manner across different object types and data sources while minimizing hardcoded logic and manual configuration.

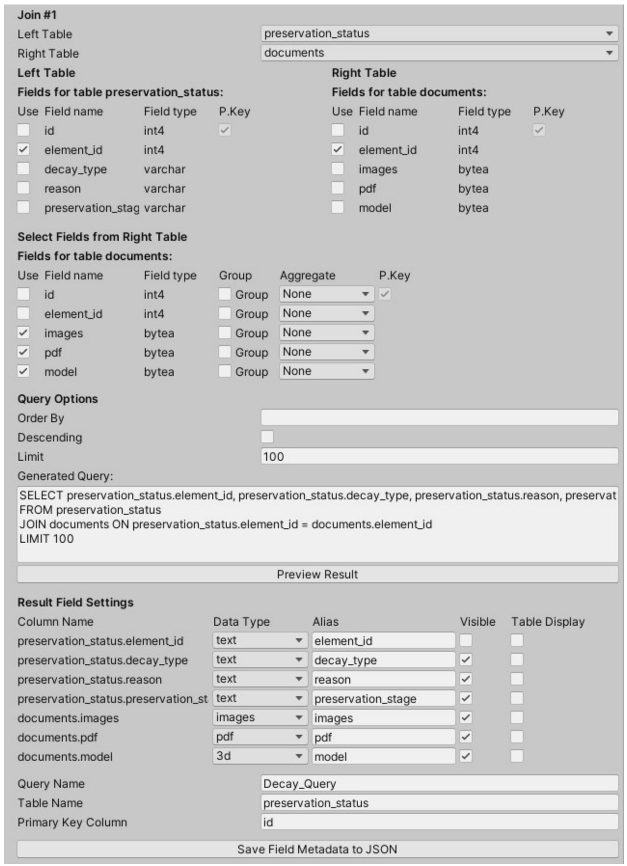


Figure 4. The query builder and result settings on Unity Editor.

## 3. Case Studies

### 3.1 Parma Cathedral

Parma Cathedral is a Romanesque building whose construction began in the latter half of the 11th century. During its lifetime, it has undergone numerous changes, including additions, damages, and restoration works that last until present day. Its architectural complexity, large scale, and continuous maintenance requirements make it a suitable case study for testing heritage data integration and management systems.

The datasets used for this study include a comprehensive survey conducted in 2017 (Bruno and Roncella, 2018), an HBIM model created in Autodesk Revit from those survey data, and a relational database developed to semantically enrich the model.
These resources have already been used for the development of an HBIM application, in which the information was stored in a

SQL Server relational database external to the BIM authoring software, allowing greater customization and flexibility while maintaining a real-time connection with the 3D model (Bruno and Roncella, 2019). This case is now revisited to assess how the new system interacts with pre-existing datasets (each with their own data-structure and logic) and to evaluate how efficiently it is possible to overcome one of the main limitations of the previous approach, in which interaction with the model geometry was constrained by the use of Autodesk Revit. The objective is to improve accessibility for a broader range of users and a broader range of client applications.

Survey data includes point clouds from terrestrial laser scanning (TLS) and close-range photogrammetry, co-registered through a topographic network georeferenced with Global Navigation Satellite System (GNSS) measurements. Photogrammetry was used to complement the TLS data, particularly for mapping material degradation on the cathedral's exterior facades. The HBIM model was created in Revit using both parametric and direct modeling to ensure accurate geometry and clear identification of elements requiring semantic data. The associated database integrates multidisciplinary information, including descriptive and textual information essential for planning conservation interventions – such as identified problems and anomalies, risk zones, interactions between adjacent elements, inspection methodologies, results of both qualitative and quantitative analyses, as well as documentation from previous surveys and restoration activities.

The point cloud, initially segmented by areas (naves, transept, and exterior), was down sampled to optimize performance within the Unity application and then imported into a dedicated table (*DuomoTLSPointClouds*) as binary point list, along with the CRS, centroid coordinates, and bounding box of each subset. A corresponding JSON file was created for each entry, describing the acquisition method, resolution, and instrument used. Metadata fields such as title, resource type, locator, temporal coverage, data type, and reference date were also added in compliance with INSPIRE guidelines. The photogrammetric mesh model was similarly imported into an additional table in the BrokerDB (*CRPmodels*), with spatial metadata and a descriptive JSON file. The *Metadata* table was updated accordingly. The HBIM model was imported through the specific Web API function that handles Revit models. Each element was converted to a mesh and stored in the *Revitmodels* table, with associated geometry, spatial metadata (centroid, bounding box, CRS), RevitID, etc. The SQL Server database used in the previous implementation was connected in read-only mode, preserving its original structure. Since its structure was based on the BIMelement entity, with a 1:1 relationship to Revit model elements via RevitID, all existing data could be linked directly to the corresponding 3D objects in the *Revitmodels* table within the new system.

