

D²EPC Building Digital Twin v1



Project Acronym: D^2EPC

Project Full Title: Next-generation Dynamic Digital EPCs for Enhanced Quality and User Awareness

Grant Agreement: 892984

Project Duration: 36 months (01/09/2020 – 31/08/2023)

DELIVERABLE 3.3

D^2EPC Building Digital Twin v1

Work Package: WP3 – Building digitalisation and inverse modelling for implementing next generation dEPCs (M5-M36)

Task: T3.3 Buildings digital twin for EPCs issuance

Document Status: Final

File Name: D^2EPC_D3.3_D^2EPC_Building_Twin_v1_CERTH

Due Date: 31.03.2022

Submission Date: 15.04.2022

Lead Beneficiary: CERTH

Dissemination Level

Public ☒

Confidential, only for members of the Consortium (including the Commission Services) ☐

Authors List

Leading Author(s)				
First Name		Last Name	Beneficiary	Contact e-mail
Nikos		Katsaros	CERTH	nkatsaros@iti.gr
Stavros		Koltsios	CERTH	skoltsios@iti.gr
Ioannis		Pastaltzidis	CERTH	gpastal@iti.gr
Nikos		Bouzianas	CERTH	nickbouzi@iti.gr
Co-Author(s)				
#	First Name	Last Name	Beneficiary	Contact e-mail
1	Vasilis	Soulios	IsZEB	v.soulios@iszeb.gr
2	Andrius	Jurelionis	KTU	andrius.jurelionis@ktu.lt
3	Darius	Pupeikis	KTU	darius.pupeikis@ktu.lt
4	Christos	Kontopoulos	GSH	c.kontopoulos@geosystems-hellas.gr
5	Christiana	Panteli	CLEO	cpanteli@cleopa.de
6	Gerfried	Cebrat	SEC	gerfried.cebrat@senercon.de
7	Mija	Sušnik	DMO	Mija@demobv.nl
8	Thanos	Kalamaris	HYP	t.kalamaris@hypertech.gr

Reviewers List

Reviewers			
First Name	Last Name	Beneficiary	Contact e-mail
Paris	Fokaides	FRC	eng.fp@frederick.ac.cy
Phoebe-Zoe	Georgali	FRC	res.gp@frederick.ac.cy
Oscar	Molins Teruel	SGS	oscar.molinsteruel@sgs.com

Version History

v	Author	Date	Brief Description
0.1	Nikos Katsaros, Stavros Koltsios	17.12.2021	Initial draft (ToC)
0.2	Nikos Katsaros, Stavros Koltsios	10.02.2022	Partners' Contributions collection
0.3	Nikos Katsaros, Stavros Koltsios, Ioannis Pastaltzidis, Nikos Bouzianas	18.03.2022	Draft version with initial content for all sections
0.5	Nikos Katsaros	29.03.2022	First draft
0.6	Nikos Katsaros	06.04.2022	Document ready for peer review
0.8	Nikos Katsaros	14.04.2022	Document updated based on peer review comments
1.0	Nikos Katsaros	15.04.2022	Final version ready for submission

Legal Disclaimer

The D²EPC project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 892984. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Climate, Infrastructure and Environment Executive Agency (CINEA) or the European Commission (EC). CINEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

© CERTH, 6th km Harilaou-Thermis, 57001, Greece. Copies of this publication – also of extracts thereof – may only be made with reference to the publisher.



Executive Summary

The current report is Deliverable 3.3, which concerns the first version of the D^2EPC BIM-based Building Digital Twin. The aim of this document is to provide a detailed description of the Digital Twin component under development, with emphasis on the designed architecture, sub-components functionalities, and utilized data schemas. The deliverable also outlines the actual implementation of the component up to M19 of the project, considering the current development status under a public dissemination level.

First, a thorough analysis of the evolution process of Building Digital Twins is carried out. The main focus is placed on the Building Information Model (BIM), which constitutes the foundation for developing Digital Twins for the built environment. The development roadmap for leveraging BIM towards a complete Building Digital Twin is presented.

The conceptual architecture of the designed solution is documented. Each sub-component, namely the BIM Parser, the Input Data Validator, and the Building Behavior Profiling, is described in terms of functionalities and interaction with each other. Additionally, the interfaces of the Digital Twin with external components, such as the D^2EPC Repository and the Added Value Services Suite, are outlined. The data model of the D^2EPC Repository as well as the exchanged information between components, is defined.

Finally, the work related to the actual software component implementation is described. Technical details regarding the sub-components operation and communication with each other are provided.

The second and final version of this deliverable, which is due by M36 of the project, is expected to present the complete design and implementation of the D^2EPC Digital Twin, in alignment with the finalized development of other interacting components of the D^2EPC solution.



Table of Contents

1	Introduction.....	9
1.1	Scope and objectives of the deliverable	9
1.2	Structure of the deliverable.....	9
1.3	Relation to Other Tasks and Deliverables	9
2	Overview of digital twin applications in the building sector	10
2.1	Digitalization of building design & management	10
2.2	Building Digital Twin Development Roadmap	10
2.2.1	Step 1: BIM as a starting point.....	10
2.2.2	Step 2: BIM Applications – dimensions	18
2.2.3	Step 3: Introduction of IoT technologies.....	20
2.2.4	Step 4: Leverage AI and ML prediction algorithms.....	22
2.2.5	Step 5: A complete Digital Twin.....	23
3	D^2EPC Building Digital Twin conceptual architecture & data models.....	25
3.1	Architectural views and sub-components’ designed operation	25
3.1.1	BIM Parser	26
3.1.2	Input Data Validator	28
3.1.3	Building Behavior Profiling	29
3.1.4	Interface with the D^2EPC Repository	30
3.1.5	Interfaces with other D^2EPC services	30
3.2	D^2EPC Data models	30
4	D^2EPC Building Digital Twin implementation	36
4.1	BIM parsing pipeline.....	36
4.2	Input data validation	39
4.3	Building Behavior Profiling	39
4.4	D^2EPC Repository interfacing	40
5	Conclusions.....	41
6	References.....	42
	ANNEX A: D^2EPC data model & repository schema.....	45
	Location.....	45
	Building systems.....	48



Zone systems.....	49
Space elements	52
Repository schema	58

List of Figures

Figure 1: Common Data Environment (CDE)	11
Figure 2: Building lifecycle stages.....	12
Figure 3: IFC four-layered schema	14
Figure 4: Entity inheritance in IFC	15
Figure 5: Detailed inheritance in IFC [10]	15
Figure 6: Attribute Inheritance in IFC	16
Figure 7: An example of a property set for an entity of IfcWall	16
Figure 8: Example of incorrect space boundary	17
Figure 9: Boundaries pointing to the same space.....	17
Figure 10: BIM dimensions	18
Figure 11: Building energy performance [23]	21
Figure 12: BDT development Roadmap.....	24
Figure 13: Building Digital Twin Functional Diagram, from D1.4	25
Figure 14: Digital Twin architecture implementation view	26
Figure 15: BIM Parser conceptual level architecture	27
Figure 16: Building level and zone level technical systems (Case study 1)	28
Figure 17: JSON file samples describing the building and its consisting parts	31
Figure 18: JSON file samples describing the building sensors/meters and the corresponding actual measurements	34
Figure 19: Parsed Elements	38
Figure 20: Parsing Pipeline.....	38



List of Acronyms and Abbreviations

Term	Description
AEC	Architecture, Engineering and Construction
ANN	Artificial Neural Network
BDT	Building Digital Twin
BIM	Building Information Modeling
CAD	Computer Assisted Design
CDE	Common Data Environment
CFD	Computational Fluid Dynamics
DBMS	Database Management System
DT	Digital Twin
DXF	Drawing Exchange Format
EPC	Energy Performance Certificate
FM	Facility Management
gbXML	Green Building Extensible Markup Language
IFC	Industry Foundation Class
IoT	Internet of Things
IML	Information Management Layer
LCA	Life Cycle Assessment
MEP	Mechanical, Electrical, Plumbing
ML	Machine Learning
NBIMS	National Building Information Modelling Standard
O&M	Operation and Maintenance
PtD	Prevention through Design
RES	Renewable Energy Systems
SVM	Support Vector Machine



1 Introduction

1.1 Scope and objectives of the deliverable

This report provides an initial description of the work under T3.3 Buildings digital twin for EPCs issuance. The BIM-based Digital Twin constitutes a major software component in the D^2EPC architecture, providing data to all other components and services. The integration of information-rich BIM into the D^2EPC solution through the component is mainly emphasized, regarding the data retrieval, validation, correlation, storage, and further exploitation, aiming to provide a solid calculation base not only for the asset-based and operational-based EPC methodologies, but also for the KPIs developed within WP2. On the other hand, the introduction of dynamic data flows i.e. meter/sensor actual collected data, and their association with static BIM data is analyzed and provides insights into the past, present and future building energy performance on the operational level and related to actual indoor conditions. The document encloses the analysis of the current status regarding BIM and building digital twins' applications, the design of the D^2EPC Digital Twin architecture and the actual implementation of the software component.

1.2 Structure of the deliverable

In accordance with the aforementioned aspects, the report is structured as follows:

- **Chapter 2** provides an overview of BIM applications, focusing on their evolvement throughout the years and up to modern digital twin applications.
- **Chapter 3** outlines the D^2EPC Digital Twin's architecture, with additional analysis of the functionality of each sub-component and the used data model.
- **Chapter 4** presents more details on actual software implementation.
- **Chapter 5** sums up the main achievements and conclusions of this deliverable and indicates the next steps for the subsequent deliverable.

1.3 Relation to Other Tasks and Deliverables

The deliverable, as mentioned, is an outcome of Task 3.3. The design of the Digital Twin architecture utilizes, as a starting point, the original scheme described in Deliverable 1.4 "D^2EPC Framework Architecture and specifications v1". The asset-based EPC calculation methodology has served as a guideline for the preliminary development of the BIM parser tool. Along with the operational-based EPC calculation methodology, both provide the basis for the design of the D^2EPC Repository and the corresponding data model. Additionally, Tasks 2.1 – 2.5 and the respectful deliverables provide useful feedback towards enhancing the processing of the BIM files and the extraction of the KPIs developed under WP2.



2 Overview of digital twin applications in the building sector

2.1 Digitalization of building design & management

The architecture, engineering and construction industry (AEC), following many other sectors, initiated the transition to the digital era quite early, when technology advancements allowed the extensive use of computers. A major step towards this was the replacement of paper designs and analog processes with digital, through computer-assisted design (CAD). Since then and over the years, a variety of software tools supporting CAD have been developed, each one upgrading different functionalities, such as moving from 2-dimensional to 3-dimensional design and increasing the complex modelling capabilities.

2.2 Building Digital Twin Development Roadmap

Even though the concept of Digital Twins (DT) is a well-established industry tool, its application in the building sector is at an initial stage. This section describes the steps that D²EPC follows to define and develop its Building Digital Twin tool. Each step is described in detail, aiming to provide a deeper understanding of the concept of BDTs, how they operate and why they are considered to bring enormous value to the building sector. The BDT development procedure is divided into five steps, which are strictly related to the maturity levels, as described in deliverable D1.3 “Aspects of Next-generation EPC’s definition v1”. The first step lays the foundation for the prescriptive maturity level with a thorough building documentation platform that leverages current practices used in the building design sector. In the second step, state-of-the-art analysis of the available building simulation and analysis techniques used for building constructions reveals the existing arsenal that could also be incorporated in the D²EPC approach, while it also indicates the gaps, errors, or mistakes that should be avoided in order to construct a solid implementation plan. The potential of DT starts to reveal in the third step, where the documentation of the building’s static information is joined with the data streams from the installed IoT devices in a digital environment that contains the proper relationships to create a coherent real-time building description model. From a DT maturity level point of view, the second and third steps are strictly related to the Informative and Predictive DT levels. The fourth step includes the use of Artificial Intelligence (AI) and Machine Learning (ML) algorithms in the BDT tool to equip the model with advanced analysing capabilities and elevate it to the Prescriptive level. Finally, in the fifth step, the building model results in a fully integrated DT with near real-time monitoring, analysing, and informative capabilities.

2.2.1 Step 1: BIM as a starting point

The foundations for a successful BDT implementation are set with the construction of a coherent and analytical data model that reassures the flawless exchange and storage of information in a central hub [1]. BIM literacy provides a well-established documentation environment in which application has been tested in the broader building construction and facility management sectors for the past decades. The adoption of the data structure used in BIM would not only drastically reduce the BDT implementation time but will also prevent the occurrence of future data incompatibilities [2]. In this chapter, a brief analysis is carried out regarding the reasons open BIM has been used for the development of BDT as well as its limitations that the D²EPC approach has to overcome.



Resulting from the evolution of CAD in the AEC industry, BIM has emerged as a highly important advancement in building design. According to the National Building Information Modeling Standard (NBIMS), provided by the National Institute of Building Sciences, the term stands for “the digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its life cycle from inception onwards.”

A key technology that differentiates the BIM-based design from earlier CAD is the so-called parametric modelling capability. It comprises the dynamic representation of objects by introducing parameters and rules that define their geometry but also other non-geometry-related attributes and features. Two major benefits are derived from this advancement:

- The added parameters create relational structures, which allow the objects to be connected. Therefore, any object-specific or contextual update automatically performs updates on other linked objects, resulting in a dynamic structure.
- User-defined objects and object parameters facilitate the exploitation from other applications, for analysis and interpretation throughout the building’s lifecycle

The backbone of BIM is the concept of a **Common Data Environment (CDE)**. Essentially, CDE is a digital environment that stores all domain-specific building partial models and documents, which are necessary for the coordination and execution of a construction project or facility management [3]. CDE enables the collaboration of the various building stakeholders in a centralized and secure way. The various building documentation is collected and disseminated to the building stakeholders without the need for one-to-one communication requests (Figure 1), reducing the time and complexity of the various processes. The centralization of data storage within the CDE reduces the risk of data redundancy and ensures the availability of up-to-date data at any time. Furthermore, CDE leads to a higher rate of reusability of information, simplifies the aggregation of model information, and simultaneously serves as a central archive for documentation. As a result, CDE serves as the foundation for a well-defined method of collaboration among all stakeholders.

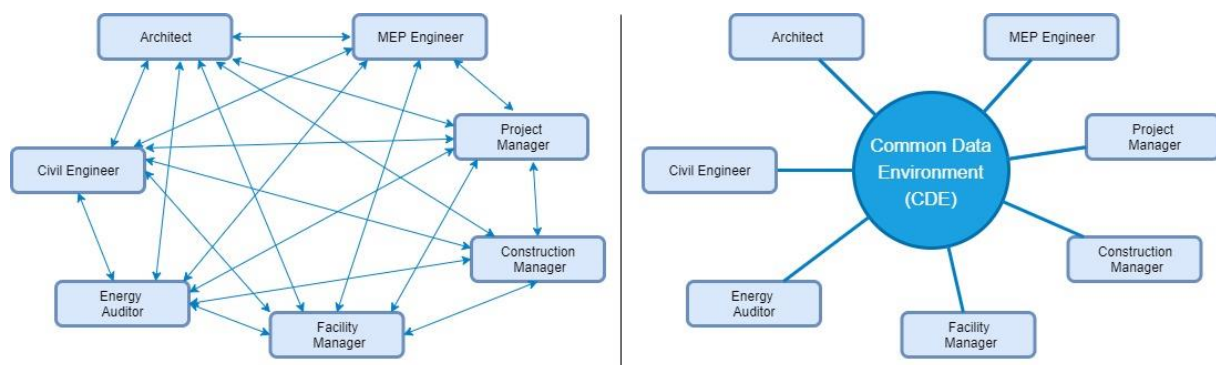


Figure 1: Common Data Environment (CDE)

The adoption of BIM in the last few years has established it as a building documentation option that can be used at all the stages of a building lifecycle, as presented in Figure 2. At the initial design stage of construction, BIM modelling can bring huge value by providing a common place for engineers, architects, or consultants to run various simulations and ensure the validity of the model in all the required domains. Proceeding in the construction phase, BIM modelling facilitates the scheduling and cost estimation, by providing an automated process that ensures the faster deduction of more efficient solutions with accurate results. During the Facility Management stage, or the Operation and Maintenance (O&M) stage, there are numerous BIM applications, namely: space management, energy management, indoor locating components for intervention, energy management, and occupational safety. BIM’s parametric intelligence and multidisciplinary adoption can contribute to the

development and operation of more human-centred buildings (in terms of human comfort, health, and safety) and at the same time reduce their environmental impact and building costs throughout their whole lifecycle [4].

