MASTER

A decision support system that uses linked building performance data from Digital Twins to support Asset Managers

Querying information from digital twins to monitor building performance

Obeid, Hassoeni S.

Award date:
2021

A decision support system that uses linked building performance data from Digital Twins to support Asset Managers

EINDHOVEN UNIVERSITY OF TECHNOLOGY

QUERYING INFORMATION FROM DIGITAL TWINS TO MONITOR BUILDING PERFORMANCE

Architecture, Building and Planning – USRE

Date
04-07-21

Version
Final

h.s.obeid@student.tue.nl

www.tue.nl

Author

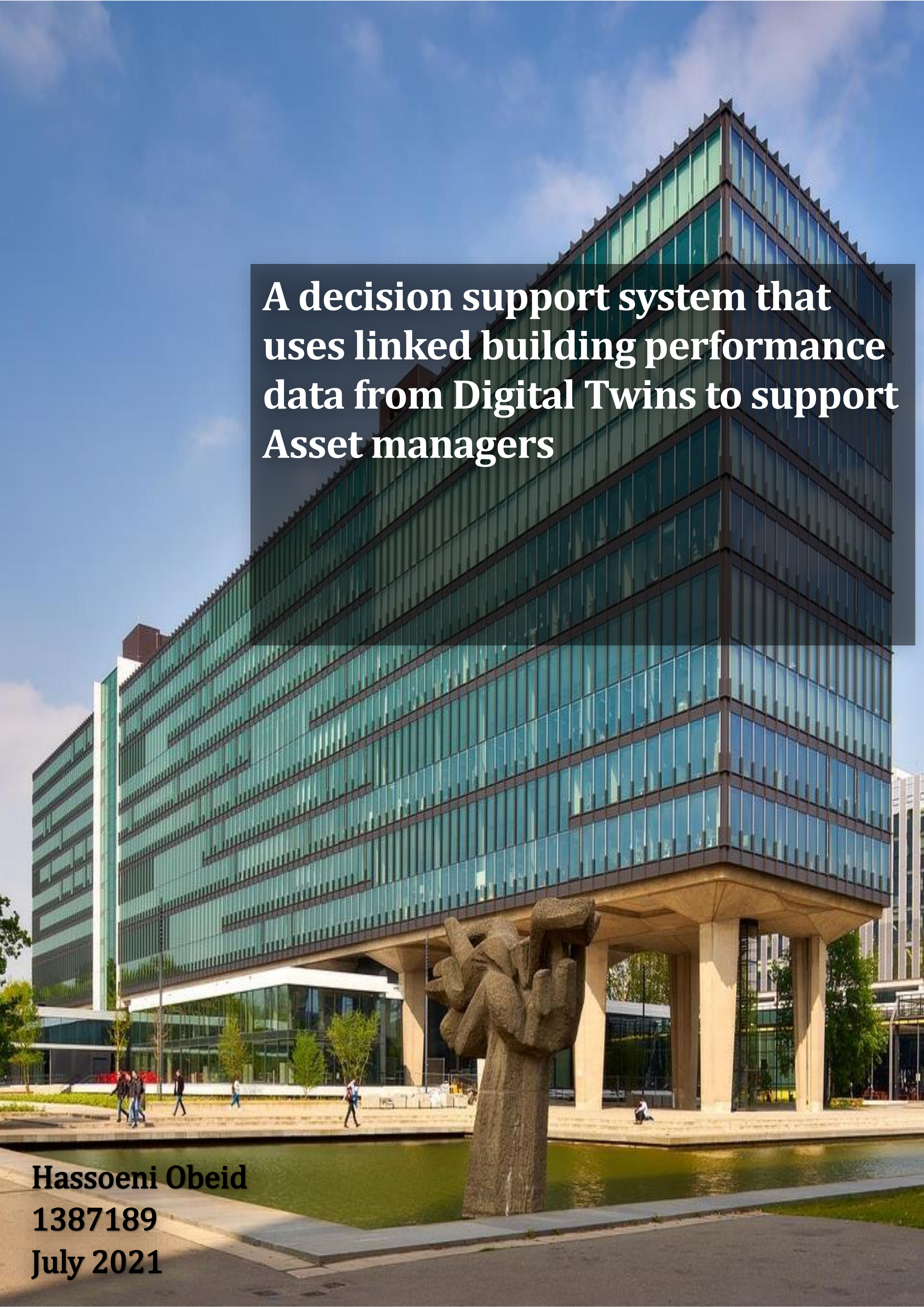
Hassoeni Obeid (H. Obeid)
Student Number: 1387189
Program: Architecture, Building and Planning
Master Track: Urban Systems and Real Estate
Public information: Yes ☒ No ☐
Master Thesis: 45 ECTS

This research has been carried out in accordance with the rules of the TU/e Code of Scientific Integrity.

Supervisors
Dr. Ir. P. (Pieter) Pauwels
Associate Professor, Department of the Built Environment, Information Systems in the Built Environment

Dr. Ir D. (Dujuan) Yang
Associate professor, Department of the Built Environment, Information systems in the Built environment

Ir. J. (Jakko) Heinen
Company owner of BIM Connected



A decision support system that uses linked building performance data from Digital Twins to support Asset managers

Hassoeni Obeid
1387189
July 2021

Acknowledgements

Prior to my master's thesis, I wrote a research paper on digitisation within the AECO industry. During that year, I was able to personally observe that the AECO industry was too slow in tackling the challenges of digitisation. This research gave me the opportunity to contribute to this and to develop myself further within an industry I am very interested in, namely information technology. It was the first time I had come into contact with software development and programming, but I can now look back with satisfaction on the past year and on what I have achieved here.

For this reason, I would like to thank everyone who contributed to this study. First of all, Professor Pieter Pauwels for the highly informative, intensive and above all, fun guidance. Thank you for taking the time. I know how hectic it was, especially in this extraordinary virtual year. In addition, I would like to particularly thank Jakko and all the people within BIM-Connected for the guidance and the opportunities to write the research within BIM-Connected. I would also like to thank Jeroen Werbrouck and Michiel Nuyts for all the advice they gave me on programming.

I would also like to thank professor Dujuan Yang for her expertise on building performance. I would also like to thank all the asset managers who helped me conduct the interviews for my research. Finally, I would like to thank my biggest supporters, my family, for all the support they have given me over the past year. Without them, I could not have done this.

Summary

Buildings are constantly subject to change, and this applies to almost all types of real estate. During the life cycle of the building, all kinds of decisions are made that are supposed to ensure that the building is used optimally during its lifetime. These choices are currently based on decisions concerning the use and consumption of the building. It appears that most of the costs for a building are generated during the O&M phase, which accounts for 60% of the total cost of ownership (National Institute of Standards and Technology, 2016; Al-Kasasbeh & Abudayyeh & Liu, 2020). The other 40% are generated during the other phases of the building process. in addition, buildings are inefficient consumers of resources. Not only do they consume a lot of financial resources, they are also responsible for 36% of global energy consumption and 40% of total CO2 emissions (Maslesa & Jensen & Birkved, 2018). Moreover, most of the financial burden of the asset is generated during the operation and maintenance phase of the building. Nowhere does this energy and financial consumption play a more important role than in commercial and utility buildings. The reason for this is that these types of property are built in large volumes and use a lot of mechanical and electrical equipment, sensors and systems.