Once the data were integrated, the Unity template was initialized and connected to the Web API. A query run on the *Metadata* table returned the available geometry datasets, which were displayed in the Unity editor. The BIM model and the point cloud were selected for visualization. The BIM model elements were assigned collider components, enabling selection and interaction. Query rules were then defined using the graphical query builder mentioned in section 2.5, allowing selection of which attributes to display per object type. For example, information on the conservation state of a column could be queried and shown directly in the application.

The application allows toggling between point cloud and Revit model visualizations. The point cloud provides more detailed geometric representation, especially for unmodeled elements, while semantic data is linked through the underlying Revit model (Figure 5).

Although the system requires some initial setup – for example, defining the queries for displaying information per element – this process is guided and flexible. The graphical interfaces developed both for the Web API and the Unity configuration templates allow more dynamic and customizable data management.

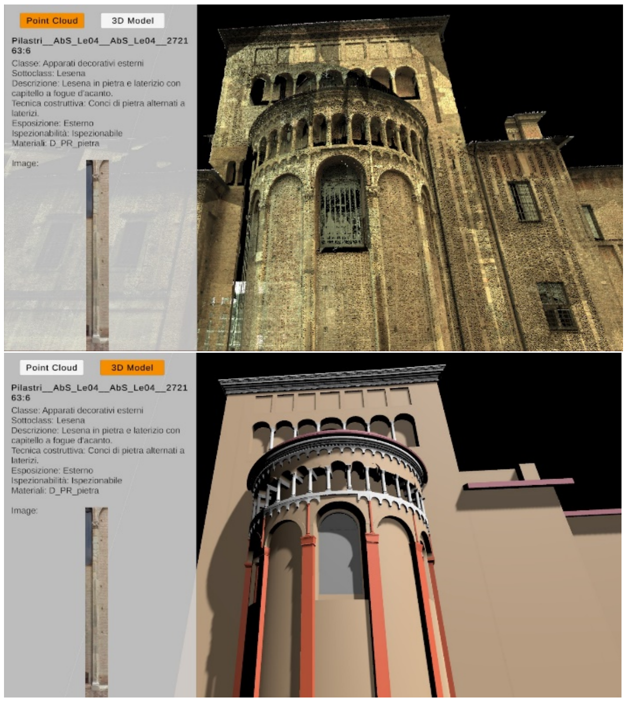

Figure 5. Example of information display in point cloud and Revit model visualization modes.

It is also straightforward to extend the system by adding new tables to the BrokerDB to associate additional information with model elements, or by connecting external databases, such as those containing sensors or monitoring data. Panoramic imagery, including Google Street View, can be co-registered for visualizing temporal changes. Surrounding urban context data (e.g., streets and buildings) can be integrated as extruded GIS layers to provide environmental context.

### 3.2 Basilica of Santa Maria della Steccata in Parma

Located in the historic centre of Parma, Santa Maria della Steccata is an important Marian sanctuary and one of the most significant examples of Renaissance architecture in the region. The site was chosen due to its cultural and historical relevance, as well as the complex preservation issues it poses. The ongoing need for restoration and conservation made it a suitable context to evaluate the system's capacity to handle architectural-scale data in a different scenario from the one adopted for the other case study.

The basilica was recently surveyed as part of a restoration planning project, which provided the opportunity to directly ingest newly acquired survey data into the system and evaluate the workflow for transforming them into an informative, queryable 3D model.

Survey data includes laser scanning of both the church's interior and exterior, Unmanned Aerial Vehicle (UAV) photogrammetry of all external facades and roofs, and close-range photogrammetry for accessible facade sections up to the first cornice level. For hardly accessible interior spaces, e.g. spiral staircases connecting the main floor to the attic, spherical photogrammetry was adopted. All datasets were co-registered to a topographic control network.

The laser scanner data were processed and imported following the same procedure described for the Parma Cathedral case. Unlike the Duomo case, no HBIM model was available. Consequently, semantic association was implemented through a combination of mesh segmentation and hotspot definition. Manual segmentation was carried out to isolate key architectural elements to which semantic content would be attached. These segmented mesh components were imported via the Web API into the *3Dmodels* table of the BrokerDB. Each entry was also documented in the *Metadata* table.

Through the Web API and its GUI, a set of information tables was created directly within the BrokerDB to store descriptive and conservation-related attributes. This process was carried out without connecting to external RDBMS: new tables and fields were defined through the interface, and data were inserted manually record by record (See Figure 2).