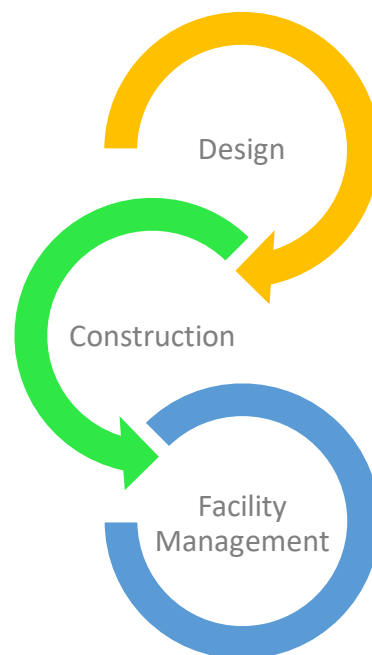


Figure 2: Building lifecycle stages

To facilitate the aforementioned process, BIM software design tools support predefined sets of object classes, which integrate fixed and dynamic attributes, along with rules to manage the behavior of the object instances that are substantiated from the classes. The latter may be modifiable or even extendable, in order to serve a variety of projects and application needs.

A vast number of BIM tools and platforms have emerged over the years, offering different capabilities and supporting various object classes, organized in digital libraries. Among the most prominent ones are Autodesk's Revit, Bentley Systems, Graphisoft's ArchiCAD, and others. The interoperability issue emerged from the existence of the numerous individual software solutions. As interoperability is referred to as the ability of technology to exchange information, interact, and collaborate with other systems or components without requiring major modifications [5]. Possible BIM interoperability with existing software modelling tools can lead to issues related to data loss, data recognition and mapping, unwanted generated data, and inconsistency of file extensions [2]. Furthermore, interoperability issues also arise between BIM and the various building IoT monitoring or Facility Management (FM) technologies, which result in information overload among systems during data extraction from BIM [6]. The solution to BIM interoperability is crucial for its future adoption in the building domain.

In past CAD systems (BIM Level 0 and 1), data exchange was mainly regarding the geometry models and realized based on common file formats, such as the Drawing Exchange Format (DXF), a proprietary exchange format introduced by Autodesk. However, the complex relations and added parameters that were established by the BIM design dictated that file formats also supporting additional information were necessary in order to progress. Additionally, it became evident that an open and publicly managed data schema and language is necessary for the communication of the various building information through different platforms.

To face the above-presented issues, a definition of open-standardization data formats is mandatory. The usage of suitable model information data, like Industry Foundation Classes (IFC) or Green Building



XML (gbXML) data formats, across BIM databases can give a bidirectional flow of information and address interoperability issues [7], [8]. Moreover, the development of API-based interfaces can safeguard the quality and quantity of building data through platforms and databases for improved integration and parameters assessment [9].

Between the two above-mentioned open data formats the IFC standard seems to prevail in the current market of building documentation and for this reason has been chosen for the development of the D²EPC documentation approach. The IFC open-source data schema was developed by buildingSMART¹. The schema, called Industry Foundation Classes (IFC), was firstly introduced in 1996 (IFC 1.0) and has since then received vigorous updates, increasing its interoperability capabilities. The two common IFC Specification versions that are currently in use are the IFC2x3 and IFC4. The conceptual schema for IFC is defined using the EXPRESS data specification language (ISO 10303-11). The data schema of IFC defines four conceptual layers and each schema is assigned to exactly one conceptual layer. These layers are:

- **Resource layer:** the lowest layer includes all individual schemas containing resource definitions. Those definitions do not include a globally unique identifier and shall not be used independently of a definition declared at a higher layer;
- **Core layer:** the next layer includes the kernel schema and the core extension schemas, containing the most general entity definitions. All entities defined at the core layer, or above carry a globally unique id and optionally owner and history information;
- **Interoperability layer:** this layer includes schemas containing entity definitions that are specific to a general product, process, or resource specialization used across several disciplines. Those definitions are typically utilized for inter-domain exchange and sharing of construction information;
- **Domain layer:** the highest layer includes schemas containing entity definitions that are specializations of products, processes, or resources specific to a certain discipline. Those definitions are typically utilized for intra-domain exchange and sharing of information.

Figure 3 represents the four-layered IFC schema.

¹ <https://www.buildingsmart.org/>



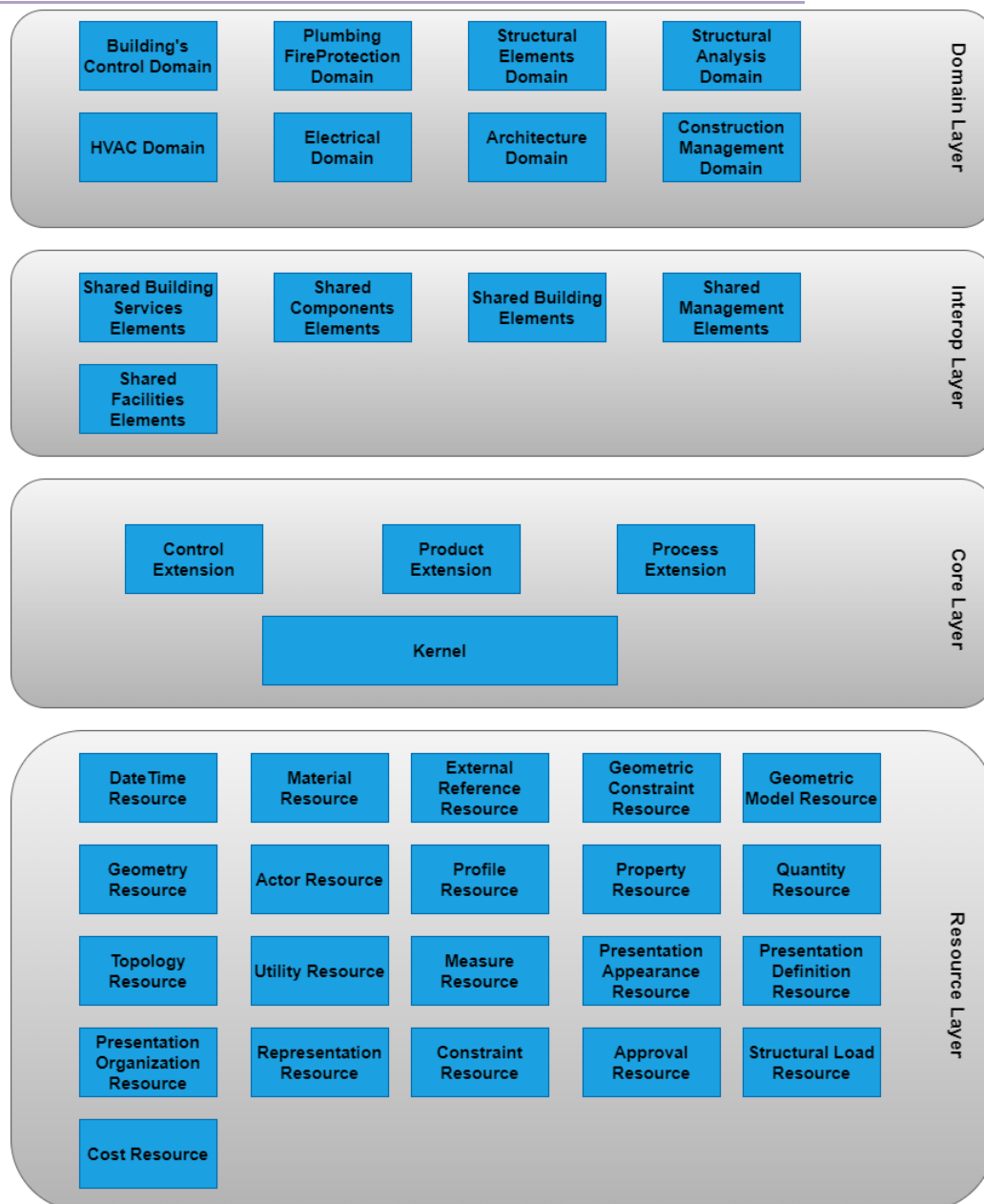


Figure 3: IFC four-layered schema

IFC elements follow an hierarchical structure, e.g. as shown in Figure 4, Figure 5 and Figure 6, the element *IfcWall* inherits the attributes of its parent classes. In Figure 5, a more detailed scheme for the entity inheritance is provided, containing the most common entities.



Entity inheritance

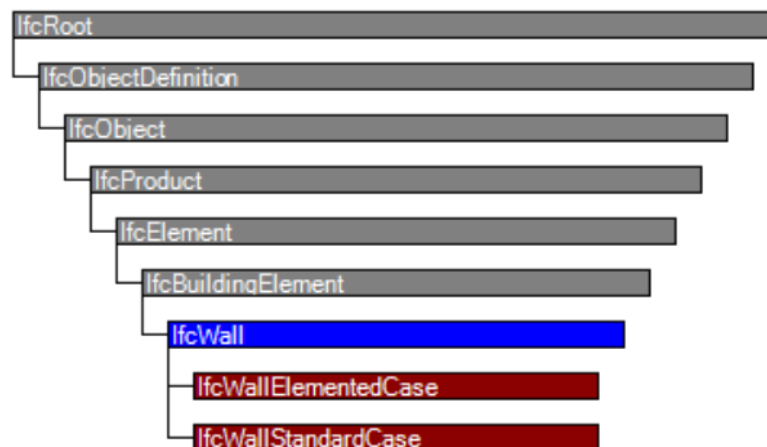


Figure 4: Entity inheritance in IFC

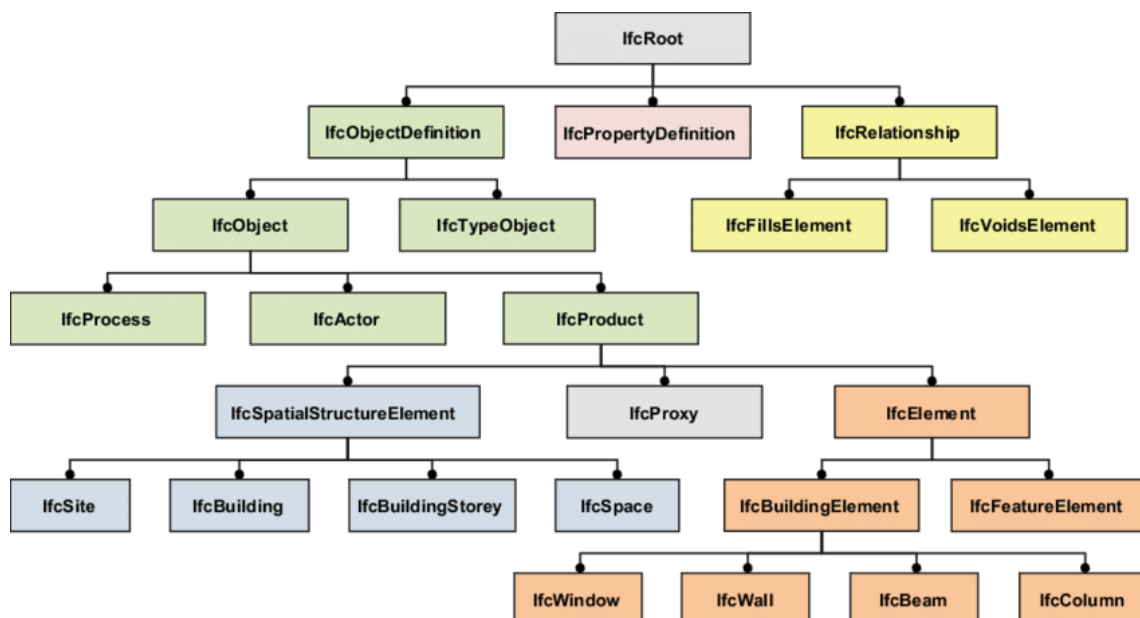


Figure 5: Detailed inheritance in IFC [10]

The attribute *OwnerHistory* for the element *IfcWall* is inherited from a higher entity (Figure 6). Properties can also be inherited if the property sets that they are a part of, have override functionality. An example of a property set for the *IfcWall* element is presented in Figure 7. It is worth noting that custom property sets can also be attributed to an *IfcObject*.

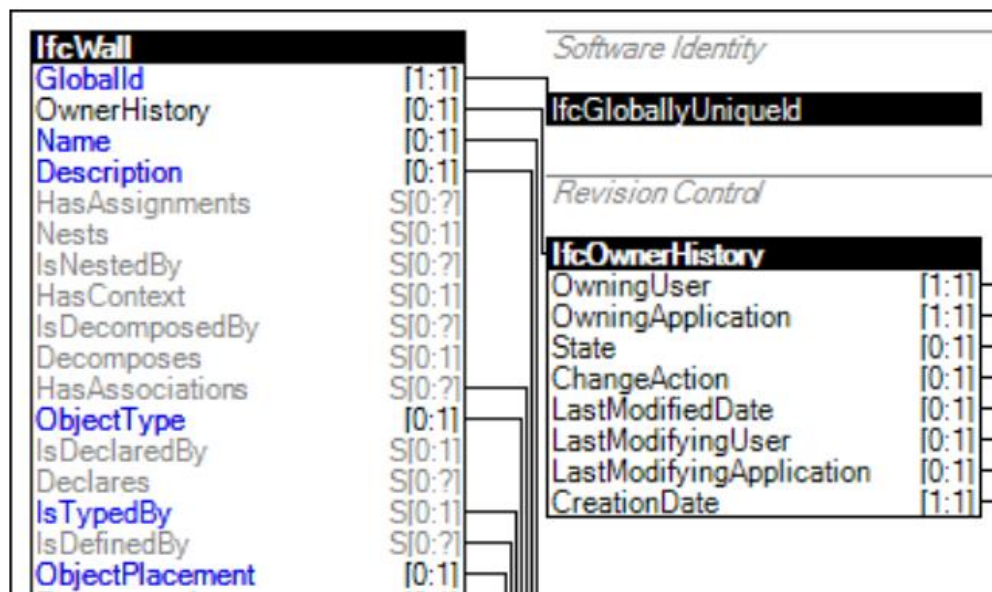


Figure 6: Attribute Inheritance in IFC

PsetName	Properties			
Pset_WallCommon	Template	PropertyName	Value	Reference
	Single Value	Reference	IfcIdentifier	
	Single Value	AcousticRating	IfcLabel	
	Single Value	FireRating	IfcLabel	
	Single Value	Combustible	IfcBoolean	
	Single Value	SurfaceSpreadOfFlame	IfcLabel	
	Single Value	ThermalTransmittance	IfcThermalTransmittanceMeasure	
	Single Value	IsExternal	IfcBoolean	
	Single Value	ExtendToStructure	IfcBoolean	
	Single Value	LoadBearing	IfcBoolean	
	Single Value	Compartmentation	IfcBoolean	
	Enumerated Value	Status	IfcLabel	

Figure 7: An example of a property set for an entity of IfcWall

In the properties of property set *Pset_WallCommon* in Figure 7, we observe that there is a “Value” index for each property. This index serves as a guideline to ensure that no matter which software is used to generate the BIM model, these properties will always be the same type. For example, an *IfcBoolean* property can have two values, “True” or “False”, while *IfcThermalTransmittanceMeasure* value will belong to the subset of real numbers as in (1):

$$\mathbb{R}_{>0} = \{x \in \mathbb{R} | x > 0\} \quad (1)$$

Exporting BIM models into IFC format using commercial software, such as Autodesk’s Revit, may lead to incorrect values in the IFC file. Another issue is that some elements can be attributed more than one time as a boundary to a space. Consequently, any tool that handles data in IFC should address this kind of issue appropriately. In Figure 8 and Figure 9, such a case is presented. This wall is defined as providing two boundaries in the IFC file, but upon further inspection, both boundaries are pointing to the same space.