The operation and maintenance (O&M) phase usually belong to the domain of an asset manager. Therefore, in recent years there has been increasing interest in exploring the possibilities of implementing effective asset management to reduce the overall consumption of resources during the O&M phase. In this case, effective asset management is understood as the realisation and preservation of value from assets, where value is realised when a balance is sought between eliminating risk, optimising costs and opportunities, and improving the building's performance. Building performance in this context is the concept that focuses on optimising the comfort and energy performance of the building. An asset manager can influence the performance of the building by controlling certain processes. To be able to respond to these processes, the asset manager needs to manage his operations more effectively. For these reasons, interest has grown in recent years with more focus on asset information management during the O&M phase. Asset managers usually rely on a wide variety of tools to help them create, collect and manage the abundant information flows generated during the O&M phase, with the aim of increasing the building's performance by achieving a more effective operational building.

Many commercial software applications have been developed to support asset managers in performing specific tasks. Some examples are the building management systems (BMS) and the building automation systems (BAS). However, this enormous diversity of software tools from different commercial suppliers has led to a vendor lock and isolated, fragmented information islands (Tang et al., 2020). Moreover, these software applications do not have 3D visualisation capabilities. As a result, the asset manager cannot effectively monitor and manage the building (Aziz et al., 2016). On the other hand, an abundance of building-related data is stored in the building information model (BIM), which with its 3D viewer, can be of great value to the asset manager. However, some implementation challenges have been encountered, such as modelling problems, information loss during handover, and issues with integration with other applications (Migilinskas et al., 2013). A combination between BIM and BMS would provide the most significant potential benefits for asset managers. Such as finding information faster, 3D sensor data viewing and improved critical decision making (Pärn et al., 2017).

Following the gap in the current knowledge, this thesis aims at developing a decision support system by creating a semantically rich datastore that contains sensor data and is connected to a BIM model. This decision support system can monitor a building's performance and is then able to present its sensor data in a web interface. In order to improve collaboration and sustainable decision-making. To support this objective, the main research question was developed to investigate to what extent BIM data can be used to realise a decision support system that helps asset managers to optimise the building performance of utility buildings during the operation phase. To answer this question, this research examines the potential use cases for the application and the possibilities of integrating sensor data from the BMS into the application. For this purpose, the possibilities of integration using semantic web technologies are explored. Furthermore, the component architecture is defined for the chosen approach, and the chosen method is evaluated and tested. To successfully implement this approach, a literature study was first carried out. The literature review primarily investigated how previous studies have researched concepts such as asset management and what tasks they perform, what asset management is within a BIM environment, and how previous

researchers have connected BMS sensor data to a static BIM model. This literature review was further supported by a case study of the Atlas Building on the TU/e campus. The potential use case followed from a combination of the literature reviewed and interviews conducted with the asset managers. The interviews consisted of 16 questions related to the topics from the literature review and resulted in 16 requirements for the use case. This use case resulted in a component framework that was tested during an alpha test phase, using the Atlas BIM model for the 8th and 9th floor, with historical sensor data for the temperature and occupancy. The BMS sensor data was made available by the Atlas Living Lab.

To build the use case, the building performance indicators were first identified and then prioritised according to the literature review. This led to the identification of the most significant indicators and the best opportunities for improvement for the asset managers. These were the indoor environmental quality (IEQ), energy and occupancy indicators. The IEQ indicators were built up from HVAC-related building components that primarily focused on the comfort perceived by the occupants. The literature review, in combination with the interviews conducted with the asset managers, resulted in a further scope narrowing and the final user requirements. Instead of the energy indicators, the IEQ and occupancy indicators were considered more important. At the same time, the following final objective for the user requirements could be drawn up. The asset manager must be able to display the triple store visualization of the BIM model in a web UI and at the same time be able to access the sensor data from the Atlas living lab within a graph.

To implement these functionalities in React, various React libraries were used, such as Axios, to handle the communication of the application programming interfaces (API) with the back end. In addition, the Chart.js library was used to display the sensor data in graphs. Moreover, the application made use of the building topology ontology (BOT), and property set definition ontology (PROPS). These ontologies define concepts in a domain and describe the relationships between them. As a result, they formally represent all knowledge about a specific domain and are indispensable in describing the sensor data. This semantically rich data was stored in the Turtle (TTL) file and uploaded to the GraphDB database in the backend. This enabled the application to communicate with the GraphDB database through SPARQL queries and APIs. In addition to the backend, the application used the geometric viewer from the LBDServer. The building-related BIM data was stored in a glTF file and uploaded to MongoDB. The application then accesses this MongoDB backend through APIs. All communication between the front and back end happens without the user being aware of it, deep inside the application. The connection between the two backends is made through a dual link. On the one hand, the user creates a repository on the backend in which all project-related data is brought together. Subsequently, when the user uploads the TTL and glTF files, the application recognises the project repository and searches for a connection based on an occurring connecting element. Namely, the GUID. Since the GUID appears in both files. The user can then execute a SPARQL with this GUID as a parameter and get a query table with the BMS sensor data.

Out of the 16 initial requirements, 11 were successfully implemented. During the alpha test phase, limitations arose. First, a connection to the Atlas Living Labs server could not be established by means of an API, as that API was not yet available. Therefore, historical sensor data was used and stored locally. This led to dependencies and flexibility issues. In addition, it was not possible to establish a link between the obtained sensor IDs and the sensor IDs of the virtual sensors in the BIM model. One of the reasons for this is that these virtual sensors were not drawn in the BIM model, so they cannot be highlighted or queried. As a solution, the spaces are taken as a reference, as these did physically contain the sensors even though they weren´t modelled in the BIM model. Moreover, this study did not include any observation of user satisfaction. However, the asset managers did indicate that this was an important objective for them. From these limitations also emerged the recommendations. For a follow-up research, all sensor data should come directly from the Atlas living lab, which solves the dependency and flexibility issues. In addition, the application with the BIM model can be made more dynamic through colour coding. The interviews show that there is an interest in giving certain rooms a different colour depending on the occupancy rate. In addition, it would be interesting to investigate whether user satisfaction can be linked live to the BIM model and sensor data. All in all, it can be concluded that the main research question has been answered successfully. The main requirement of integrating sensor data with the BIM model using semantic web technologies has been met, thus enabling the asset manager to formulate critical decisions based on the sensor data.