This case also provided an opportunity to test spherical image integration. Starting from the photogrammetric project in Metashape, panoramic images were generated and stored in the *steccata_pano* table of the BrokerDB. Each image was exported in three resolution levels to optimize data transfer and reduce rendering overhead when viewed far from the user's viewpoint. The images were oriented consistently with the coordinate system of the surrounding 3D model, eliminating the need for rotation at runtime.

The Unity project used the same base template as the Duomo case, validating the adaptability of the system to alternative datasets. Upon initialization, the metadata query returned the available geometry resources: the point cloud, segmented mesh models, and panoramic images. The point cloud was included for contextual visualization. The segmented meshes were equipped with collider components to enable selection and data querying. Queries were then defined via the Unity query builder to visualize information associated with each mesh segment (Figure 6).

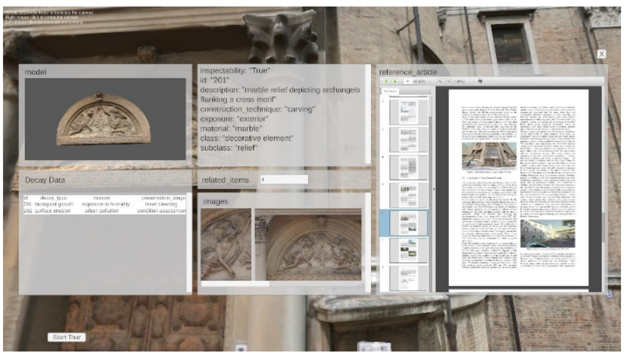

Figure 6. Example of information display for diverse data types.

For the panoramic images, as shown in Figure 7, each imported spherical image was mapped onto a textured sphere instantiated at its corresponding capture location within the 3D environment. When a user clicks one of these spheres, the associated equirectangular image is rendered as the skybox, ensuring consistent spatial alignment with the 3D context. Interaction with elements visible in the image is handled through underlying, invisible colliders on the corresponding 3D geometry. These colliders retain links to semantic content in the database. Upon interaction, a query is dynamically sent to the database via the Web API, and the relevant information is retrieved and displayed in context. This approach enables users to access and interact with semantically enriched 3D data within a panoramic-photo interface, offering an alternative to conventional methods that rely on image-based tags or annotations. In addition, the network of spherical image positions is used to define navigable waypoints, supporting the creation of a virtual tour for the exploration of normally inaccessible areas of the basilica, such as attic walkways and upper galleries.

Figure 7. Example of imported spherical images displayed as interactable objects in the scene.

## 4. Conclusion

The implementation of the proposed system in the presented case studies demonstrates its potential to support the integration, organization, and delivery of heterogeneous datasets in the context of cultural heritage documentation. The Parma Cathedral case study validated the interoperability of the workflow by integrating an existing HBIM project into the BrokerDB and visualizing its contents together with heterogeneous data from an external relational database in the Unity application. This process enabled the reuse of semantically enriched models beyond the limitations of application-specific environments. The case study of Santa Maria della Steccata illustrated the system's ability to corelate semantic information with spherical imagery, enabling the development of low-cost, high-fidelity virtual representations for architectural heritage.

These case studies confirm the feasibility of using a modular, data-driven framework to streamline the creation of interactive and informative cultural heritage applications. The system facilitates the reuse of existing documentation, lowers the technical threshold, and reduces the time investment typically required for immersive application development. However, several challenges remain in the phases of data integration, data handling, data description, and data accessibility, each suggesting directions for further refinement.

As far as data integration is concerned, future development could compliance with open standards such as Industry Foundation Classes (IFC) for building-level data and CityGML for geospatial urban-scale information. Moreover, the integration of live data streams—such as those generated by IoT devices or environmental monitoring systems—offers further opportunities for extending the system's functionality. These enhancements would support the development of Digital Twin applications, enabling real-time monitoring and preventive conservation for heritage sites.

With respect to data description and management, the current reliance on manual metadata updates introduces risks related to inconsistency and outdated references. Furthermore, while the existing metadata schema enables basic organization and filtering, it lacks semantic depth. To address this, the introduction of an ontology-based layer atop the BrokerDB is considered. This layer would facilitate the formal structuring of domain-specific knowledge and enable richer querying capabilities. By aligning with semantic web standards – such as CIDOC Conceptual Reference Model (CRM) – the ontology layer could provide a shared vocabulary that enhances cross-domain interoperability and improves data discoverability and contextualization.

In the experimentation so far, the modular structure of the Web API and the ability to organize information within the BrokerDB in a completely flexible manner (with the exception of a few system tables, such as those for managing connections to different datasets or for metadata related to external references) have enabled the rapid and effective development of several modules (such as the one for importing and/or automatically generating panoramic images from a Metashape project). It is expected that, in the same way, the newly proposed integrations will also be implemented in the future without the need to modify the overall structure and functioning of the system. Such experiments would also contribute to a more robust demonstration of data interoperability within the framework.