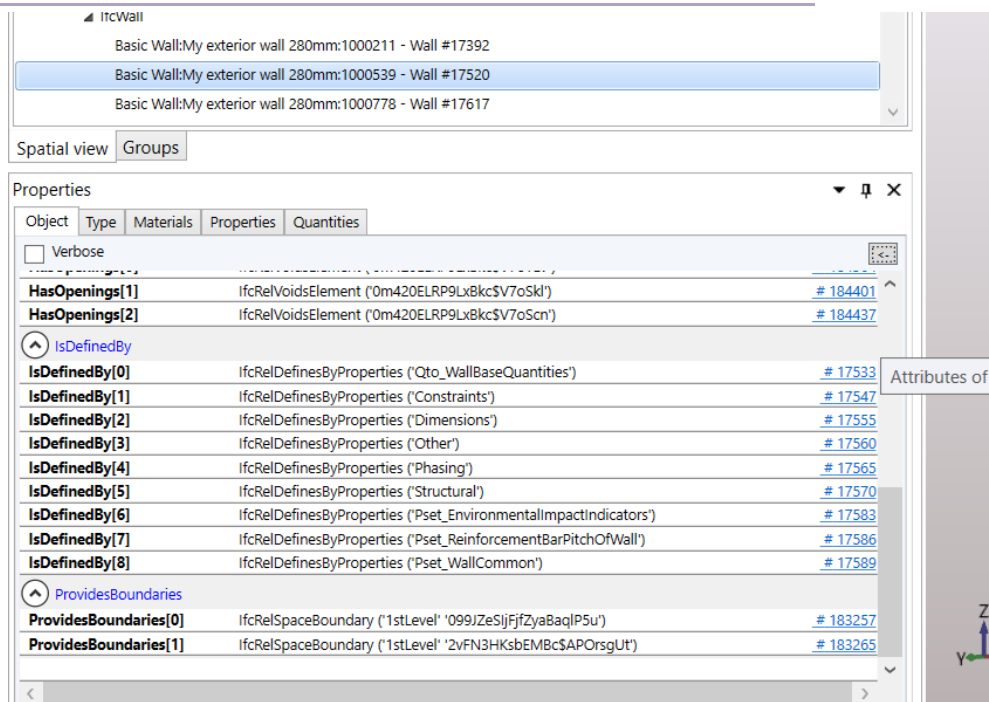


Figure 8: Example of incorrect space boundary

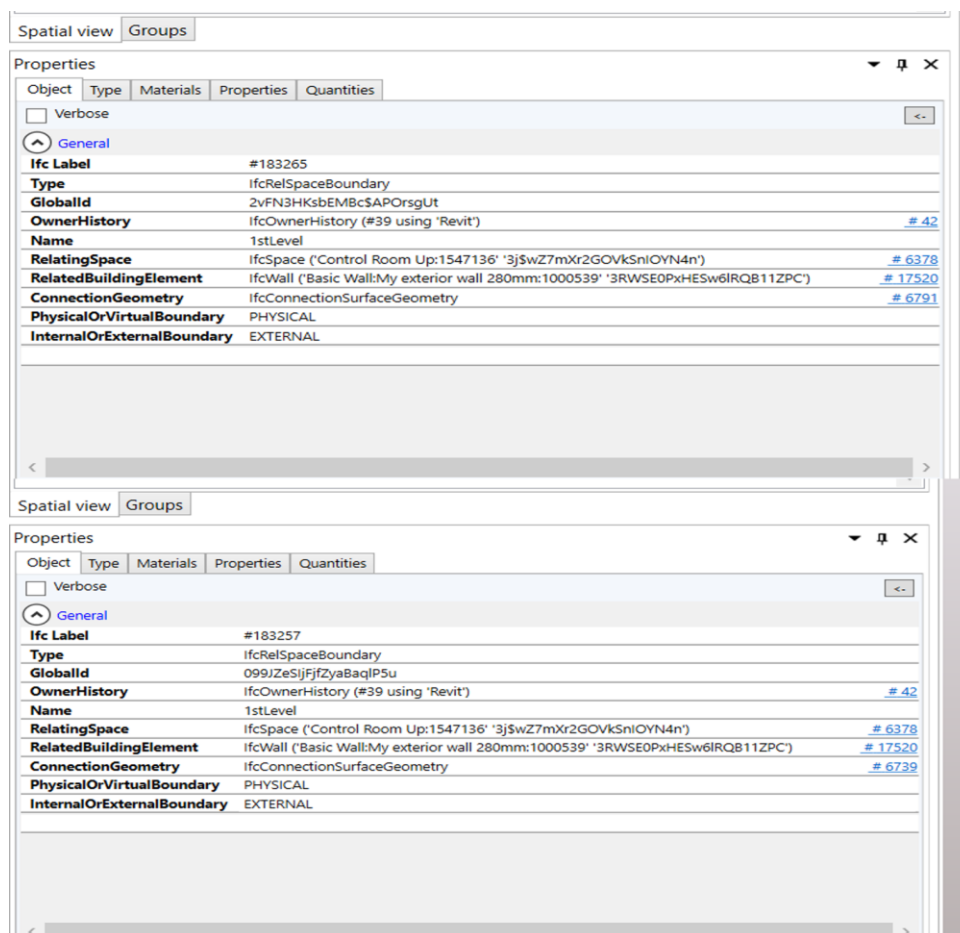


Figure 9: Boundaries pointing to the same space

In conclusion, BIM provides a potential environment for the documentation and dissemination of building static or dynamic information. Moreover, the 3D illustration of the building information provides a holistic view of different building stakeholders, at the various stages of the building's lifecycle. Nevertheless, its current deployment in closed market ecosystems prevents its wider adoption as a common documentation tool and prohibits its application for mass applications in the building sector.

2.2.2 Step 2: BIM Applications – dimensions

In the second step, the D²EPC approach aims to explore the utilization of BIM technology for multi-disciplinary building analysis and its simulation functionalities. The accurate geometric representation of the construction is semantically enriched with various types of information to serve the needs of each specific domain. According to the included information, the model expands to eight predefined dimensions [11], which are depicted in Figure 10. The first three dimensions refer to the already existing 3D geometric representation. The fourth dimension is related to the construction and scheduling of the project, while cost is considered the fifth dimension. As proceeding to the sixth dimension, several simulations in the energy domain aim to estimate the performance of the building. The seventh dimension examines the sustainability of the construction over its whole life cycle. Finally, the eighth dimension is related to health and safety issues. In the following paragraphs, the main characteristics of each domain are presented, with special emphasis on the aspects related to the D²EPC methodology.

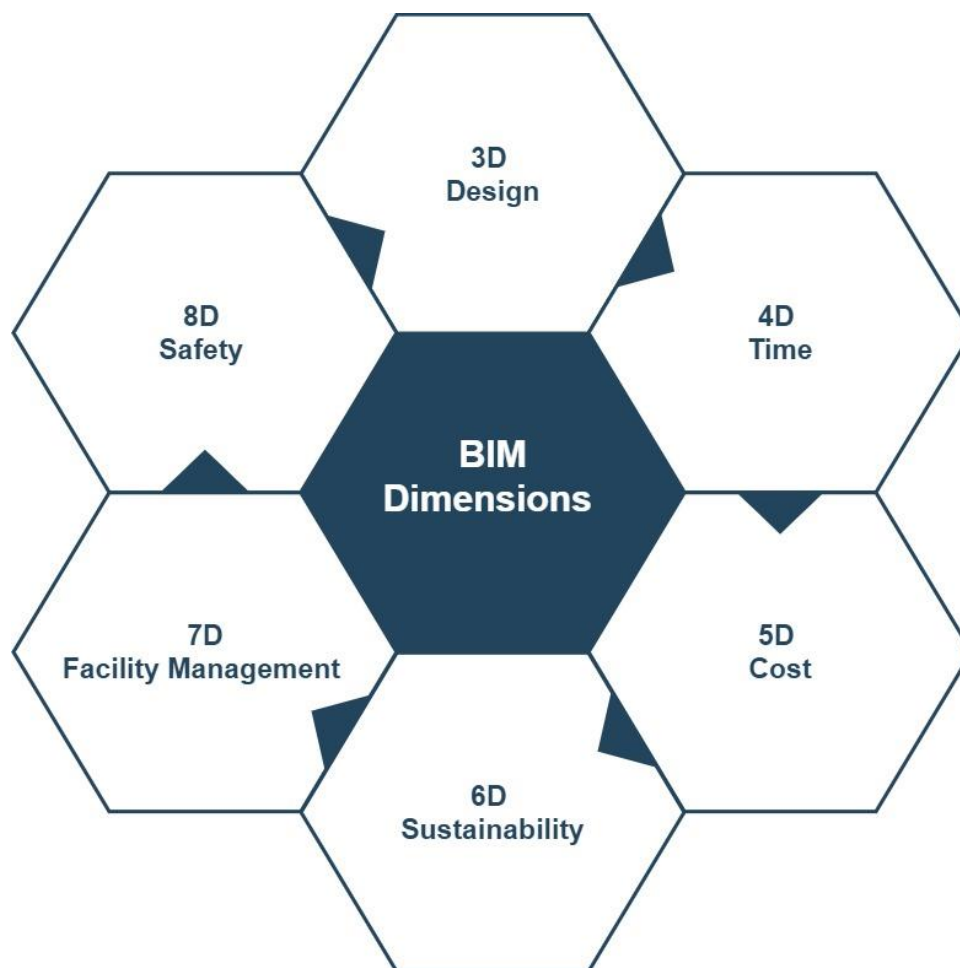


Figure 10: BIM dimensions

The construction industry has been one of the first sectors that leveraged the benefit of BIM to optimize the ongoing processes. The connection of 3D models with the construction activities lead to the visualization of the project development progress against time and offered the opportunity to assess the robustness and workflow planning of a project. In this essence, the dimension of “time” (4D) has been inserted into the BIM modelling process [12]. The development of automated scheduling algorithms gave project managers the opportunity to plan and evaluate various planning approaches taking into consideration a larger spectrum of parameters, thus reducing the construction time, material use, and labour hours. The adoption of automated construction planning approaches provides more efficient solutions with dynamic productivity plans and more accurate project estimation. In addition, the safety aspect of a construction project can be investigated with the utilization of dedicated algorithms that can reduce the potential faults and waste in the safety planning [13].

The fifth BIM dimension (5D) introduces the functions of cost estimation and budgeting to the model. 5D BIM is particularly useful for creating cost estimates in the early stages of a project, as well as for operational expenses later on when the project is in the execution phase [14]. The detailed cost information may contain various costs for equipment, material, and labor expenses. The objective is to correctly evaluate and anticipate the project's Return on Investment (ROI), with historical data playing an important role in this process. In addition, 5D BIM will improve the documentation of the asset's existing Bills of Material (BOMs) and Bills of Quantities (BOQs).

The sixth BIM dimension (6D BIM) is referred to as the sustainability phase and focuses on the success of making a building or other structure a self-sustainable, safe, and energy-efficient entity. BIM modeling can offer significant value at the early design stage of new construction, as the designer has the opportunity to run efficiently several simulation scenarios and find the optimal solution in a faster and more accurate way. Although the domain of energy performance has a broad spectrum of applications and numerous simulations types that can have to be conducted in order to gain a holistic view of the building's overall energy behavior.

The main calculation for the energy performance evaluation of a construction is related to its power and energy demand for maintaining the desired indoor comfort conditions. Advanced Computational Fluid Dynamics (CFD) calculation methods use BIM files as the basis to analyze the heat flow through the building elements and calculate the energy demand for heating and cooling under pre-defined conditions [15]. BIM literacy can be used as the basis for investigating the energy performance of construction, as it can describe in detail the characteristics both of the building fabric and its technical systems [16]. Regarding the building envelope, BIM files typically contain information about the utilized materials in construction, while it is relatively easy to extend the data model and include several information essential for the energy calculations. Similarly, the designer can define the installation of the technical systems, and how they are articulated (production unit, distribution system, terminal units), and enrich the model with their technical characteristics. The most important characteristic of the BIM schema is its expendability, which can be used to enrich the model with any required information according to the energy performance calculation algorithm.

The accurate energy performance evaluation of a building's construction requires the calculation of the building's exposure to solar radiation. Parameters like the building's location, orientation, and morphology of the surrounding environment, but also the construction of the building's envelope (both opaque elements and glazing) are taken into consideration to calculate the heat gains from the solar radiation. The same parameters are used to calculate the amounts of natural lighting that insert into the indoor environment and can be exploited to calculate the achieved indoor luminance levels during the day hours. This study in combination with analytical documentation of the lighting fixtures and their placement in the building spaces can be utilized to conduct automated lighting studies for examining the lighting comfort levels and energy consumption. At this point, it is important to mention that lighting is considered to have a major role in the energy consumption of larger buildings in the tertiary sector.



The utilization of BIM files as a basis for conducting simulations can be extended to several fields in the broader domain of building design, even if they are out of the scope of energy performance. For instance, acoustic simulation has utilized BIM models for analyzing and evaluating the design of auditoriums and theaters [12]. The BIM model can be enriched with additional custom parameters to express the acoustic qualities of materials (e.g., the absorption coefficient of each surface). The latter is inserted along with the geometry of the spaces in dedicated calculation modules to perform the acoustic simulations and analysis [4]. From the above, it is obvious that BIM models have a much broader application field and can serve any aspect of building design.

As proceeds from the design and construction phases to the building operation, BIM can be used by owners and facility managers for operational management and maintenance of the building and its components thought its life cycle [17]. 7D BIM is the dimension that tracks an asset's status, technical specifications, maintenance logistics, warranties, operation manuals, etc. Procedures like predictive maintenance or the replacement of parts are facilitated with the centralized documentation of 7D BIM. Furthermore, audits can leverage 7D BIM to ensure the building's efficiency, safety, and compliance with the national building standards at any stage of the building's life cycle. Lastly, the deployed monitoring system in the building can be used to optimize the resources and maintenance costs.