Abstract

Buildings are inefficient consumers of resources. Buildings and the construction sector together account for 36% of world energy consumption and 40% of total CO2 emissions (Maslesa & Jensen & Birkved, 2018). This is particularly relevant for utility buildings, as they predominantly consist of large volumes and operate with a large amount of mechanical and electronic equipment, sensors and systems. To manage this, it is therefore important to effectively implement asset management and, in particular, asset information management. Currently, a large amount of static building data and dynamic sensor data is generated but hardly linked to each other, while a large amount of asset information is hidden in this combination of data. Therefore, this research aims at developing a decision support system by creating a semantically rich datastore that contains sensor data and is connected to a BIM model. For this purpose, the following research question was formulated: to what extent can BIM data be used to realise a decision support system that helps asset managers to optimise the building performance of utility buildings during the exploitation phase? Asset management is understood as the realization and retention of value of assets, whereby realization of value occurs in the process of balancing the costs, risks, opportunities, and performance benefits. In which the performance benefits are part of the concept of building performance. Building performance is the concept that focuses on optimizing the building comfort and energy performance. To answer the research question, a literature review was conducted. The literature review primarily investigated how previous studies have researched concepts such as asset management and what tasks they perform, what asset management is within a BIM environment, and especially how previous researchers have connected BMS sensor data to a static BIM model. This literature review was complemented by a case study of the Atlas building on the TU/e campus. The potential use case followed from a combination of the literature reviewed and interviews conducted with the asset managers. The interviews consisted of 16 questions related to the topics from the literature review, with the aim of finding out what the user requirements are. A total of 16 requirements were drawn up for the use case. This use case resulted in a component framework that was tested during an alpha test phase, using the Atlas BIM model for the 8th and 9th floor, with historical sensor data for the temperature and occupancy. The case study shows that the application is successfully able to integrate the sensor data and static building data and display these data in graphs. The asset manager is then able to critically formulate decisions on this data. The case study also shows that from the initial 16 requirements, only 11 could be implemented successfully. Based on this, it is recommended to further investigate the most important limitations. First of all, by extending the application with time series data directly from an API. Second, by integrating the sensor data more dynamically with the BIM model using colour coding. This increases the chance for an asset manager to formulate even better decisions in real-time. In addition, user satisfaction plays an important role in measuring building performance. For a follow-up study, it would be interesting to investigate whether user satisfaction can be linked live to the BIM model and sensor data, so that a complete understanding of the user experience in the building can be developed.

Keywords:
Semantic web technologies, BIM for O&M, semantic web technologies for asset management, decision support systems

Contents

List of abbreviations

BP	Building performance
O&M	Operation & maintenance
DT	Digital Twin
CMMS	Computerized Maintenance Management System
EAMS	Enterprise Asset Management System
AECO	Architecture, Engineering, Construction and Operation
LD	Linked Data
LBD	Linked Building Data
SSN	Semantic Sensor Network ontology
BACS	Building Automation and Control Systems
BACnet	Building automation and control network
BOT	building topology ontology
PROPS	property set definition ontology
BMS	Building Management Systems
BAS	Building Automation Systems
IAM	The Institute of Asset Management
API	Application Programming Interface
BIM	Building Information Model
HVAC	Heating, Ventilation, Air Conditioning
IFC	Industry Foundation Classes
IEQ	Indoor Environmental Quality
HMI	Human Machine Interface
SCADA	Supervisory Control and Data Acquisition
PLC	Programmable Logic Controller
IoT	Internet of Things
IT	Information Technology
TTL	Turtle
glTF	GL Transmission Format
UI	User Interface
RDF	Resource Description Frameworks
JSON	JavaScript Object Notation
DOM	Document Object Model

1 Introduction

The built environment is subject to intense change. Increasing attention to sustainability and circular development and various shocks towards the market, such as the Dutch nitrogen crisis and the recently erupted corona pandemic, have led to a negative effect on, for example, project development in the real estate market, the volume of property sales[1], the development of new and stricter policies and a change in the way buildings should be used and valued (Tanrivermis, 2020). All of this combined with increased customer requirements to achieve better performing, cheaper, and more efficient real estate (Jurczynski, 2020).

To address this, Roberts (2020) emphasises the importance of asset management for achieving optimal life cycle performance. For the time being, however, the role of an asset manager is fully represented in the operational phase. Nevertheless, according to Al-Kasasbeh et al. (2021), this separation of the asset manager's involvement and input during the design and construction phases, as well as the lack of an effective system for managing the building-related assets, leads to significant unnecessary costs incurred by the current asset management activities. This is partly due to the asset manager's inability to make the right decisions during the operation and maintenance (O&M) phase. Further understanding of this problem requires a disentanglement of the role of the asset manager. According to the Institute of Asset Management (2015), the ISO 55000 standard states that asset management is the collaborative activity of an organisation to realise/recover value from assets, in which realisation of value occurs in the process of balancing the costs, risks, opportunities and performance benefits through either the approach or planning and implementation of pre-set up plans.

In response to this, Brous et al. (2015) stress that asset management organisations should, as a solution, increasingly put more effort into further digitisation of their business operations by incorporating more IT-driven systems. In order to achieve this higher value from assets. According to Al Dakheel et al. (2020), it is clear that IT systems that are part of assets, also known as smart buildings, whose sensors are linked to a variety of components, such as HVAC, lighting, and shading systems, are important in measuring the building performance. For example, by using Internet of Things, a smart building can retrieve external weather information and adjust its internal setting to suit the end-user, allowing the building to be efficient in its energy consumption. In doing so, the building achieves an improvement in its building performance through energy reductions.

Several definitions are surrounding the term building performance. According to de Wilde (2019), building performance represents the concept of quantifying how well a building fulfils its own function. In this context, performance refers to how well the building fulfils its function through its inherent behaviour. This fulfilment of its own function is strongly connected to the concept within asset management, where value realisation is pursued. The value of the building is therefore dependent on the function that the building is expected to fulfil (Lima, McMahon & Costa, 2020). If an object or system is used, it has a value. If it is not used, it is "worthless". A building may be in excellent condition, but it has no functional value if it does not fulfil an enabling function. The type of value that an object fulfils can differ from economic, aesthetic, technical, political, symbolic to operational value. According to de Wilde (2019), Preiser & Nasar (2008) and Li et al. (2020), there are more than 83 indicators that can be identified as indicators to manage the building performance. According to these authors, the most important indicators when measuring building performance are the energy and comfort indicators. Comfort indicators include all indicators that relate to the comfort of the user, such as CO2 values, temperature, humidity and air velocity. After all, the aim of any building is to provide the user with the most optimal comfort experience possible. According to Al Dakheel et al. (2020), all this must be monitored in real-time by distributing system operators in a distributed network and implementing building energy schemes.

[1] COVID 19 Impacts on Netherlands Real Estate | Coronavirus | Netherlands | Cushman & Wakefield

Although the tasks of an asset manager have become more complex, Elhakeem & Hegazy (2012) suggest that realising smart buildings incrementally enables an asset manager to better assess the current condition of the building. So that he can subsequently better predict future depreciation, with the result that he is able to strategically select what to repair and how to develop repair strategies, and thus at the same time can relocate and prioritize funds and resources. All this is possible because the asset manager is able to spend less time searching for information and has better access to new data sources, which enables him to make efficient long-term strategy decisions. From all these aspects, the latter aspect stands out the most because having access to new data sources opens up new possibilities, which have not yet been fully explored (Re Cecconi et al., 2020).