## Acknowledgements

Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5–Call for tender No. 3277 of 30/12/2021 of the Italian Ministry of University and Research funded by the European Union–NextGenerationEU. Project code ECS00000033, Concession Decree No. 1052 of 23/06/2022 adopted by the Italian Ministry of University and Research, No. CUP D93C22000460001, "Ecosystem for Sustainable Transition in Emilia-Romagna" (Ecosister), Spoke 4.

## References

Banfi, F., Brumana, R., Stanga, C., 2019. Extended reality and informative models for the architectural heritage: From scan-to-BIM process to virtual and augmented reality. Virtual Archaeology Review 10, 14–30.

Bitelli, G., Gatta, G., Guccini, A.M., Zaffagnini, A., 2019. GIS and Geomatics for archive documentation of an architectural project: The case of the big Arc of entrance to the Vittorio Emanuele II Gallery of Milan, by Giuseppe Mengoni (1877). J Cult Herit 38, 204–212.

Boje, C., Guerriero, A., Kubicki, S., Rezgui, Y., 2020. Towards a semantic Construction Digital Twin: Directions for future research. Autom Constr 114, 103179.

Botín-Sanabria, D.M., Mihaita, S., Peimbert-García, R.E., Ramírez-Moreno, M.A., Ramírez-Mendoza, R.A., Lozoya-Santos, J. de J., 2022. Digital Twin Technology Challenges and Applications: A Comprehensive Review. Remote Sensing 2022, Vol. 14, Page 1335 14, 1335.

Bruno, N., Roncella, R., 2019. HBIM for Conservation: A New Proposal for Information Modeling. *Remote Sensing*, *11*(15), 1751.

Bruno, N., Roncella, R., 2018. A Restoration Oriented Hbim System for Cultural Heritage Documentation: the Case Study of Parma Cathedral. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII–2, 171–178.

Cursi, S., Martinelli, L., Paraciani, N., Calcerano, F., Gigliarelli, E., 2022. Linking external knowledge to heritage BIM. Autom Constr 141, 104444.

Ferdani, D., Fanini, B., Piccioli, M.C., Carboni, F., Vigliarolo, P., 2020. 3D reconstruction and validation of historical background for immersive VR applications and games: The case study of the Forum of Augustus in Rome. J Cult Herit 43, 129–143.

Lei, Y., Bruno, N., Roncella, R., 2024. Integrating 3D survey and open datasets for creating low-cost urban digital twins in VR. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-2/W8-2024, 295–302.

López, F. J., Lerones, P. M., Llamas, J., Gómez-García-Bermejo, J., & Zalama, E. (2018). A review of heritage building information modeling (H-BIM). *Multimodal technologies and interaction*, *2*(2), 21.

Lovell, L.J., Davies, R.J., Hunt, D.V.L., 2023. The Application of Historic Building Information Modelling (HBIM) to Cultural Heritage: A Review. Heritage 2023, Vol. 6, Pages 6691-6717 6, 6691–6717.

Penjor, T., Banihashemi, S., Hajirasouli, A., Golzad, H., 2024. Heritage building information modeling (HBIM) for heritage conservation: Framework of challenges, gaps, and existing limitations of HBIM. Digital Applications in Archaeology and Cultural Heritage, 35, e00366.

Pybus, C., Graham, K., Doherty, J., Arellano, N., Fai, S., 2019. New Realities for Canada's Parliament: a workflow for preparing Heritage BIM for Game Engines and Virtual Reality. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42, 945–952.

Quattrini, R., Pierdicca, R., Morbidoni, C., 2017. Knowledge-based data enrichment for HBIM: Exploring high-quality models using the semantic-web. J Cult Herit 28, 129–139.

Rodrigues, F., Teixeira, J., Matos, R., & Rodrigues, H. (2019). Development of a web application for historical building management through BIM technology. *Advances in Civil Engineering*, *2019*(1), 9872736.

Sánchez-Aparicio, L.J., Masciotta, M.G., García-Alvarez, J., Ramos, L.F., Oliveira, D. V., Martín-Jiménez, J.A., González-Aguilera, D., Monteiro, P., 2020. Web-GIS approach to preventive conservation of heritage buildings. Autom Constr 118, 103304.

Xia, H., Liu, Z., Efremochkina, M., Liu, X., & Lin, C. (2022). Study on city digital twin technologies for sustainable smart city design: A review and bibliometric analysis of geographic information system and building information modeling integration. *Sustainable Cities and Society*, *84*, 104009.

Yang, X., Grussenmeyer, P., Koehl, M., Macher, H., Murtiyoso, A., Landes, T., 2020. Review of built heritage modelling: Integration of HBIM and other information techniques. J Cult Herit 46, 350–360.