The eighth dimension (8D) is related to occupational safety and health in the construction industry and can also be benefited by the use of BIM technology. Processes like accident prevention or risk assessment of the selected components for a designed facility present difficult to implement so far, due to the limited designers' knowledge about safety during construction. The collaboration of BIM modeling with Prevention through Design (PtD) databases facilitates the automatic completion of hazard audits on BIM models and then provides suggestions for the replacement of critical building elements and for on-site hazards control [18]. Nevertheless, the potential of BIM for PtD is yet to be explored. The collaboration of BIM literacy with new emerging technologies can revolutionize the way designers understand and assess buildings, going a step further from just a basic simulation and examining the effects on human behavior and the surrounding environment. Such an application is currently examined in the domain of building safety. The synergy of BIM with virtual reality can generate several simulation scenarios to assess human behavior in several evacuation scenarios, such as in the case of fire [19]. The outcomes can be leveraged for advances in safety design, especially in the case of larger buildings.

All the above-analysed BIM applications examine various aspects of the building's construction from the early design stage, while they are also leveraged in the construction phase for management purposes. Despite the vast amount of added value that BIM modelling can offer to the sensing and construction phases of a building, its applicability has not been expanded yet in the everyday operation of a facility. The operation and Maintenance (O&M) stage consists of the most time-intensive stage of the building and has a greater impact on the building's life cycle. The synergy of BIM literacy with measuring and sensing equipment paves the way for holistic documentation of the building's life cycle, further elaborated in the following section.

2.2.3 Step 3: Introduction of IoT technologies

The large deployment of IoT devices in a major part of the newly-erected and existing constructions has the potential to introduce advanced monitoring capabilities into the future EPC schemes and pave the way for their transition into the "Industry 4.0" era. The closed monitoring of the asset's energy performance and indoor conditions in combination with the 3D depiction of the buildings' real-time status, can increase user's awareness and help them reduce their environmental impact while maintaining their desired comfort levels.



The majority of current IoT building applications include the fields of energy performance, indoor monitoring, human comfort monitoring, building safety, and public monitoring. In the first case, IoT devices provide energy consumption data, which are adjusted with the BIM model resulting in better energy monitoring and decision-making for building operations. Furthermore, the IoT integration to BIM provides information about the current state of the equipment with real-time tracking and identification of the components, thus it can be used for fault detection and predictive maintenance [[20], [21]. The domain of energy performance can also be examined from the public monitoring aspect. A variety of sensors is used for electrical power, thermal energy, occupants' heating behavior, and environmental data collection. Typically, all these data streams have JSON format and they are stored in No-SQL databases [22].

Concerning the indoor environment of a building, IoT integration with the BIM model can provide better comfort, health, safety, and indoor management. The collected data flows from sensors can refer to a variety of human-comfort parameters like temperature, humidity, or carbon dioxide concentration. The visualized presentation of these results in a BIM environment, for example with the use of colour codes (Figure 11), can provide significant feedback to the occupants and help them better manage their behavior and correlate it with their energy consumption profiles [23]. Moreover, BIM with IoT applications is also found in the field of building safety. Fire safety is one of the most typical examples, where the response in case of emergency is increased even more with the BIM visualization of data deriving from smoke, heat, and toxic gases detection devices [24]. Lastly, it is worth mentioning that various BIM integrated IoT applications are used in the construction sector, using technologies like RFID for better localization, quality, and cost management of the prefabricated components or ensuring the workers' on-site safety [25].

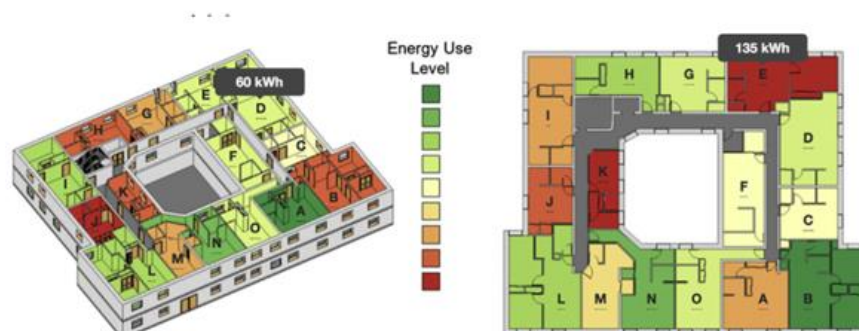


Figure 11: Building energy performance [23]

In the literature, there are various applications of IoT integration with BIM modelling. This paragraph presents information flow in a typical application for indoor monitoring. Firstly, there are various sensors deployed in the building spaces that measure humidity, temperature, light intensity, dust accumulation, and median particles. The data streams are accumulated with the use of microcontrollers (e.g., Arduino) and stored in different kinds of databases, like InfluxDB [26]. The next step is to correlate the BIM model with the accumulated data streams stored in the database, for better parameter mapping [27]. The use of visual programming scripts simplifies the data exchange between the BIM model and sensors by concerting data streams into JSON data formats [21]. Another approach for resulting data flow is the implementation of IFC open standards [28]. Finally, the data streams are presented to the end-users either integrated into a 3D visualization of the building or directly through the respective monitoring platform.

Despite the big potential of BIM cooperation with IoT, there are still certain challenges to be addressed before proceeding to large-scale applications. The first challenge is related to the discrepancy in the

quality specifications of each data protocol. For instance, each application (e.g., temperature measurement) has different measuring and data exporting requirements, which can create problems with the integration of data collection with another parameter. Thus, several problems can be transferred to the processes of data storage, retrieval, and finally to the integration with the BIM model. Furthermore, data acquisition failures are a risk due to the high dependence on internet connectivity, which is unstable and non-secure. Lastly, current BIM and IoT systems are characterized by slow response times, which prohibits several applications, e.g. in the case of an emergency that requires the urgent evacuation of the building.

To sum up, the collaboration of BIM modelling with IoT technology provides the opportunity to improve the monitoring of building performance. With IoT extension, BIM transforms into a dynamic representation of a building's lifecycle, which improves the quality of living of its occupants, reduces the carbon footprint, and allows a strategic schedule for better decision-making during maintenances. However, several technical challenges need to be taken into consideration during the BDT development process, to ensure its successful implementation and alignment with the present IoT market solutions.

2.2.4 Step 4: Leverage AI and ML prediction algorithms

All the above steps lead to the creation of a large pool with various building data deriving either from BIM, simulation results, or IoT data streams. The rise of AI computational capabilities enables the further processing of this enriched data set for forecasting and advanced decision-making processes. Leveraging this opportunity for advanced building performance analysis can lead to sustainable building management and raise occupant's energy awareness while minimizing environmental impact.

The major domains that AI models have been leveraged in the building sector are strictly related to the energy consumption of the building. Firstly, AI algorithms can be used to predict the various energy demand sources (e.g., heating, cooling, etc.) for better systems management [29]. The combination of ML tools and IoT controllers can lead to intelligent temperature regulation based on the occupant's temperature target and reduce building energy consumption [30]. The second type of AI algorithm related to the buildings' energy consumption examines the design of the construction and provides forecasting with retrofitting actions to result in the optimum design solution for minimizing its energy consumption [31], [32]. The utilization of AI will be necessary during the design and operational phases of energy-efficient constructions in the near future.

The domain of FM can also benefit from the adoption of AI algorithms. More specifically, the use of AI enables automated strategic decision-making procedures to reduce the maintenance cost or increase the service life of the buildings through preventive maintenance. AI models ensure accurate predictions for the building's energy performance and scheduled maintenance by training with historical data, testing predictions validations with testing data, and finally making future predictions. AI provides predictions for buildings' systems and mechanical, electrical, and plumbing (MEP) components at the preventive maintenance stage. Regarding systems, the AI model can predict faulty operation across all system parts [33]. Moreover, AI cooperation with IoT data supports the prediction of the operational condition of MEP components to avoid possible degradation and extend the lifetime of the component [34]. Finally, AI integration with the BIM tool and IoT can evaluate human thermal comfort through sensors measurements. In this case, AI models consider environmental, human state, and human change positions parameters, resulting in an optimal thermal environment [35].

There are two main ML methods used for complex building applications, which are using an efficient non-linear analysis for forecasting, namely Artificial Neural Network (ANN) or Support Vector Machine (SVM). ANN method provides high accuracy forecasting for building complex energy systems with rapid



execution times. SVM method provides high accuracy results with significant computational time. Both methods require historical data for their input.

Despite all dynamic services for building performance, AI models are facing some major challenges. Training of AI models requires a large amount of qualitative data to ensure the accuracy of their solution to complex problems. Furthermore, forecasting with traditional simulation methods is time-consuming due to the enormous amount of data. The developed AI models should be constantly updated with the last acquired data streams to ensure the accuracy of the system over time. With this data-intensive process with AI implementation arises the need for better data security and data management, like data filtering for enhanced quality assurance. Finally, the integration of AI models into the existing systems is a complex and time-consuming process that requires the expertise and experience of an AI solution provider.

2.2.5 Step 5: A complete Digital Twin

All the above steps can be summarized in a joint effort for the systematic documentation of the building's static and dynamic data, combined with the use of simulation and data analysis tools. Even though these steps are necessary for BDT development, they are not sufficient. A successful implementation strategy should include an overarching step to guide the overall process and ensure the sustainability of the resulted system. Faults and best practices from past DT development attempts, either in the building sector or in a broader set of domains, shall be taken into consideration to guide the implementation process. The main challenges that the DT development has to overcome are related to the collection and management of the numerous data streams. The data communication within the DT environment is characterized by large volumes, high transfer rates, and high diversity of information. Thus, the creation of DTs shall facilitate the efficient and secure exchange of data.

From the literature review, three main sets of data challenges have been identified for the successful development of a DT, namely: data models standardization, data synchronization, as well as, security and privacy [36]. Firstly, the data model standardization challenge; the crucial data required as input for monitoring and prediction algorithms are frequently spread across multiple systems that use different software platforms and database systems. Firstly, the variety of data protocols, formats, and processes create interoperability and data sharing problems, thus, data standardization is crucial to creating an interoperable data system for DT applications. The second challenge of data synchronization is related to the integration of data streams with different sampling characteristics. The discrepancies can be either in the time scale of the sampling or at the values of the measured parameters. A trade-off between synchronization cost and data quality has to be made in order to mitigate this challenge. Lastly, the increased risk of unauthorized access to personal (and confidential) information is critical throughout the lifecycle operation of a building. Thus, the establishment of access levels and permissions is essential from a data protection point of view [37]. To enable the preservation of personal data and privacy, the protection of business interests and intellectual property, and the minimization of dangers originating from data aggregation, digital twins must be secure by design.

A solid implementation plan is a prerequisite in order to develop a solid DT model that will face all the above-presented issues and, at the same time, incorporate the functionalities presented in steps 1 to 4. As shown in Figure 12, the systematic method for D²EPC BDT creation may be broken down into seven stages [38]. The first step is to determine the BDT model's overall objective. Secondly, this purpose may be further articulated using the expected applications and functionalities that will support the established set of goals. In the third phase, all the various data resources shall be validated and categorized to choose or create the data schema of the central digital model in step 4. The next step is to create logical and reliable data transmission networks, which will enable efficient data



communication between the physical and the digital world. Stage 7 adds data processing and analysis functionalities to the model by using the above-described simulation and machine learning methods. The constant testing of data systems during the building's operation is examined in stage 8, as it is crucial for ensuring the validity and quality of the BDT. Finally, the described procedure should be developed within an acceptable schedule and workflow process that anticipates any complications that may emerge.

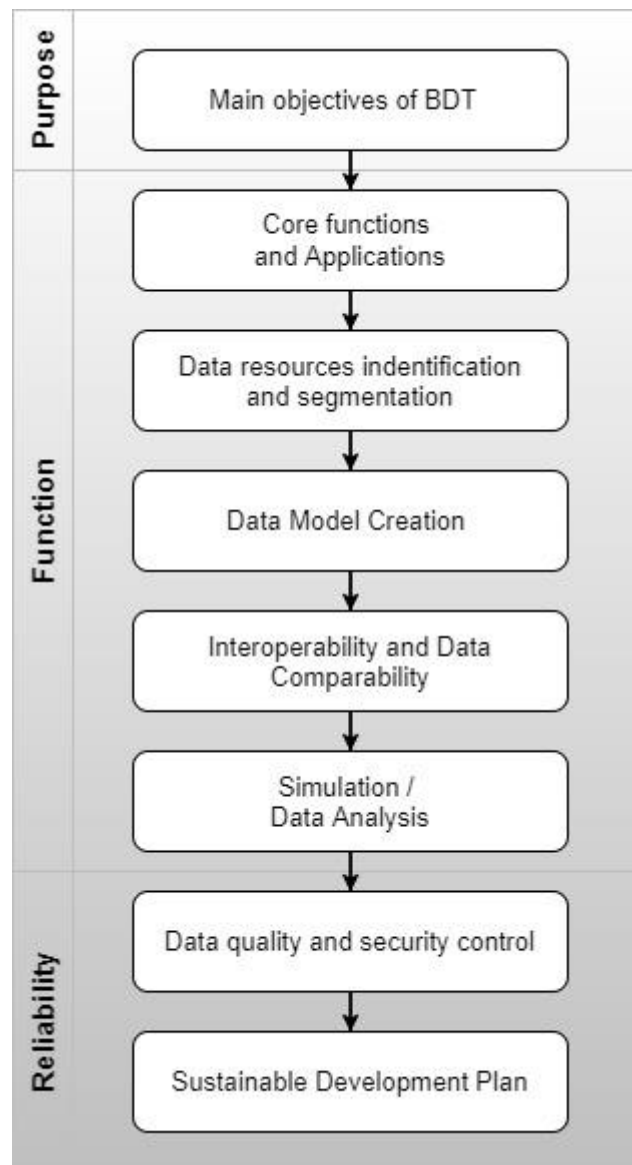


Figure 12: BDT development Roadmap

3 D²EPC Building Digital Twin conceptual architecture & data models

3.1 Architectural views and sub-components' designed operation

The functional view of the BIM-based Digital Twin architecture, as documented in the D1.4 “D²EPC Framework Architecture and Specifications v1”, is presented in Figure 13. It includes two core sub-components, namely the BIM Parser and the Building Behavior Profiling, while it is equipped with interfaces for other modules that require access to static and dynamic data.