Looking at the individual software applications that an asset manager often uses in the O&M phase, the most used can be identified as; computerized maintenance management systems (CMMS), building management systems (BMS), and enterprise asset management software (EAMS). CMMS is a software package that stores information about the maintenance of an organisation in a computer database (Wienker & Henderson & Volkerts, 2016). CMMS offers multiple primary maintenance functionalities such as asset tracking, inventory control, preventive & predictive maintenance, and equipment data management. However, according to Lu et al. (2020), it still takes a lot of effort and time for an asset manager to collect the various O&M information they need, such as data stored in the CMMS database, and building related specifications. In addition, there is a lack of an integrated platform that can manage information distributed in different databases and support different activities through the O&M process. EAMS arose as an extension of CMMS and can be defined as a software that creates business process capabilities in all areas of asset management, including asset operations, preventive maintenance scheduling, materials management, asset life-cycle management, product sourcing, supplier contract management, and provides a set of management information for each of these activities (Holland et al., 2005). BMS consist of hardware and software that integrates various open standards such as JSON, BACnet, and REST API and can be defined as a computer control system mainly installed within buildings to allow asset managers to control and monitor the buildings mechanical and electrical equipment through various sensor data. This includes ventilation, lighting, power & fire systems, and security systems (Iddianozie & Palmes, 2020). Some forms of BMS systems are Siemens's Desigo Control Point that simplifies the O&M of HVAC, lighting, and shading in small or medium buildings[2]. Johnson Controls have their own building automation system (BAS)[3]. In addition, Honeywell's web-enabled BMS uses a BAS, a web-enabled user interface, and saves energy by coordinating all energy-consuming loads in a building[4].

Although these systems become operational after the construction of the building, a large amount of data is collected at an early stage by means of the Building Information Model (BIM). BIM technologies cover various aspects of the construction sector, including digital 3D visualisation techniques, the ability to exchange information, and innovative co-development processes (Malcolm, Werbrouck & Pauwels, 2020). According to Re Cecconi et al. (2020), BIM enables professionals to make informed, reliable, and up-to-date decisions throughout the life cycle of an asset. In general, BIM distinguishes itself by the fact that the interpretation of information has shifted from 2-D plans and paper documentation to a fully integrated model containing digital building data (Malcolm, Werbrouck & Pauwels, 2020). Guillen et al. (2016) state that BIM is better related to facility management (FM) than to asset management. However, according to the ISO 55000 standard, in its essence, FM is a part of asset management. Asset management would require an asset manager to make decisions based on a continuous process that requires the integration of all kinds of data, processes, and software systems. (Al-kasasbeh et al., 2020).

Al-Kasasbeh et al. (2020) believe that the number of BIM tools that support asset management processes is far too limited. Moreover, attempts to enrich 3D models with asset component data in the current BIM platforms are turning out to be very challenging. This is because there is a discrepancy between standard built-in classification systems and the asset management data structure. Some integrated and comprehensive O&M management solutions have been

[2] Top 10 Building Management System Companies in the World 2019 | Global Building Management System Market Research - Technavio

[3] Metasys® Building Automation Systems – BAS | Johnson Controls

[4] Specification Tools | Honeywell

proposed in which BIM has been applied and systems have been developed to improve data interoperability and integration. Furthermore, development on an efficient and detailed BIM-based data integration tool that can be maintained and updated throughout the O&M phase is ongoing (Lu et al. 2020). This demonstrates that there is still room for a platform system that links O&M building data.

According to Zhang et al. (2009) and Guillen (2016), the real strength of BIM lies in its current state benefits for cooperation, risk reduction, and its potential knowledge database that can be used in combination with other software (Eastman, Teicholz, Sacks, and Liston, 2011). In order to bring out this true strength of BIM, both Dibley 2011) and Dave et al. (2018) point out that BIM needs to be connected to real-time data collection systems in order to improve the efficiency of the O&M processes. The necessity to monitor and govern assets (manufactured components, buildings, etc.) throughout their life cycle, along with technology advancements, has prompted numerous study fields to investigate the possibility of digital twins (Boje et al., 2020). A digital twin (DT) is the idea of a virtual, digital equivalent of a physical product (Grieves, 2015). BIM is seen as a starting point for creating a DT that works as a semantically rich 3D model that can be used in different applications. The 3D BIM model is fed with time and sensor data to create a simultaneous offline and online simulation of the reality for various areas of interest, such as energy, safety, human comfort, and well-being (Boje et al., 2020). However, the information related to the O&M system within a BIM model is usually given in unstructured digital files, which makes it difficult for the asset manager to derive useful information from them (Hu et al., 2018; Guillen et al. 2016; Soibelman et al. 2008). Since a construction project involves individual parties, several software tools are usually used to carry out specific tasks. This increases the demand for the sharing of data/information between the project parties throughout the project's life cycle. As Shen et al.(2010) state, this makes the exchange and interpretation of data and information difficult as there are too many different exchange flavours available.

Moreover, as Salman (2011) points out, there are two core challenges, namely, a technical and a managerial challenge. According to Bernstein and Pittman (2005), the technical challenge can be divided into three categories: the importance of in-process models that eliminate data interoperability, the requirement to make digital design data computable, and the need for the exchange and integration of information between BIM components. The management aspect focuses on the implementation of BIM. Such as the way different software vendors develop an application to solve certain aspects and not entire processes. This creates a fragmented landscape. There is a great interest in standardization. This standardization is partly solved with the industry foundation classes (IFC). The purpose of IFC is to describe all architectural construction data throughout the life cycle of a building. The IFC remains accessible to all specialised software, facilitating cooperation between the various project partners (Boje et al., 2020). In addition, to some degree, both BIM and IFC have upscaling limitations because they can only offer a limited number of building component classes (Werbrouck et al., 2020). This leads to various possibilities of insufficient integrating and interpretation of O&M processes, elements, and software solutions. Translation of information from native tools, in which various BIM modelling guidelines are followed, into the IFC data model is problematic and challenging.

Recently, de Barros Lima (2020) made building data accessible via the web by means of a web-based BIM application. This solution offers the possibility to give more openness about the exchange of building data. In addition, it becomes possible to display and exchange both spatial and non-spatial data via web servers and browsers, therefore, improving the communication between all stakeholders. In addition, Malcolm, Werbrouck & Pauwels (2020) argue that data can then be directly linked to alternative spatial representations such as point clouds, different types of networks, simplified CSG models, and related semantic data (e.g. facility management data, heritage data, geospatial data) that are presented in a web-friendly way, thus increasing the usability of building data. This highlights the importance of fully digitised and integrated construction data (Ding et al., 2016). In addition, this concept would result in less data being lost when exchanging data and enables stakeholders to deal more efficiently with building data.

To achieve web-based BIM, building data must become web-ready. According to Malcolm, Werbrouck & Pauwels (2020), several initiatives have been launched to prepare building data for the web, such as the Linked Building Data

(LBD) initiative. This initiative aims to develop ontologies and terminologies based on a linked data approach (RDF graphs) to display building data on the web. However, according to Malcolm, Werbrouck & Pauwels (2020), a platform is needed to manage the available linked construction data, with a 3D spatial visualisation of the data. This platform, as described by Malcolm, Werbrouck & Pauwels (2020), is a web-based building data platform that uses (semantic) web technologies, more commonly known as the Linked Building Data server (LBD server). The LBD server is an online application that lets users submit building data and see how it's represented geometrically in a user interface.

Nevertheless, there are still several obstacles in this area of research. For example, linking building data (RDF nodes) to other semantically rich data sources makes it possible to enrich the model with additional information. Moreover, the LBD server does not provide users with a visual representation of the real-time sensor data (Malcolm, Werbrouck & Pauwels 2020; Werbrouck et al., 2020). Thus, there is still a gap in building a web interface that displays the semantic data for certain asset management subjects, for example, on various building performance criteria. By presenting the building performance components for asset managers in an LBD structure, which is also visually interactive in 3D via the web. It will be possible, for example, to use the connected triple store databases to search for additional information and thus access information and data that was not previously possible. This may influence the budget and maintenance considerations during the decision-making process. Therefore, this study aims to build a decision support system (DSS) by creating a semantically rich datastore that contains sensor data and is connected to a BIM model. This DSS can use predefined KPIs for measuring the building performance and then present them in a web interface to improve collaboration and enhance sustainable decision-making intended for assessing the occupants' needs more broadly. This ensures that the user is offered the best possible user experience and satisfaction.

1.1 Research aim

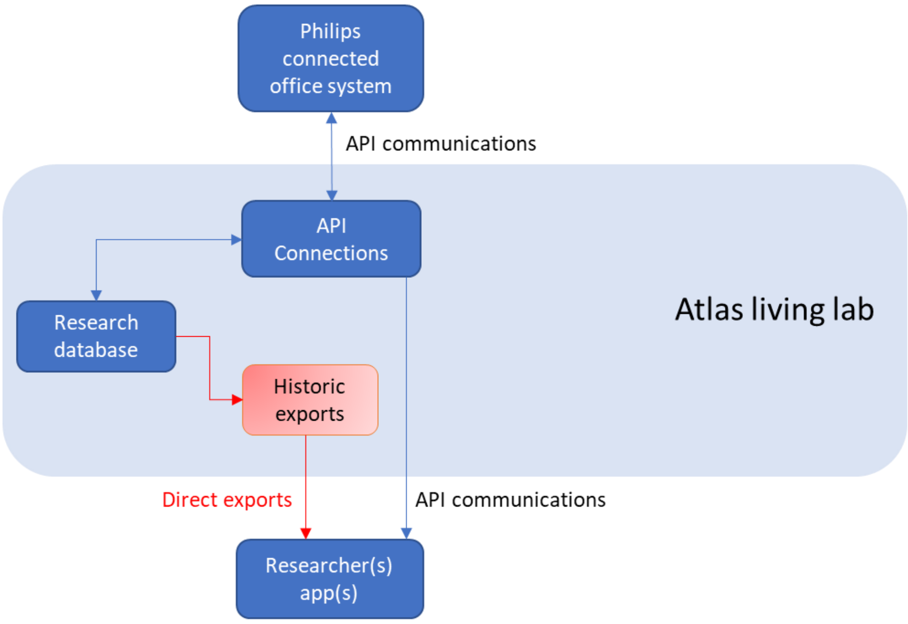
Buildings are constantly subject to change, and this applies to almost all types of real estate. During the life cycle of the building, all kinds of decisions are made that are supposed to ensure that the building is used optimally during its lifetime. These choices are currently based on decisions concerning the use and consumption of the building. It appears that most of the costs for a building are generated during the O&M phase, which accounts for 60% of the total cost of ownership (National Institute of Standards and Technology, 2016; Al-Kasasbeh & Abudayyeh & Liu, 2020). The other 40% are generated during the other phases of the building process. in addition, buildings are inefficient consumers of resources. Buildings and the construction sector together account for 36% of world energy consumption and 40% of total CO2 emissions (Maslesa et al., 2018). According to Li et al. (2020), by quantifying building performance against energy use and targeting this performance, an average of 36% energy consumption can be saved. Nowhere does this energy consumption play a greater role than in commercial and utility buildings. This is because these types of real estate are often built-in large volumes and operate with many mechanical and electronic equipment, sensors, and systems. Therefore, it is interesting for this study to examine the possibilities of effective asset management in this context. In this case, effective asset management is understood as the realisation and retention of the value of assets, whereby realisation of value occurs in balancing the costs, risks, opportunities, and performance benefits. An asset manager can respond to this by controlling certain buttons, which usually happens when an asset manager tries to improve one of the following six aspects: optimize cost & asset O&M, optimize capital investment across the services, optimize asset and network performance, reduce & manage risks and safety, minimize environmental impacts, and improve building performance. Li et al. (2020) suggest that building performance should be the focal point to achieve the above-mentioned energy reduction. This same study identifies multiple building performance KPIs worth investigating, among which three stand out the most. These are energy consumption, land and building use, and indoor environmental quality (IEQ). Moreover, the research conducted by de Barros Lima (2020) shows that there are opportunities for the development of a web-based BIM application that focuses on building performance aspects for asset management.

Therefore, this research tries to contribute to the current scientific literature by improving upon the research conducted by Malcom, Werbrouck and Pauwels (2020) and the research conducted by de Barros Lima (2020) by looking at building performance in the context of web-based BIM systems for asset management. This includes identifying sensor data and KPIs related to building performance and then linking this to both a BIM model and to a semantically rich datastore that can be queried for information. This would enable the possibility to build a decision support system. The overarching aim of this research is to demonstrate that a web-based BIM system enables asset managers to make informed, responsible, sustainable choices and thus become more closely involved in the overall construction process. Moreover, building a system emphasizes the importance of further integration between BIM and asset management.

1.1.1 Scope

To evaluate whether the chosen method provides the right solution, the decision support system will be tested in a living lab environment for utility buildings. In this case, it concerns the Atlas living lab (see, Figure 1), for which floors 3 to 14 have sufficient sensory data at their disposal.

Figure 1 Atlas Living Lab



1.2 Research questions

Based on the problem definition and demarcation, the following main question can be identified.

" To what extent can BIM data be used to realise a decision support system that helps asset managers to optimise the building performance of utility buildings during the exploitation phase?”

To dissect the main question and examine the underlying relationships, it is important to solve the following sub-questions.

Theoretical Part

- Q1 What is asset management and how is it currently related to BIM environments.
- Q2 How is building data managed within and outside a BIM model (cfr. Digital Twin)
- Q3 What is IEQ, occupancy & energy consumption, and how can building performance be quantified and evaluated.