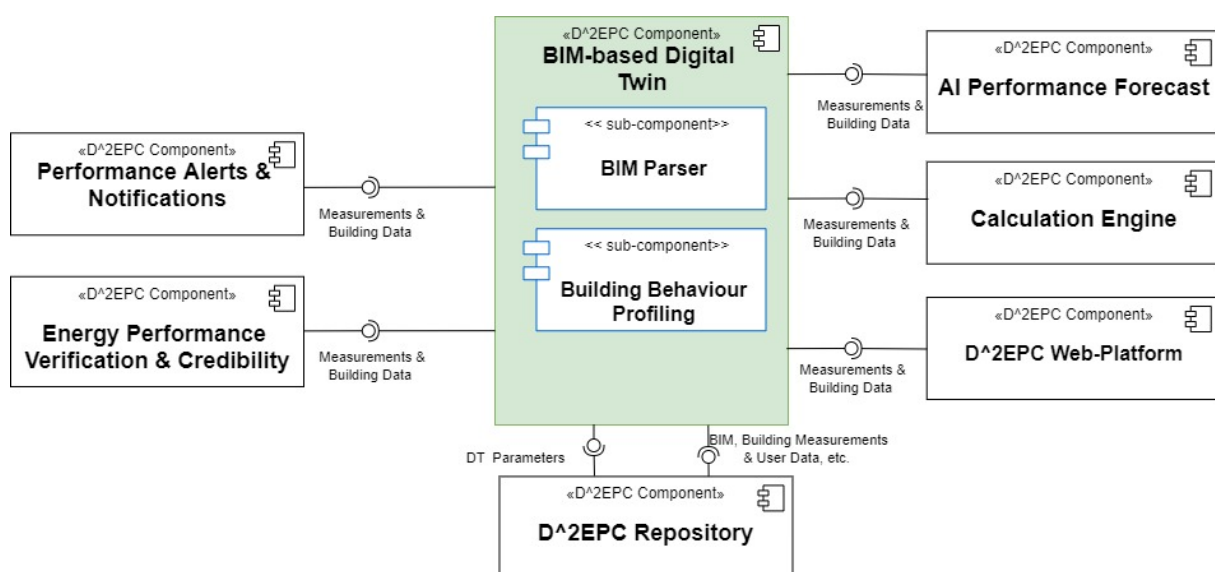


Figure 13: Building Digital Twin Functional Diagram, from D1.4

A more detailed view of the architecture is illustrated in Figure 14. The interactions between different modules, either feeding data to the Digital Twin or provided by it, have been included to facilitate the description of the entire process.

The main input data providers are the user, inserting static data through the WEB platform, and the Information Management Layer (IML), delivering close to real-time (dynamic) data, which is collected from sensors and meters installed in the actual building sites. Input from the user may be realized using two different formats. On the one side, a BIM file in IFC format can be uploaded to the platform and then utilized by the BIM Parser to extract any necessary data automatically. On the other side, any information, which is not or cannot be included in the IFC file, yet is necessary for the desired calculations, can be inserted manually through a proper web form. On the other side, any information that is not or cannot be included in the IFC file, yet necessary for the desired calculations can be inserted manually through a proper web form.

The communication with the external components, either the IML or the Web Platform, is realized through a single REST-API, which permits reading, modifying, creating and removing resource-related activities.

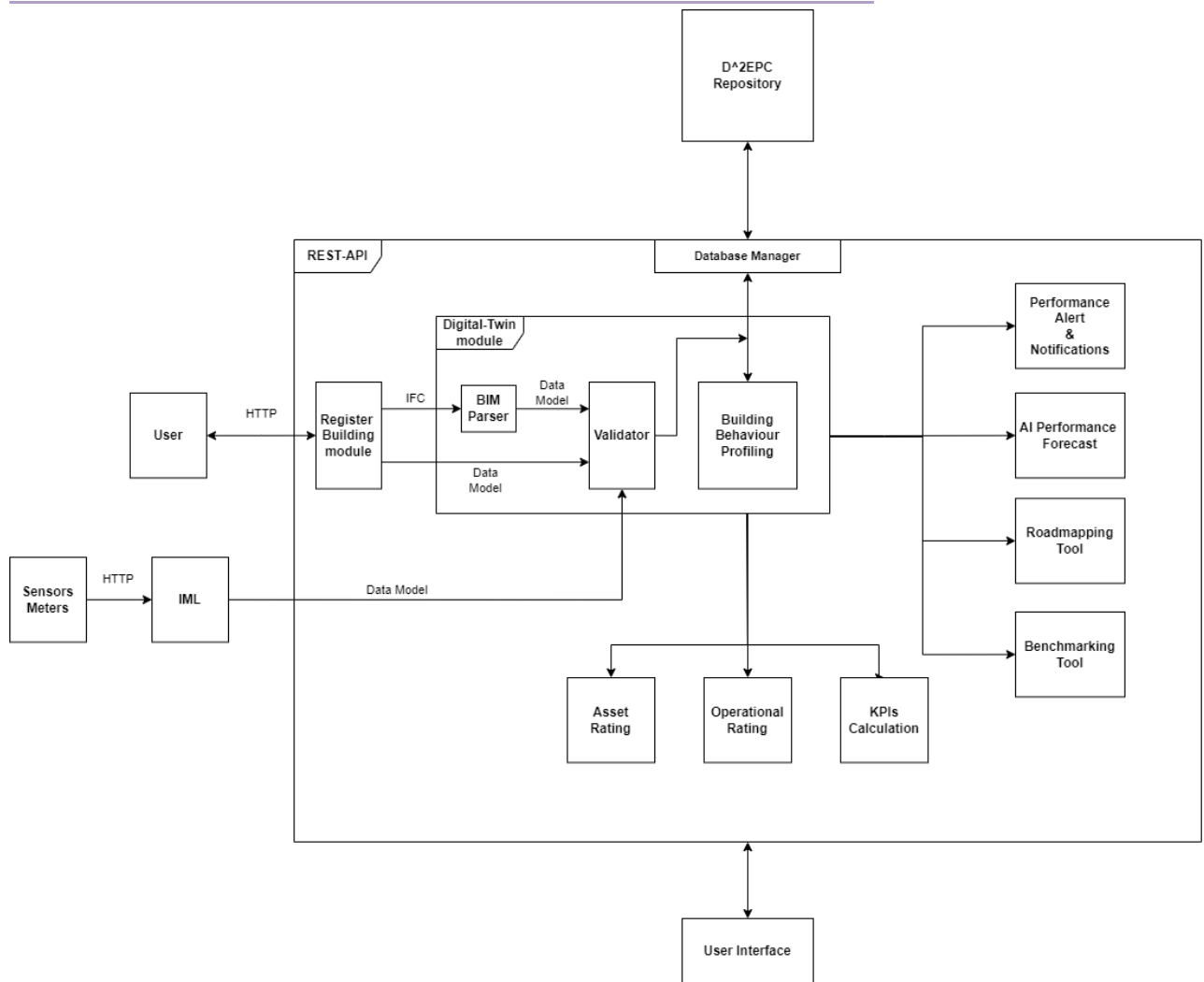


Figure 14: Digital Twin architecture implementation view

3.1.1 BIM Parser

A BIM Parser is being developed as a core component of the Digital Twin module, serving as a tool that enables the extraction of building information from an IFC file. In the current deliverable, an initial version of the parser is presented. For each file that is provided through the web platform, an output in JSON format is generated containing all relevant information of the corresponding building in the context of the D²EPC ecosystem, which is then forwarded to the validation tool.

In order to extract the entire information available in the IFC file, the parsing process is applied on four different levels (Figure 15), with each one containing:

- **First level:** The building, along with the information retrieved as a whole
- **Second level:** The building-level technical systems, such as the Renewable Energy Sources (RES), the thermal zones and the building's location
- **Third level:** The thermal zone usage information, the spaces that are part of the thermal zone, and the zone-level technical systems (heating, cooling, lighting)
- **Fourth level:** The elements of the spaces (opaque / transparent) along with the information that is needed for calculations from other components



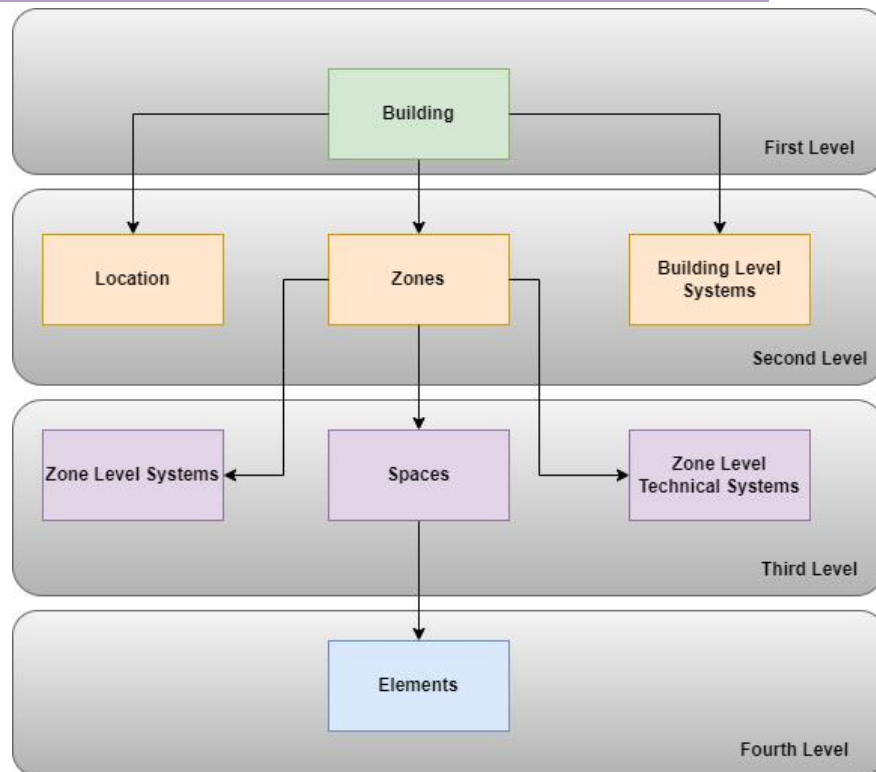


Figure 15: BIM Parser conceptual level architecture

Considering the aforementioned, the information utilized for the asset-based EPC rating calculation and extracted on each level can be further elaborated. At the second parsing level, the location entry comprises of the country and the municipal region. Building Level Systems are examined for various pieces of information for extraction based on their type; rated wattage of RES photovoltaic systems, slope and angle of the unit in solar collector systems for water heating, overall efficiency in Domestic Hot Water systems. Zone usage and other parameters (heated/ non-heated, construction type, total volume, and area) are extracted at the third level. Additionally, regarding zone-level technical systems for heating and cooling, the overall efficiency and the installation capacity are retrieved, along with the corresponding energy source, be it oil, electricity or natural gas. For zone-level lighting systems, the total wattage is extracted. Finally, at the fourth parsing level, where the building structural elements reside, two groups of them are present, the opaque and transparent elements. Both types include the net side area, the thermal transmittance, the orientation and the tilt, while the thermal transmittance and solar heat gain coefficient are specifically retrieved from opaque and transparent elements, respectively.

In Figure 16, sample output from the BIM file of Case Study 1 (CERTH's Smart Home DIH) is presented. The building has a photovoltaic installation comprising nine solar modules, each having a power output between 0.9 to 1.15 kW. Furthermore, it contains two boilers and a solar collector. The zone "offices" includes two heat pumps that provide necessary heating and cooling services. In Section 4.1, the parsing pipeline will be explained in detail.



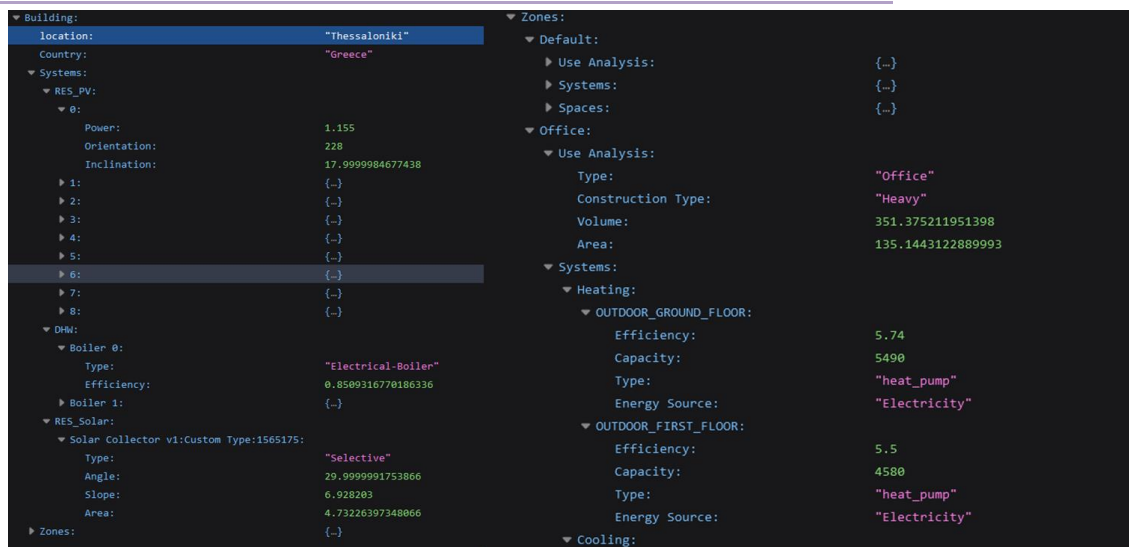


Figure 16: Building level and zone level technical systems (Case study 1)

3.1.2 Input Data Validator

As a second stage of input data handling, a component responsible for validation is designed, following the BIM Parser. Data validation is the process of confirming the correctness and quality of data. It is accomplished by including multiple checks into a system or report to assure the logical consistency of input and stored data. As automated systems receive data inputs with little to no human oversight, it is critical to guarantee that the data entering the system is valid and satisfies the specified quality criteria.

There are several forms of data validation. Most data validation methods will execute one or more tests to confirm that the data is accurate before saving it in the database. During the registration procedure, the validation component will run the following validation checks upon the input data:

- **Data Type verification** ensures that the data entered is of the right data type; for example, a field may only take numeric values. If this is not the case, the system should reject any data including additional characters such as letters or special symbols.
- **Code verification** confirms that a field is chosen from a valid set of values or that specific formatting standards are followed.
- **Range verification**, determines whether or not the input data is inside a preset range.
- **Uniqueness verification** validates that some data, such as IDs or physical addresses, are inherently unique. These fields in a database should most likely contain unique entries. A uniqueness check guarantees that an item is not stored into a database more than once.
- **Consistency verification** checks that data has been inserted in a logically consistent manner. For example, the telemetry values from a sensor must contain the quantity and the unit of each value.

Before constructing the data object from the given input (the output of the BIM parser), the input data validator is responsible for performing all of the essential data validation checks. The component verifies that the produced object specifies a building properly. This is a vital process that will also ensure appropriate data input to other components and services of the D²EPC solution, as the Digital Twin serves as the core component and main provider of information to the rest of the system.

Based on the description of the BIM parser in the previous section, examples of the component's validation actions performed on actual data inputs are as follows when registering a new building:

- If its location exists in the already registered locations/countries (**code verification**);
- If it contains zones and systems that also respect their expected data type, which is a list of items (**data type verification**);
- If its construction year is actually a valid year (**range verification**);
- If the building ID provided in the IFC file is unique (**uniqueness check**).

For all the above fields, the validator checks the type of every value prior to inserting it to the D^2EPC Repository (**data type check**).

If the validator detects that some fields retrieved from the BIM parser that do not meet the requirements based on the above validation checks, it generates a report containing all the fields that are ineligible to pass the validation. Subsequently, it informs the user of the changes that are necessary to be made to the input file in order for the validation to be completed without errors.

Another important feature of the component is the identification of any missing input data from the BIM Parser, which, however, is marked as required for the calculations of the asset-based or/and the operational-based EPC ratings, respectively. To this end, the input data validator provides a proper output that prompts the user to enter the necessary data manually through the web platform, which will provide the appropriate data input forms for better guidance. Evidently, the aforementioned validation checks will also be performed on user data inputs as well, to ensure that after the completion of the building registration process, the digital twin will have integrated all inputs into a single relational representation of the building, able to provide all necessary components and services with the required information.

Additionally, the input data validator receives input from the Information Management Layer, i.e. actual data measurements from the building under study. Prior to database insertion, the component is responsible for ensuring that the provided ID for each received measurement corresponds to an existing sensor entry, as defined from the static BIM data (**consistency verification**). This enables the effective integration of real data flows to the overall building digital representation and further exploitation by other components.