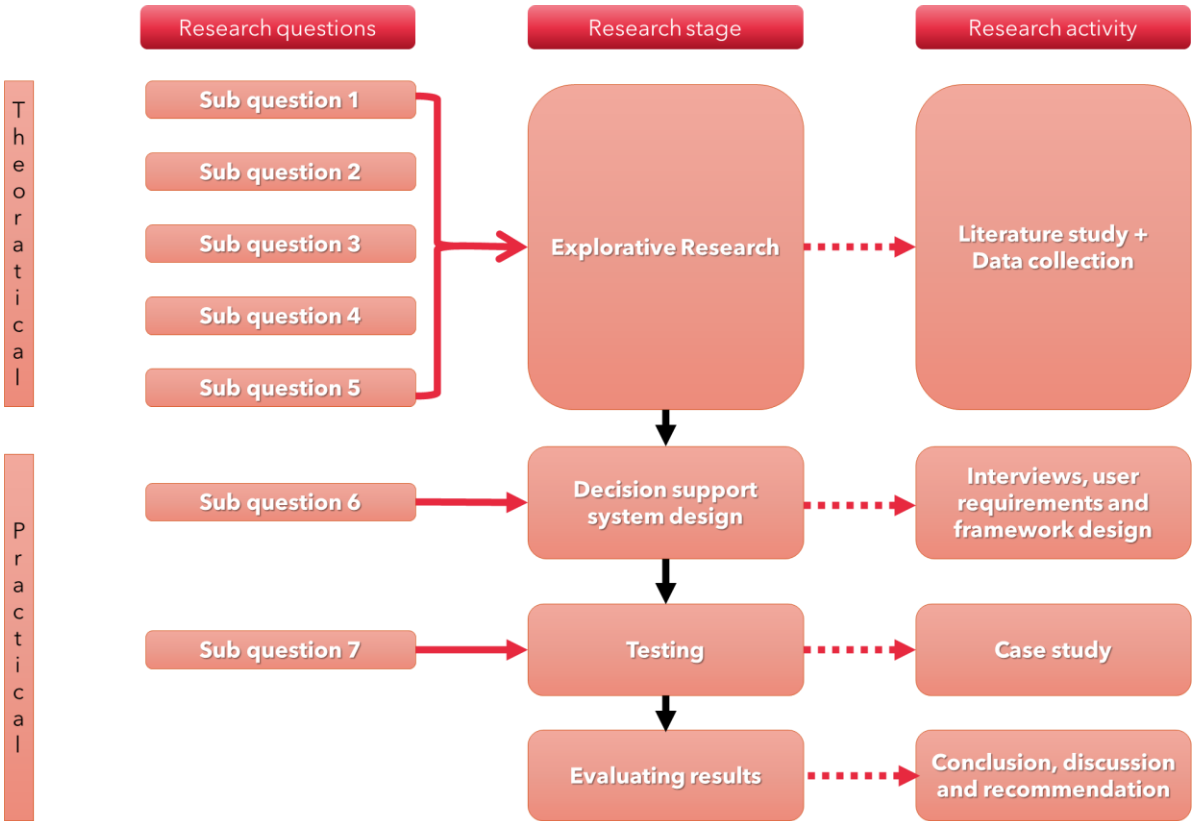
- Q4 Which underlying relationships between BIM and building performance can be established?
- Q5 How can static building data (geometry, element types, properties) be combined with real-time building data (sensor readings, space use, etc.) to form a digital twin of the building?

Practical part

- Q6 What system architecture is needed to build a decision support system from BIM data and building performance related sensor data?
- Q7 How does a decision support system contribute to better asset management?

This research can be divided into four main phases as shown in Figure 2: namely (Phase 1) Explorative research, (Phase 2) Decision support system design, (Phase 3) Testing, and (Phase 4) evaluating the results.

Figure 2 Research Design



1.3 Reading Guide

Phase 1 (Chapter 2) focuses on the literature review. In particular, the different concepts within this thesis are examined and possibilities from previous studies on how BIM and sensor data can be linked using semantic web technologies are explored. Chapter 3 explains the methodology used. The fourth chapter, System Design and Framework, follows from phase 1. This chapter identifies the user requirements and the use case, and creates a component architecture in line with the use case. Chapters 5 and 6 together form phase 3, in which the proof of concept is prepared, and the chosen framework is applied to the Atlas case study. Finally, phase 4 is concluded in chapters 7 and 8. Chapter 7 (discussion) provides a detailed assessment of the results and limitations, while Chapter 8 concludes with the conclusion and recommendations.

2 Literature Review

In this chapter, a literature review is conducted on key concepts within this master thesis. The intended purpose is to gain insight into how previous studies view and connect these key concepts. More importantly it examines how sensor data can be connected to static BIM data. First of all, the concept of asset management is explained, what it entails, how value is created, and what activities asset managers perform. From these activities, consideration is given to how asset managers are supported in their current ways of working by means of current IT systems and BIM implementations. From the limitations in these systems, attention is paid to how semantic web technologies can overcome these limitations. Finally, attention is paid to how semantic web technologies can relate to building performance, and it concludes with a chosen initial framework and overall conclusions.

2.1 Asset management

Asset management is described by the ISO 55000 as the collaborative activity of an organisation to realise/recover value from assets, whereby the realisation of value is achieved by striving for a balance between the costs, risks, opportunities, and performance improvements in the building, with the aim of achieving organisational goals. An asset management system is a methodical approach that links organizational goals to the planned development, coordination, and control of an organization's asset operations throughout various life cycle phases (see Figure 3). Understanding the concept of value is fundamental to understanding the universal objective that an asset manager has in mind for his activities.

Figure 3 Asset management Systems. Source: ISO:55000

According to Lima, McMahon & Costa (2021), there exist several expectations among various stakeholders towards the concept of asset value and the benefits that can be achieved through asset life cycle management. This may involve ensuring financial returns for investors, quality measures, creating value for users, and extending the lifespan of a building through maintenance.

If an object or system is used, it has value. If it is not used, it is "worthless". A building may be in excellent condition, but it has no functional value if it does not fulfil an enabling function. The type of value that an object fulfils can differ from economic, aesthetic, technical, political, symbolic to operational value. If, for example, economic value predominates, then asset management is a means of realising as much economic value as possible. For example, by ensuring that the building does not run out of power or that mechanical systems fail, which will deter occupants from occupying the building. Thus, limiting it in achieving its functional value. Value can also shift; for example, a factory may have economic value, but it is not immediately worthless if it is not used. It may have a cultural or symbolic value, and one can still choose to maintain the building[5]. An asset can take various forms and can be an object, space, personnel, material, or any other entity that generates value for an organization (Farghaly et al., 2018). An asset is referred to in this study as a building.

According to that same ISO 55000:2014 standard, asset managers do not only impact the overall business performance, but these asset managers also identify the expected benefits to the organisation from implementing and improving all asset management activities. According to Riso (2012), asset management capacities should cover resources, processes, and technologies designed to improve and deliver asset management plans, activities and assist their continuous improvement. Moreover, according to Love et al. (2015), asset management enables an organisation to monitor the building and system-specific performance and connect it to the company's objectives. This is done by integrating business operations management systems with each other to exclude separate individual operations.

According to Lima, McMahon & Costa (2021), an asset manager is an entity that is responsible for the decisions concerning the asset, the plans, and the activities, whereby a risk-oriented approach is used related to the objectives of the organisation. Organizations invest extensively in asset management in order to increase the value of their assets and achieve greater returns for their business (Lima, McMahon & Costa, 2020). However, a broader perspective on the overall view on asset management activity areas is needed to identify the exact steps needed to improve specific asset management activities. According to the institute of asset management (2015), there are six activity areas for asset managers outside the field of lifecycle delivery. These can be defined as the context of an organisation, leadership, planning, support, operation, and performance evaluation (see Figure 4).

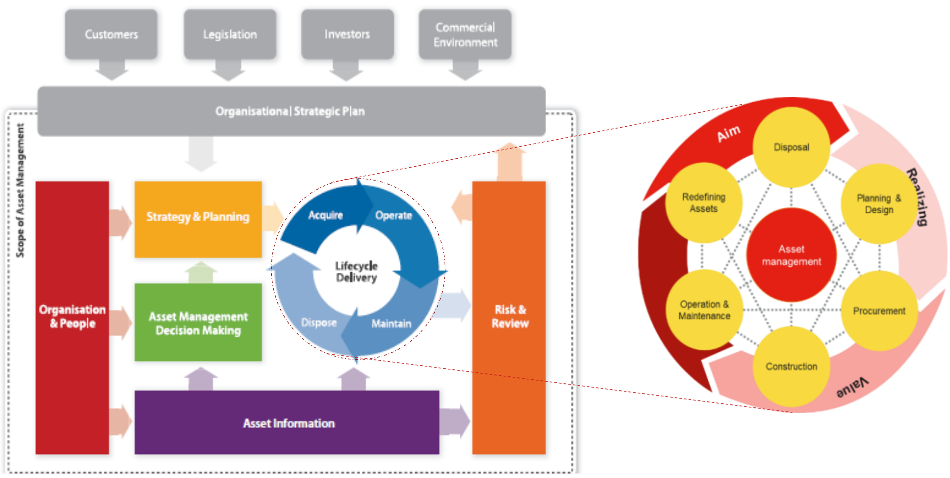


Figure 4 six activity areas for asset management. Combined framework based upon Hasim (2011) framework on lifecycle delivery and the institute of asset management anatomy framework.

[5] Waardecreatie - iAMPro (iampro-portaal.nl)

After evaluating all the above, asset management can thus be seen as a web of activities that occurs during the entire lifecycle of the building, where a connection is sought between the organizational business policy and the realization of value from the assets. This approach is also supported by Farghaly et al. (2018), who sees asset management as a means of translating business objectives into asset-related decisions to achieve optimal financial and non-financial decisions and meet both short and long-term planned goals.

So, the conclusion that can be drawn is, that the asset owner determines what function the building has to fulfil and, therefore, what value can be attached to the building, based upon whether the building is fulfilling the intended function. In the case of Atlas at the TU/e, the building has a commercial function (restaurants and businesses) and an educational function. However, Atlas is also a national monument and therefore has a cultural value. So, there is an economic -, cultural- and social value. These are important values on their own, however, securing the operational value of Atlas is crucial to the functioning of the building and its ability to be occupied. This stems from the various functions that the building fulfils and the number of users it has daily. Therefore, the school building must meet operational standards that ensure the safety of all its users. So, value is created when the operational functioning of the school building is guaranteed.

In order to determine how the operational functioning of the building can be guaranteed, one must first look at the area of activity in which the measurement of the operational activities takes place. For this purpose, the Institute of asset management (2015) lays down the right framework for identifying the overall asset management activity areas. However, there is still a fundamental lack of in-depth understanding of how exactly these asset management activity areas are occupied with key working processes and tasks. Lin, Shien, Gao, Jing & Koronios, (2006) give substance towards this and argue that asset management requires a high level of skill and knowledge in the following activity areas; data acquisition, real-time monitoring, and computer-aided classification and recording of discrepancies from standard operations. On the other hand, according to Lima, McMahon & Costa (2020), one can use the activity areas as set out by the IAM in order to identify a total of 39 asset management work processes and tasks that are responsible for managing the entire asset life cycle. Table 1 contains a combination of the findings mentioned above. First, the 39 asset management work processes are mapped out, as well as the tasks that an asset manager performs within these processes. Next, these are compared with the findings from the study conducted by Lin, Shien, Gao, Jing & Koronios, (2006). After all, they conducted a preliminary study on identifying key processes, see Table 1 for an overview of the analysis and Appendix A for a complete overview of the analysis (GFMAM, 2014; the institute of asset management, 2015).

Activity Area's as defined by (IAM, 2015)	Asset management work Process as defined by (GFMAM, 2014 ; Lima, McMahon & Costa, 2020)	key process (Lin et al., 2006)	Related Activities & Tasks	
Strategy & Planning	1. asset management Policy	×	Make demand forecast/scenarios/history analysis/ management strategies	
	2. Demand Analysis	×	Set up asset management policies & strategies	
	3. asset management Strategy & Objectives	×	Integrating asset management plans with other organizational plans (e.g Finance, HR, health & Safety)	
	4. Strategic Planning	×	Make cost and work volume analysis	
	5. asset management Planning	×		
Asset management Decision making	6. Capital investment Decision making	√	Prioritize capital investment	
	7. O&M Decision making	√	Life cycle analyse (costs, risks, informatie)	
	8. Resourcing Strategy	√	Set up Costumer quality & Asset capabilities requirements	
	9. Lifecycle value realization	×	Maintenance standards and equipment strategies	
	10. Shutdowns & Outages Strategy	×	Criteria set up for decision making	
Organization & People	11. Procurement & Supply chain management	×	Procurement policy, out/insourcing policy, contracts, service level specification, improvement plans	
	12. asset management Leadership	×	Leadership Management strategies/ Gap analysis/ continuity management plan/	
	13. Organizational Structure	×	Culture management strategy/ culture surveys / behavioral patterns analysis	
	14. Organizational Culture	×	Competence framework/assessment, training needs/analysis/specifications	
	15. Competence Management	×		
Risk & Review	16. Risk Assessment & Management	×	Generating/developing/executing Risk management policies	Aligning strategic, tactical and operational risks towards risks registers,
	17. Contingency planning & Resilience Analysis	×	Set up risk mitigation strategies	Set up contingency plan
	18. Sustainable development	×	Environmental/social, and financial impact of AMT plans	Development Change management plans/policies/processes and execute them
	19. Management of Change	√	Define critical measures across all asset lifecycle stages, all linked towards objectives	Establish monitoring programs for the evaluation of performance measure, analysis and the use of this information for management
	20. Assets Performance & Health monitoring	√	Establish criteria in order to monitor deviations in the required asset performance	Establish plans to monitor, measure and evaluate the asset across all stages of the life cycle.
	21. Asset management System Monitoring	×	Predict future asset performance and health	Assessment of purpose of the AMT systems
	22. Management Review, Audit & Assurance	√	Development of audit policies/measurement and execution	Asset valuation register, expenditure reports
	23. Asset Costing & Valuation	√	Stakeholder policies, execution plans.	
	24. Stakeholder Engagement	×		
Lifecycle Delivery	25. Technical standards & Legislation	×	Multiple Acquisition related products such as; Strategy, Request, Agreement, agreement change request and communication reports	
	26. Asset Creating & Acquisition	×		
	27. Systems Engineering	×	Program/project management reports	
	28. Configuration management	×	Various configuration products such as; config management strategies, records, plans, status reports, system releases, baselines and agreements.	
	29. Maintenance Delivery	√		
	30. Reliability Engineering	√	Various system related documents such as engineering management plan, engineering performance measures, requirements, system analysis plan & reports, and description	
	31. Asset Operations	×		
	32. Resource Management	×		
	33. Shutdown & Outage Management	×	Maintenance staffing requirements, tools, strategy & tactics, and ICT infrastructure	
	34. Fault & Incident response	×		
	35. Asset Decommissioning & Disposal	×		
Asset Information	36. Asset information strategy	×	Asset information policy, strategy, business cases, system business requirements	
	37. Asset information Standards	×	Asset information standards and guidelines, asset data dictionary, asset data quality definations and guidelines	
	38. Asset information Systems	√	IT strategy, information systems architecture/strategy & business case / implementation & migration plan / governance	
	39. Data & information management	√		

Table 1 Identifying key asset management activities.