3.1.3 Building Behavior Profiling

This subcomponent is mainly responsible for managing the data feed from the Digital Twin to other D^2EPC components. Based on the linking of static, highly relational BIM data with actual data received from the building sensors/meters, as described in the previous section, it is able to identify common building behavioral patterns and provide insights in combination with other services.

The core functionality of the module is the mapping of each entire building entity from the D^2EPC Repository, including nested relations, to a class object, which serves as a copied instance of the original building. The main class includes not only attributes in accordance with the database schema, but also useful functions to easily access any level or even alter selected parameters, which allow other services to examine the building response to potential changes. This results in a dynamic building model that can support the energy performance simulation in various aspects. A representative example of such a process is required within the operation of the Roadmapping tool, which is planned to perform iterative simulations to yield the optimal building renovation actions, i.e. building parameters' alteration.

The utilization of the simulation capability mentioned above is user-driven, as the services that interface the digital twin are on-demand, called through the web platform.



3.1.4 Interface with the D²EPC Repository

To enable interaction (storage/retrieval) with the building data, a database management system interface will be utilized for communication between the REST-API and the D²EPC repository. A database management system (DBMS) stores data and responds to requests through a query language, such as SQL, while the DBMS interface facilitates the submission of queries to the database without the usage of the query language itself.

In the DBMS, each logical and physical object, such as a building, location, zone, systems, and so forth, will be represented by a database table. These tables provide all of the relevant fields for identifying the entity. Objects in the form of code classes will be utilized by the REST-API to interact with these tables. Each class will include some fundamental functions that will be utilized globally for all objects, as well as some private functions that will correlate single objects based on their functionality. As an example, the building model will include *set* and *get* functions for registering and retrieving the building's information, as well as specialized functions, e.g. *get_zones*, that retrieve the building's zones.

3.1.5 Interfaces with other D²EPC services

A new interface will be developed for each service to manage communication between the service and the user. Utilizing the Building Behavior Profiling subcomponent, each interface will be represented by a class object, and the attributes of the class object will contain the data that the service requires. The service can access the Digital Twin data either directly as a submodule or through the REST-API previously described.

3.2 D²EPC Data models

Considering the large amount of data and corresponding interrelations that the Digital Twin aims to handle, creating and utilizing a data model is necessary. A data model organizes elements of data, standardizes relationships among them and presents how they relate to the properties of real-world entities. It can represent not only logical and conceptual data but physical data as well. In software development, data models are often used as an aid to communication among a project's components and facilitate interoperability. Furthermore, via data models, it is feasible to illustrate the requirements of the information system and design proper responses to those requirements.

After evaluating D²EPC data flow needs, a data model was constructed in order to accommodate these needs for the following reasons:

- The Data Model provides a clear picture of data and can be used by the database developer to create a physical database that will be used to store all the information needed;
- It helps to define the relational tables, primary and foreign keys;
- It has a crucial role in the validation of data provided either from users via the web platform or the Information Management Layer (IML);
- Ensures that all data required for the various functionalities are accurately represented and described.

The data model construction was conducted in two phases. The first phase included the construction of the physical data model, which is inextricably linked with the construction of the database schema. A diagram software was used to build a schema presenting how the data will be physically stored within the database. As such, it offers a finalized design that can be implemented as a relational database, including tables that illustrate the relationships among entities and the primary and foreign keys that will be used to maintain those relationships. The D²EPC Repository schema included in Annex A presents the aforementioned physical data model.



The second phase of the data model construction included the creation of a set of JSON files to facilitate the data validation and data storing processes. JSON files come in a text file format that works with all programming languages while being readable from both humans and machines. The main idea is to keep a JSON file for each table presented in the schema mentioned above. The entity relationships will be maintained via primary and foreign keys in correspondence with the physical data model. Furthermore, within the JSON files, there is a description of each entity attribute and its datatype as well.

In the current phase of development, the JSON files will be utilized for feeding data to the data validation component and for storing data within the database. Firstly, the BIM parser will be providing its output in JSON format to the data validator. If there are no inconsistencies, the JSON input will be deserialized with proper post-processing to build data objects that will populate the D²EPC Repository. . On the other hand, if any errors or missing values are detected during the data validation process, proper messages will be sent to the web platform for users to know that they need to insert values for the missing or wrong data detected manually. In the remaining part of this section, selected JSON file samples are presented (Figure 17, Figure 18), while the rest can be found in the Annex. Modifications may be implemented during the future development stages, depending on any changes in data requirements.

Figure 17: JSON file samples describing the building and its consisting parts

```
{
  "building": {
    "building_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a building ID - Primary Key for Buildings Table"
    },
    "location_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a location ID - Foreign Key to Locations Table"
    },
    "use_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A string describing briefly the usage of a building"
    },
    "ownership": {
      "datatype": "string",
      "value": "A string value",
      "description": "Valuable information about building ownership"
    },
    "const_year": {
      "datatype": "integer",
      "value": "An integer value",
      "description": "The construction year of a building"
    }
  }
}
```

```
{
  "building_system": {
    "b_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A building system ID - Primary Key for Building Systems Table"
    },
    "building_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a building ID - Foreign Key for Buildings Table"
    }
  }
}
```




```
{
  "location": {
    "location_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "The location ID - Primary Key for Locations Table"
    },
    "latitude": {
      "datatype": "string",
      "value": "A string value",
      "description": "The latitude of the a location "
    },
    "longitude": {
      "datatype": "string",
      "value": "A string value",
      "description": "The longitude of the a location"
    },
    "region_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "The wider area of a location - Foreign Key for Regions Table"
    },
    "country_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "The country of a location"
    }
  }
}
```

```
{
  "thermal_zone": {
    "zone_id": {
      "datatype": "string",
      "value": "An string value",
      "description": "A unique string defining a thermal zone ID - Primary Key for Thermal Zones Table"
    },
    "building_id": {
      "datatype": "string",
      "value": "building_001",
      "description": "A unique string defining a building ID - Foreign Key for Buildings Table"
    },
    "zone_type_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "Description of a thermal zone - Foreign Key for Thermal Zone Requirements Table "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "m2",
      "description": "The area of a thermal zone measured in "
    },
    "height": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "m2",
      "description": "The height of a space(ex: room) that belongs to a thermal zone"
    },
    "volume": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "m3",
      "description": "The volume of a space(ex: room) that belongs to a thermal zone"
    }
  }
}
```




```
{
  "zone_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a zone system ID - Primary Key for Zone Systems Table"
    },
    "zone_id": {
      "datatype": "string",
      "value": "building_001",
      "description": "A unique string defining a zone ID - Foreign Key for Thermal Zones Table"
    }
  }
}
```

```
{
  "space": {
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Primary Key for Spaces Table"
    },
    "zone_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A string defining a building zone ID "
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": "The use of a space "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": "The area of a space "
    },
    "height": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m ",
      "description": "The height of a space "
    },
    "volume": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^3 ",
      "description": "The volume of a space "
    }
  }
}
```



```
{
  "space_element": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string space element ID - Primary Key for Space Elements Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string for defining space ID - Foreign Key for Spaces Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": " type has either opaque value or transparent value"
    },
    "contact": {
      "datatype": "string",
      "value": "A string value",
      "description": " A space element can be in contact with ground, external air or unconditioned zone "
    }
  }
}
```

The common ID correspondence between sensor/meters and measurement data, as described in Section 3.1.2, can be observed in the following model samples.

Figure 18: JSON file samples describing the building sensors/meters and the corresponding actual measurements

```
{
  "sensor": {
    "sensor_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a sensor ID - Primary Key for Sensors Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string for defining a space ID - Foreign Key for Spaces Table"
    },
    "measurement_id": {
      "datatype": "string",
      "value": "A string value",
      "description": " A unique string defining a measurement ID - Foreign Key for Measurements Table "
    }
  }
}
```

```
{
  "meter": {
    "meter_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string space element ID - Primary Key for Meters Table"
    },
    "measurement_id": {
      "datatype": "string",
      "value": "A string value",
      "description": " A unique string defining a measurement ID - Foreign Key for Measurements Table "
    },
    "building_id": {
      "datatype": "string",
      "value": "A string value",
      "description": " A unique string defining a building ID - Foreign Key for Buildings Table "
    }
  }
}
```



```
{
  "measurement": {
    "measurement_id": {
      "datatype": "string",
      "value": "A string value",
      "description": " A unique string defining a measurement ID - Foreign Key for Measurements Table "
    },
    "meas_type": {
      "datatype": "string",
      "value": "A string value",
      "description": " Describes what kind of energy is a measurement about "
    },
    "timestamp": {
      "datatype": "timestamp",
      "value": "A timestamp value",
      "description": " The timestamp of a Measurement "
    },
    "unit": {
      "datatype": "string",
      "value": "A string value",
      "description": " A unit of measure "
    }
  }
}
```



4 D²EPC Building Digital Twin implementation

4.1 BIM parsing pipeline

The BIM parser component's development is based on Python 3.8. The *IfcOpenShell*² library is used for the decomposition of IFC files, while the *Shapely*³ library for geometric calculations. The parsing pipeline is initiated by getting all the *IfcZone*-type entities in the file. For each instance of this type, the corresponding spaces that are contained inside that zone are retrieved, and, for each space, its elements (floor, roof, external walls, internal walls, doors, and windows) are parsed. *IfcOpenShell* methods on objects are aligned with the IFC schema, enabling their application without any deviation from it. The following code snippet illustrates the retrieval of thermal zones and their corresponding spaces from an IFC file by applying these methods.

```
zones = ifc_file.by_type("IfcZone")
related_spaces = zone.IsGroupedBy[0].RelatedObjects
```

To extract the building elements from each space, the first step is to extract the space boundaries using the *BoundedBy* method. Then, an identification of the element type (wall, slab, roof, etc.) can be performed.

```
boundaries = space.BoundedBy
if boundaries is not None:
    for boundary in boundaries:
        boundary = boundary.RelatedBuildingElement
        if boundary is not None and boundary.is_a() == "IfcWall":
            {do something}
        if boundary is not None and boundary.is_a() == "IfcSlab":
            {do something}
    ...
```

The parsing of windows and doors is conducted within the wall entities. If a wall includes doors or windows built into it in the IFC schema, those elements can be accessed via the *HasOpenings* method, as shown in the code snippet below. Since external walls can cover multiple spaces, a check is necessary to ensure that only the doors and windows that belong to the corresponding space are parsed.

```
openings = boundary.HasOpenings
if len(openings) > 0:
    for opening in openings:
        filling = opening.RelatedOpeningElement.HasFillings[0].RelatedBuildingElement
        if filling.is_a() == "IfcDoor" and (
            filling.ProvidesBoundaries[0].RelatingSpace == space):
            {do something}
        elif filling.is_a() == "IfcWindow" and (
            filling.ProvidesBoundaries[0].RelatingSpace == space):
            {do something}
```

In order to parse the technical systems that exist in a thermal zone, two methods have been developed. The first one links the space terminals with the external power source via a name property. If no property links a terminal to an external source, the latter is retrieved using the IFC schema methods. For example, if a terminal unit is assigned to a system, it is possible to search through the elements contained in that system using the *HasAssignments* and *RelatingGroup* methods. Moreover, by iterating through the elements in that group and taking advantage of the different predefined types for those elements, it is possible to parse the external power sources. In the case of HVACs, which are entities

² <http://ifcopenshell.org>

³ <https://github.com/shapely/shapely>



of *IfcUnitaryEquipment* in the IFC schema, the terminals have the predefined type of rooftop units, while the external sources are labeled as air-conditioning units.

The building elements are categorized into two groups; the opaque elements (walls, slabs, doors, etc.) and the transparent ones (windows). For each of these two groups, different properties are needed for the building's asset-based EPC calculation. In the case of opaque elements, the thermal transmittance and absorbance are extracted from the IFC file, along with their area, orientation, and tilt. Regarding the transparent elements, their thermal transmittance, area, orientation and tilt are parsed with the addition of the solar heat gain coefficient.

Two data dictionaries are created for each element, one from the Object properties and one from the Type Object properties, e.g. in the case of walls, the *IfcWall* and the *IfcWallType* are used, each one containing all the properties for that specific element. The properties that need to be extracted are then searched within these dictionaries. For thermal transmittance, three common naming conventions are considered, namely "*ThermalTransmittance*", "*Heat Transfer Coefficient (U)*" and "*U_value*". A similar procedure is conducted for the absorbance and the solar heat gain coefficient.

This route was followed since different software might apply different property names when extracting the BIM model in IFC format. If the thermal transmittance, absorbance, or solar heat gain coefficient are not defined within the IFC file, *None* value is returned from the parser, which will be later filled by the user/auditor.

Regarding the extraction of the elements' area, if the element is an internal wall, the area from the quantities is parsed, while if it is a window or door, the area is calculated from its width and height, as it was observed that some elements had incorrect set values.

A more complex procedure is followed for the parsing of the external walls. Firstly, based on the IFC schema, the wall connections are identified using the *ConnectTo* and *ConnectedFrom* methods, as shown in the following code sample.

```
connections=wall.ConnectedFrom

for connection in connections:

    relating_element=connection.RelatingElement
    related_element=connection.RelatedElement

    related_connection_type=connection.RelatedConnectionType
    relating_connection_type=connection.RelatingConnectionType

{retrieve global ids and coordinates}

connections=wall.ConnectedTo

for connection in connections:
    relating_element=connection.RelatedElement
    related_element=connection.RelatingElement

    relating_connection_type=connection.RelatingConnectionType
    related_connection_type=connection.RelatedConnectionType

{retrieve global ids and coordinates}
```

After extracting the global IDs and the coordinates where the wall is connected with other elements, only the coordinates from the elements that provide boundaries for the specific space are kept. By multiplying the distance between the points by the wall's height and subtracting the doors and windows built into the wall, it is possible to calculate the wall's area inside that space.

Regarding the calculation of the elements' orientation, the first step that has to be taken is to calculate the space's polygon. Thereupon, using the coordinates that were retrieved during the calculation of the wall's area, the vector that connects these points, $\vec{a} = (x, y)$ is calculated. Then, the perpendicular vectors to \vec{a} , $\vec{b}_1 = (y, -x)$ and $\vec{b}_2 = (-y, x)$ are identified, keeping the vector pointing outwards from the polygon, and using the angle from the True North, as defined in the IFC file, the element's



orientation is estimated. Doors and windows inherit this value. A sample output for the elements is displayed in Figure 19.

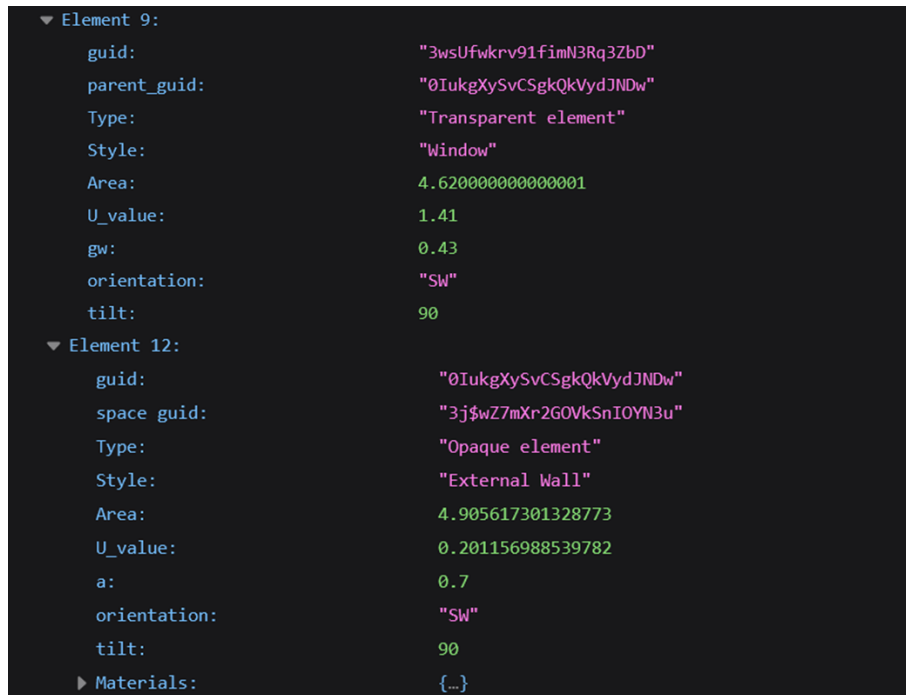


Figure 19: Parsed Elements

Throughout the entire process, several validation steps take place. As highlighted in Section 2.2.1, exporting BIM files to IFC format may come with caveats. It was observed that some elements can be attributed as space boundaries multiple times in the same space, which in turn would lead to incorrect output. To tackle such an issue, the global ID of an IFC element is used in order to ensure that if an element has already been added to the space that provides boundaries, other instances of this same element, if present, will not be taken into consideration.

After the completion of the main parsing phase, another validation check is performed. The internal walls that are not partitions between thermal zones are disregarded, whereas in case of an internal wall being a partition between a thermal and a non-thermal zone, it is labeled as such.

The overall parsing process is visualized in Figure 20.

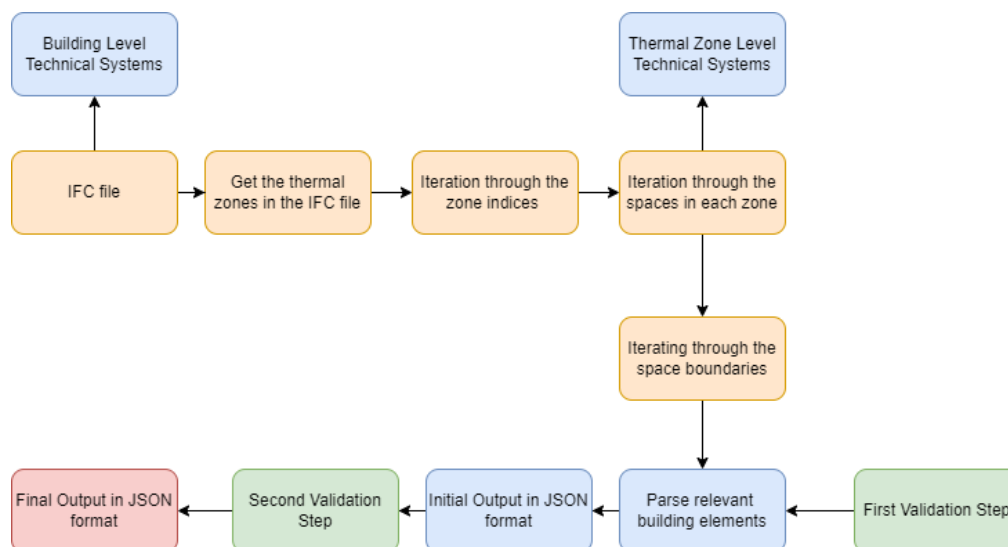


Figure 20: Parsing Pipeline

4.2 Input data validation

For the implementation of the validation tool, the Python programming language will be used, as it provides a variety of open-source libraries to develop the necessary functionalities. A prominent one is the Marshmallow⁴ library, commonly used for the development of validation tools, which receives raw data as input and delivers a data object as output, using serialization and deserialization methods. Marshmallow is an ORM/ODM/framework-independent package for converting complicated datatypes like objects to and from native Python datatypes.

Marshmallow schemas can be used to:

- **Validate** data input;
- **Deserialize** data and convert it to app-level objects that can be used by DBMS;
- **Serialize** app-level objects to Python basic types. After that, the serialized items are able to be translated to common formats like JSON and used in an HTTP API.

The validation procedures comprises two stages. The first step is to perform all of the verification checks on the raw data (BIM parser output). One of the Marshmallow library's functions will be used in this procedure. This function acts like a python decorator and is called when the raw data is deserialized. This method will search the raw data for certain fields that belong to classes, such as the building class, and extract the data before running it through a series of validation checks depending on their type. Their type is assigned during the DBMS initialization procedure and is determined by the data model presented in the previous sections. Each data model field type must match the corresponding one in DBMS. The validation tool will verify the type of each field against the DBMS given type to ensure that the data can be imported into the database.

During the second step, considering the validation is completed without errors, the tool deserializes the data into DBMS object data, which is then ready to be imported into the database. If the validation process fails, all errors are compiled into a report and sent to the user to manually enter all the fields missing from the IFC file.

4.3 Building Behavior Profiling

The Building Behavior Profiling subcomponent is represented by a Python class. Based on the provided building's unique identifier, it retrieves all relevant data from the D²EPC Repository, preserving their in-between interrelations. Additionally, a class function will be implemented for each external component that needs access to this data, having a bilateral functionality; data initialization and tool execution. All class variables are populated with the retrieved building's information, e.g. thermal zones, underlying systems and actual measurements, based on the requirements of the corresponding tool. For the deployment of the latter, two possible integration methods are foreseen; either its import as an independent Python module or access to it through the REST-API. In both cases, the required parameters will be provided, as mentioned above, to perform the requested tool calculations.

⁴ <https://github.com/marshmallow-code/marshmallow>



4.4 D²EPC Repository interfacing

The PostgreSQL⁵ is used as a database system, which is an open source relational DBMS. A database schema was created based on the requirements of the D²EPC project, included in Annex A. It comprises all of the tables that describe the logical object and physical entities, as well as their connections. Each table represents either a physical/real-world entity, such as a building, a location, the systems, the sensors, or logical entities that do not have a physical representation, such as thermal zones. Each table in the database includes attributes that describe every entity as well as properties that link the entities together. A building entity, for example, may include systems like Battery Storage or an EV charger.

Each of the systems' tables, as well as the building tables, have a unique ID per system and building respectively, where there is a relationship between them. This relationship is stored in the PostgreSQL.

Relationship patterns may be divided into four categories:

- One to Many, where a foreign key is used in each child table to refer to the parent. The parent's relationship () is thus described as referring to a collection of things represented by the child. A *location* (parent), for example, is related to a number of *buildings* (child), and the *location_id* is used, in the building table, to map the buildings to that specific location;
- Many to One, where, on the other hand, a foreign key is used in the parent table to refer to the child. The relationship () is declared on the parent, and a new scalar-holding property is established. A *sensor* (parent), for example, is related to a number of measurements (child), and the *measurement_id* is used, in the sensor table, to map the measurements to that specific sensors;
- One To One, which is fundamentally a bidirectional relationship. A *location* (parent), for example, is related to one conversion factors (child) configuration, and the *country_id* is used, in both tables, to map the location to the corresponding conversion factors;
- Many To Many is established by creating an association table with two classes and setting characteristics with their foreign keys. The secondary argument indicates that there is a connection (). Each connection () has a bidirectional relationship. A collection exists on both ends of the relationship. This relationship does not yet exist in the project's database schema, at the time of writing this document.

Each database table corresponds to a database management system (DBMS) model, which is represented by a Python class. Special functions are implemented in these classes to get data utilizing the above-mentioned relationships. This is achieved by identifying the relationship between the tables using the aforementioned unique IDs (foreign keys).

⁵ <https://www.postgresql.org/>



5 Conclusions

The first version of the deliverable D3.3 aimed to provide an initial overview of the D^2EPC BIM-based Digital Twin, both in terms of architecture design and actual component development up to M19 of the project.

A detailed review of the current state-of-the-art regarding Building Digital Twins was presented, concentrating on its BIM-based core and describing a development roadmap towards modern applications.

The deliverable specified the conceptual architecture of the component, illustrating its consisting submodules, their functionality and their internal cooperation. As the Digital Twin is considered an essential module, providing data to many other components and services, its interfaces with the latter are outlined. Additionally, a data model was created to define the information exchange among components, along with the retrieval and storage of data to the D^2EPC Repository.

Concluding, development-related technical details were provided, highlighting several implementation insights regarding the operation of each Digital Twin's sub-component and communication among them.

The second and final version of this deliverable, which is due by M36 of the project, will document the finalized design and implementation of the D^2EPC BIM-based Digital Twin tool, also based on the completion of the other linked D^2EPC components.



6 References

- [1] M. Deng, C. C. Menassa, and V. R. Kamat, "From BIM to digital twins: A systematic review of the evolution of intelligent building representations in the AEC-FM industry," *Journal of Information Technology in Construction*, vol. 26, pp. 58–83, 2021, doi: 10.36680/J.ITCON.2021.005.
- [2] E. Kamel and A. M. Memari, "Review of BIM's application in energy simulation: Tools, issues, and solutions," *Automation in Construction*, vol. 97. Elsevier B.V., pp. 164–180, Apr. 2019. doi: 10.1016/j.autcon.2018.11.008.
- [3] C. Preidel, A. Borrmann, H. Mattern, M. König, and S. E. Schapke, "Common data environment," *Building Information Modeling: Technology Foundations and Industry Practice*. Springer International Publishing, pp. 279–291, Apr. 2018. doi: 10.1007/978-3-319-92862-3_15.
- [4] K. din Wong and Q. Fan, "Building information modelling (BIM) for sustainable building design," *Facilities*, vol. 31, no. 3. pp. 138–157, Apr. 2013. doi: 10.1108/02632771311299412.
- [5] S. Shirowzhan, S. M. E. Sepasgozar, D. J. Edwards, H. Li, and C. Wang, "BIM compatibility and its differentiation with interoperability challenges as an innovation factor," *Automation in Construction*, vol. 112. Elsevier B.V., Apr. 2020. doi: 10.1016/j.autcon.2020.103086.
- [6] Paul Teicholz, *BIM for Facility Managers*. IFMA Foundation, 2013.
- [7] M. K. Ansah, X. Chen, H. Yang, L. Lu, and P. T. I. Lam, "A review and outlook for integrated BIM application in green building assessment," *Sustainable Cities and Society*, vol. 48. Elsevier Ltd, Apr. 2019. doi: 10.1016/j.scs.2019.101576.
- [8] X. Gao and P. Pishdad-Bozorgi, "BIM-enabled facilities operation and maintenance: A review," *Advanced Engineering Informatics*, vol. 39. Elsevier Ltd, pp. 227–247, Apr. 2019. doi: 10.1016/j.aei.2019.01.005.
- [9] M. K. Dixit, V. Venkatraj, M. Ostadalimakhmalbaf, F. Pariafsai, and S. Lavy, "Integration of facility management and building information modeling (BIM): A review of key issues and challenges," *Facilities*, vol. 37, no. 7–8, pp. 455–483, Apr. 2019, doi: 10.1108/F-03-2018-0043.
- [10] T. H. Kolbe and A. Donaubaue, "Semantic 3D City Modeling and BIM," *Urban Book Series*. Springer Science and Business Media Deutschland GmbH, pp. 609–636, 2021. doi: 10.1007/978-981-15-8983-6_34.
- [11] R. Charef, H. Alaka, and S. Emmitt, "Beyond the third dimension of BIM: A systematic review of literature and assessment of professional views," *Journal of Building Engineering*, vol. 19, pp. 242–257, Apr. 2018, doi: 10.1016/j.jobbe.2018.04.028.
- [12] B. Soust-Verdaguer, C. Llatas, and A. García-Martínez, "Critical review of bim-based LCA method to buildings," *Energy and Buildings*, vol. 136. Elsevier Ltd, pp. 110–120, Apr. 2017. doi: 10.1016/j.enbuild.2016.12.009.
- [13] H. Wang and X. Meng, "Transformation from IT-based knowledge management into BIM-supported knowledge management: A literature review," *Expert Systems with Applications*, vol. 121. Elsevier Ltd, pp. 170–187, Apr. 2019. doi: 10.1016/j.eswa.2018.12.017.
- [14] Leopard Project Controls, "INTRODUCTION TO BIM DIMENSIONS AND MORE SPECIFICALLY, 5D BIM." <https://consultleopard.com/5d-building-information-modeling-bim-and-cost-analysis/>
<https://consultleopard.com/5d-building-information-modeling-bim-and-cost-analysis/> (accessed Apr. 07, 2022).
- [15] H. H. L. Kwok, J. C. P. Cheng, A. T. Y. Li, J. C. K. Tong, and A. K. H. Lau, "Multi-zone indoor CFD under limited information: An approach coupling solar analysis and BIM for improved



- accuracy,” *Journal of Cleaner Production*, vol. 244, Jan. 2020, doi: 10.1016/j.jclepro.2019.118912.
- [16] Y. Yang, Y. Pan, F. Zeng, Z. Lin, and C. Li, “A gbXML Reconstruction Workflow and Tool Development to Improve the Geometric Interoperability between BIM and BEM,” *Buildings*, vol. 12, no. 2, p. 221, Feb. 2022, doi: 10.3390/buildings12020221.
- [17] M. Andreani, S. Bertagni, C. Biagini, and F. Mallo, “7D BIM for sustainability assessment in design processes: A case study of design of alternatives in severe climate and heavy use conditions,” *Architecture and Engineering*, vol. 4, no. 2, pp. 3–12, 2019, doi: 10.23968/2500-0055-2019-4-2-3-12.
- [18] I. Kamardeen, “Association of Researchers in Construction Management.” pp. 281–289, 2010.
- [19] H. Park, D. S. Panya, H. Goo, T. Kim, and J. Seo, “BIM-based Virtual Reality and Human Behavior Simulation For Safety Design.”
- [20] C. Z. Li, F. Xue, X. Li, J. Hong, and G. Q. Shen, “An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction,” *Automation in Construction*, vol. 89, pp. 146–161, Apr. 2018, doi: 10.1016/j.autcon.2018.01.001.
- [21] C. Han and H. Ye, “A Novel IoT-Cloud-BIM Based Intelligent Information Management System in Building Industrialization.”
- [22] R. Shipman and M. Gillott, “SCENE things: IoT-based monitoring of a community energy scheme,” *Future Cities and Environment*, vol. 5, no. 1, 2019, doi: 10.5334/fce.64.
- [23] A. Francisco, H. Truong, A. Khosrowpour, J. E. Taylor, and N. Mohammadi, “Occupant perceptions of building information model-based energy visualizations in eco-feedback systems,” *Applied Energy*, vol. 221, pp. 220–228, Apr. 2018, doi: 10.1016/j.apenergy.2018.03.132.
- [24] E. Preprint, R. Eftekhari, M. Nik-Bakht, and A. Hammad, “Extending IFC for Fire Emergency Real-Time Management Using Sensors and Occupant Information.” 2018.
- [25] Y. Fang, Y. K. Cho, S. Zhang, and E. Perez, “Case Study of BIM and Cloud-Enabled Real-Time RFID Indoor Localization for Construction Management Applications,” *Journal of Construction Engineering and Management*, vol. 142, no. 7, p. 5016003, Apr. 2016, doi: 10.1061/(asce)co.1943-7862.0001125.
- [26] K. Kang, J. Lin, and J. Zhang, “BIM- and IoT-based monitoring framework for building performance management,” *Journal of Structural Integrity and Maintenance*, vol. 3, no. 4, pp. 254–261, Apr. 2018, doi: 10.1080/24705314.2018.1536318.
- [27] R. Wehbe and I. Shahrour, “Use of BIM and Smart Monitoring for buildings’ Indoor Comfort Control,” doi: 10.1051/mateconf/20192.
- [28] I. I. Ignatov and P. N. Gade, “Data formatting and visualization of BIM and sensor data in building management systems. DATA FORMATING AND VISUALIZATION OF BIM AND SENSOR DATA IN BUILDING MANAGEMENT SYSTEMS.” [Online]. Available: <https://www.researchgate.net/publication/337274994>
- [29] Z. Hou and Z. Lian, “An application of support vector machines in cooling load prediction.”
- [30] J. Ahn, S. Cho, and D. H. Chung, “Analysis of energy and control efficiencies of fuzzy logic and artificial neural network technologies in the heating energy supply system responding to the changes of user demands,” *Applied Energy*, vol. 190, pp. 222–231, 2017, doi: 10.1016/j.apenergy.2016.12.155.



- [31] F. Ascione, N. Bianco, C. de Stasio, G. M. Mauro, and G. P. Vanoli, "Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach," *Energy*, vol. 118, pp. 999–1017, 2017, doi: 10.1016/j.energy.2016.10.126.
- [32] A. Alshibani and O. S. Alshamrani, "ANN/BIM-based model for predicting the energy cost of residential buildings in Saudi Arabia," *Journal of Taibah University for Science*, vol. 11, no. 6, pp. 1317–1329, Apr. 2017, doi: 10.1016/j.jtusci.2017.06.003.
- [33] S. Kalogirou, S. Lalot, G. Florides, and B. Desmet, "Development of a neural network-based fault diagnostic system for solar thermal applications," *Solar Energy*, vol. 82, no. 2, pp. 164–172, Apr. 2008, doi: 10.1016/j.solener.2007.06.010.
- [34] J. C. P. Cheng, W. Chen, K. Chen, and Q. Wang, "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms," *Automation in Construction*, vol. 112, Apr. 2020, doi: 10.1016/j.autcon.2020.103087.
- [35] G. Ma, Y. Liu, and S. Shang, "A building information model (BIM) and artificial neural network (ANN) based system for personal thermal comfort evaluation and energy efficient design of interior space," *Sustainability (Switzerland)*, vol. 11, no. 18, Apr. 2019, doi: 10.3390/su11184972.
- [36] N. I. Commission, "Data for the public good."
- [37] M. Shahzad, M. T. Shafiq, D. Douglas, and M. Kassem, "Digital Twins in Built Environments: An Investigation of the Characteristics, Applications, and Challenges," *Buildings*, vol. 12, no. 2, Apr. 2022, doi: 10.3390/buildings12020120.
- [38] Q. Lu *et al.*, "Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus," *Journal of Management in Engineering*, vol. 36, no. 3, p. 5020004, Apr. 2020, doi: 10.1061/(asce)me.1943-5479.0000763.



ANNEX A: D²EPC data model & repository schema

Location

```
{
  "region": {
    "region_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a region ID - Primary Key for Regions Table"
    },
    "cl_zone_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "The climatic zone ID - Foreign Key for Climatic Zones Table "
    }
  }
}
```

```
{
  "conversion_factor": {
    "factor_id": {
      "datatype": "integer",
      "value": "An integer value",
      "description": "The location ID - Primary Key for Conversion Factors Table"
    },
    "country_id ": {
      "datatype": "string",
      "value": "A string value",
      "description": "The country for which a conversion factor takes different values"
    },
    "energy_carrier": {
      "datatype": "string",
      "value": "A string value",
      "description": "The different types of energy, ex electricity "
    },
    "f_primary": {
      "datatype": "float",
      "value": "A float value",
      "description": "Primary energy conversion factor"
    },
    "f_co2": {
      "datatype": "float",
      "value": "A float value",
      "description": "CO2 emissions"
    },
    "cost": {
      "datatype": "float",
      "value": "A float value",
      "description": "cost of energy per kilowatt"
    }
  }
}
```



```
{
  "climatic_zone": {
    "cl_zone_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a climatic zone ID - Primary Key for Climatic Zones Table"
    },
    "temperature": {
      "datatype": "float",
      "value": {
        "value_01": "A float value",
        "value_02": "A float value",
        "value_03": "A float value",
        "value_04": "A float value",
        "value_05": "A float value",
        "value_06": "A float value",
        "value_07": "A float value",
        "value_08": "A float value",
        "value_09": "A float value",
        "value_10": "A float value",
        "value_11": "A float value",
        "value_12": "A float value"
      },
      "measuredIn": " Degrees Celsius",
      "description": " An array of ten temperature values "
    },
    "specific_pv_production": {
      "datatype": "float",
      "value": {
        "value_01": "A float value",
        "value_02": "A float value",
        "value_03": "A float value",
        "value_04": "A float value",
        "value_05": "A float value",
        "value_06": "A float value",
        "value_07": "A float value",
        "value_08": "A float value",
        "value_09": "A float value",
        "value_10": "A float value",
        "value_11": "A float value",
        "value_12": "A float value"
      },
      "measuredIn": "kWh/kWp",
      "description": " The annual energy production from a photovoltaic system per installed kWp "
    },
    "SR_H": {
      "datatype": "float",
      "value": {
        "value_01": "A float value",
        "value_02": "A float value",
        "value_03": "A float value",
        "value_04": "A float value",
        "value_05": "A float value",
        "value_06": "A float value",
        "value_07": "A float value",
        "value_08": "A float value",
        "value_09": "A float value",
        "value_10": "A float value",
        "value_11": "A float value",
        "value_12": "A float value"
      },
      "measuredIn": "kilowatt/m^2",
      "description": " Solar Radiation Horizontal"
    },
    ...
  }
}
```



```
{
  "thermal_zone_requirements": {
    "zone_type_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a zone type ID - Primary Key for Thermal Zone Requirements
Table"
    },
    "heat_period": {
      "datatype": "float",
      "value": "A float value",
      "description": "Operation hours during nighttime"
    },
    "cool_period": {
      "datatype": "boolean",
      "value": {
        "value_01": " 0 or 1",
        "value_02": "0 or 1",
        "value_03": "0 or 1",
        "value_04": "0 or 1",
        "value_05": "0 or 1",
        "value_06": "0 or 1",
        "value_07": "0 or 1",
        "value_08": "A0 or 1",
        "value_09": "0 or 1",
        "value_10": "0 or 1",
        "value_11": "0 or 1",
        "value_12": "0 or 1"
      },
      "description": "An array of 0 or 1 of 12 values in total, representing each month. 1 if a month is
in cool period or 0 if a month is in heat period "
    },
    "heat_setpoint": {
      "datatype": "integer",
      "value": " An integer value ",
      "description": " Degree setting on a thermostat controller"
    },
    "vent_point": {
      "datatype": "float",
      "value": " A float Value",
      "description": " The amount of air changes in a thermal zone "
    },
    "internal_gains": {
      "datatype": "float",
      "value": " A float value",
      "measuredIn": "watt/m^2",
      "description": " The heat generated by internal heat sources "
    },
    ...
  }
}
```



Building systems

```
{
  "photovoltaic_system": {
    "b_system_id": {
      "datatype": "string",
      "value": "building_001",
      "description": "A unique string defining a photovoltaic system ID - Primary Key for Photovoltaic Systems Table"
    },
    "nom_power": {
      "datatype": "float",
      "value": "A float value",
      "description": "Installed system power of a photovoltaic system"
    },
    "orientation": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "degrees",
      "description": "The orientation of a photovoltaic system "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "degrees",
      "description": "The angle of a photovoltaic system was installed"
    },
    "inst_year": {
      "datatype": "integer",
      "value": "An integer value",
      "description": "The year that the photovoltaic system was installed"
    }
  }
}
```

```
{
  "battery_storage_system": {
    "b_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a battery storage system ID - Primary Key for Battery Storage Systems Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of the electrochemical element that a battery is made of"
    },
    "capacity": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "kilowatt",
      "description": "The capacity of a battery storage system"
    },
    "grid_connection": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of grid connection"
    },
    "inst_year": {
      "datatype": "integer",
      "value": "An integer value",
      "description": "The installation year of a battery storage system"
    }
  }
}
```



```
{
  "ev_charger": {
    "b_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining an electric vehicle charger system ID - Primary Key for EV
Chargers Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of an electric vehicle charger can be either single phase or three phase"
    },
    "total_power": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "kilowatt",
      "description": "The nominal power of an electric vehicle charger"
    },
    "plugs_num": {
      "datatype": "integer",
      "value": "An integer value",
      "description": "The number of plugs an electric vehicle charger provides "
    }
  }
}
```

Zone systems

```
{
  "heating_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a heating system ID - Primary Key for Heating Systems
Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": "What type is a heating system "
    },
    "energy_source": {
      "datatype": "string",
      "value": "A string value",
      "description": "The energy source used to power up a heating system "
    },
    "efficiency": {
      "datatype": "float",
      "value": "A float value",
      "description": "Overall efficiency of a heating system "
    },
    "capacity": {
      "datatype": "float",
      "value": "A float value",
      "description": "Total capacity of a heating system "
    },
    "cov_ratio": {
      "datatype": "float",
      "value": "A float value",
      "description": "The annual average ratio of the energy need for heating covered from this
specific heat production unit "
    }
  }
}
```



```
{
  "cooling_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a cooling system ID - Primary Key for Cooling Systems Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": " What type is a Cooling System"
    },
    "energy_source": {
      "datatype": "string",
      "value": "A string value",
      "description": " The energy source used to power up a cooling System"
    },
    "efficiency": {
      "datatype": "float",
      "value": "A float value",
      "description": " Overall efficiency of a cooling system "
    },
    "capacity": {
      "datatype": "float",
      "value": "A float value",
      "description": " Total capacity of a cooling system "
    },
    "cov_ratio": {
      "datatype": "float",
      "value": "A float value",
      "description": " The annual average ratio of the energy need for cooling covered from this specific cooling production unit "
    }
  }
}
```

```
{
  "dhw_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a DHW system ID - Primary Key for DHW Systems Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": " What type is a DHW System "
    },
    "energy_source": {
      "datatype": "string",
      "value": "A string value",
      "description": " The energy source used to power up a DHW System"
    },
    "efficiency": {
      "datatype": "float",
      "value": "A float value",
      "description": " Overall efficiency of a DHW system "
    },
    "cov_ratio": {
      "datatype": "float",
      "value": "A float value",
      "description": " The annual average ratio of the energy need for DHW covered from this specific heat production unit "
    }
  }
}
```

```
{
  "solar_thermal_collector_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a solar thermal collector system ID - Primary Key for Solar Thermal Collector (STC) Systems Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": " What type is a solar thermal collector (STC) System"
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": " The area that a solar thermal collector panel covers "
    },
    "efficiency": {
      "datatype": "float",
      "value": "A float value",
      "description": " Overall efficiency of a solar thermal collector system "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": " The orientation of a solar thermal collector panel "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " degrees Celsius ",
      "description": " The slope of a solar thermal collector panel "
    }
  }
}
```

```
{
  "lighting_system": {
    "z_system_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a lighting system ID - Primary Key for Lighting Systems Table"
    },
    "type": {
      "datatype": "string",
      "value": "A string value",
      "description": " What type is a lighting system "
    },
    "energy_source": {
      "datatype": "string",
      "value": "A string value",
      "description": " The energy source used to power up a lighting system "
    },
    "power": {
      "datatype": "float",
      "value": " A float value",
      "measuredIn": " kilowatt ",
      "description": " Total power wattage of the system "
    }
  }
}
```



Space elements

```
{
  "opaque_contact_ext_air": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining an opaque contact external air element ID - Primary Key for Opaque - Contact Ext Air Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": "The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": "The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "m^2 ",
      "description": "The area that an element covers "
    },
    "u_value": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "watt/(m*K) ",
      "description": "The thermal transmittance of an element "
    },
    "absorption": {
      "datatype": "float",
      "value": "A float value",
      "description": "The dimensionless absorption coefficient to the solar radiation, at the external surface of the element "
    },
    "f_shade": {
      "datatype": "float",
      "value": "A float value",
      "description": "The dimensionless shading reduction factor due to the existence of external obstacles and the horizon "
    }
  }
}
```



```
{
  "opaque_contact_ground": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining an opaque contact ground element ID - Primary Key for
Opaque - Contact ground Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": " The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": " The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": " The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": " The area that an element  covers "
    },
    "u_value": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "watt/(m*K) ",
      "description": " The thermal transmittance of an element  "
    }
  }
}
```



```
{
  "opaque_unconditioned_thermal_zone": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining an opaque unconditioned thermal zone element ID - Primary
Key for Opaque - Unconditioned thermal zone Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": "The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": "The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": "The area that an element  covers "
    },
    "u_value": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "watt/(m*K)",
      "description": "The thermal transmittance of an element "
    }
  }
}
```




```
{
  "opaque_internal": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining an opaque internal element ID - Primary Key for Opaque - Internal Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": "The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": "The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": "The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "m^2 ",
      "description": "The area that a opaque internal element covers "
    }
  }
}
```



```
{
  "transparent_contact_ext_air": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a transparent contact external air element ID - Primary Key
for Transparent - Transparent - Contact Ext Air Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": " The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": " The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": " The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": " The area that an element  covers "
    },
    "u_value": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "watt/(m*K) ",
      "description": " The thermal transmittance of an element  "
    },
    "f_glaz": {
      "datatype": "float",
      "value": "A float value",
      "description": " The dimensionless absorption coefficient to the solar radiation, at the external
surface of the element "
    },
    "f_shade": {
      "datatype": "float",
      "value": "A float value",
      "description": "The dimensionless shading reduction factor due to the existence of external
obstacles and the horizon "
    }
  }
}
```



```
{
  "transparent_unconditioned_thermal_zone ": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a transparent unconditioned thermal zone element ID -
Primary Key for Transparent - Unconditioned thermal zone Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": " The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": " The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": " The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": " The area that an element  covers "
    },
    "u_value": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "watt/(m*K) ",
      "description": " The thermal transmittance of an element  "
    }
  }
}
```



```
{
  "transparent_internal": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a transparent internal element ID - Primary Key for
Transparent - Internal Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "style": {
      "datatype": "string",
      "value": "A string value",
      "description": " The type of an element "
    },
    "orientation": {
      "datatype": "string",
      "value": "A string value",
      "description": " The element orientation "
    },
    "tilt": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": "Degrees",
      "description": " The tilt of an element "
    },
    "area": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m^2 ",
      "description": " The area that an element  covers "
    }
  }
}
```

```
{
  "thermal_bridge": {
    "element_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a thermal bridge ID - Primary Key for Thermal Bridge Table"
    },
    "space_id": {
      "datatype": "string",
      "value": "A string value",
      "description": "A unique string defining a space ID - Foreign Key for Spaces Table"
    },
    "length": {
      "datatype": "float",
      "value": "A float value",
      "measuredIn": " m ",
      "description": " The length of a thermal bridge "
    },
    "lin_th_tr": {
      "datatype": " float ",
      "value": "A float value",
      "measuredIn": " W/(m*K) ",
      "description": " Linear thermal trasmitance of a thermal bridge "
    }
  }
}
```

Repository schema