Identifying the work processes and related activities that an asset manager performs is important for narrowing the scope of asset management. This will ensure that the focus can be placed primarily on those activities and processes that are relevant for assessing and improving the overall performance of the building. Lima, McMahon & Costa (2020) managed to identify, based upon the ISO:55001, the most significant key activity areas regarding the improvement of the overall building performance. The main activity areas they have identified are asset management decision making, risk & review, lifecycle delivery, and asset information (see Table 2). These activity areas also explicitly have an underlying relationship with each other. This is because, from the organisational layer, decisions and strategies are delivered on the financing of the O&M phase and which resources should be allocated for conducting the maintenance. From this organisational layer, which is mainly focused on achieving financial performance from the assets by optimising the O&M, on a strategic level, the risks of the building are analysed, and the financial calculations are made, on the one hand by benchmarking the building on the current performance and on the other hand by looking at which maintenance activities lead to a significant impact on the operational asset performance. Finally, from the operational layer, attention is paid towards how to deliver the lifecycle in the most optimal way, based on the current asset information and data, and how to prolong the building's lifecycle. The effect of the above-mentioned key steps results in improving the quality of the building, services and extends the building's lifecycle. All this should achieve the intended goal of reducing the O&M burden through increasing the operational performance of the building and thus lower the total cost of ownership of the asset.

Table 2 Enrichment on Table 1, showing the main asset management activities for improving the performance of the building.

Asset management Definition	Value Delivery	Main activity Area's	#	Asset management work Process	Significant in influencing asset performance
Asset management is the collaborative activity of an organisation to realise/recover value from assets, whereby the realisation of value is achieved by striving to balance the costs, risks, opportunities and performance improvements in the building with the aim of achieving organisational goals	If an object or system is used, it has value. If it is not used, it is "worthless". A building may be in excellent condition, but if it does not fulfill an enabling function, it has no functional value	Asset management decision making	6	Capital investment Decision making	Yes
			7	O&M Decision making	Yes
			8	Resourcing Strategy	Yes
		Risk & Review	19	Management of changes	Yes
			20	Asset Performance & Health monitoring	Yes
			22	Management Review, Audit & Assurence	Yes
			23	Asset Costing & Valuation	Yes
		Lifecycle delivery	29	Maintenance Delivery	Yes
			30	Reliability & Engineering	Yes
		Asset Information	38	Asset & Information Systems	Yes
			39	Data & information management	Yes

A key element in achieving the intended goal of value creation is through the work processes related to asset information and data. For most asset managers, the problems do not lie in the realisation of value within their asset or business processes, nor in the lack of information about their assets. However, the problem primarily lies in the large amount of information that is available and the lack of appropriate methods that describe how to deal with these large amounts of asset data (Munir et al., 2019; Brous et al., 2016). Yet, asset owners are continuously seeking more opportunities to add to the current information stream by seeking accurate, comprehensive, and reliable asset information and data to help support them in making more effective decisions towards the execution of all building-related maintenance and operations (Munir et al., 2019). This information must be maintained and stored for several years to identify long-term trends (Lin, Gao and Koronios, 2006). These asset information and data are mostly created whenever an asset manager performs one of the aforementioned asset management tasks through the use of systems, technologies, or techniques. Munir et al. (2019) define this as asset information management, which encompasses the integrated processes, techniques, and technologies involved in organising, storing, controlling, analysing, and mostly connecting asset information across different sets of information models found within a single database or connected through several alone standing databases.

2.2 Asset management IT tools

Asset managers usually depend on a combination of various digital tools that help them to create, collect and manage the abundant information flows that are generated during the O&M phase, in order to achieve an improvement of the building performance by achieving operational excellence. Operational excellence refers to the realisation of an optimal process that focuses on a higher form of automation and visualisation in the field of building maintenance (von Rosing et al., 2015). The diversity of tasks related to the set-up of maintenance management requires that computer systems are used more efficiently, in order to reach this operational excellence for building performance.

Nowadays, Computerized Maintenance Management Systems (CMMS) are used in all kinds of sectors, and they have proven to be very beneficial for planning, organizing, and controlling the administration of maintenance activities. In general, a CMMS system is a software package that stores information about the maintenance of an organisation in a computer database (Wienker & Henderson & Volkerts, 2016). A CMMS offers multiple primary maintenance functionalities such as asset tracking, inventory control, preventive & predictive maintenance, and equipment data management. However, according to Lu et al. (2020), it still takes asset managers a lot of effort and time to gather the necessary O&M information. For example, data stored in CMMS databases or information about specifications, or 3D models are still missing. In addition, there is a lack of an integrated platform that can manage information distributed in different databases and support different activities through the O&M process. In addition, some individual CMMS programs are able to solve some of these aspects. However, they are vendor locked by commercial parties who manage them.

CMMS can be seen as a part of EAMS, in which CMMS focuses mostly on the O&M activities, while EAMS covers the entire lifecycle of asset management from acquisition to sale. EAMS, therefore, originated as an extension of CMMS and can be defined as software that creates business process capabilities in all areas of asset management, including asset operations, preventive maintenance planning, materials management, asset life-cycle management, product sourcing, supplier contract management, and provides a set of management information for each of these activities (Holland et al., 2005). EAMS thus maximises the overall performance of physical and capital assets, which directly or indirectly impact the achievement of corporate objectives (Lin, Gao and Koronios, 2006). However, EAMS also has its limitations. Previous studies indicate that EAMS is especially hampered by a lack of high-quality data (Kim et al., 2007, Woodhouse 2001, IPWEA 2002; Lin, Gao and Koronios, 2006).

This lack of high-quality data-led software application developers to create Building management systems (BMS) or the closely related Building automation systems (BAS). A BMS is a computer system installed in buildings to control and monitor the mechanical and electrical systems, including HVAC (heating, ventilation, air conditioning), lighting, power systems, fire systems, and security systems primarily. In comparison, a BAS is an intelligent system comprising both hardware and software and allows systems including HVAC, lighting, security, and other systems to connect and communicate with each other on a single platform (Joseph, 2018). Furthermore, both these systems have a large number of sensors attached to building-specific elements (e.g., HVAC and security systems) that can then be used to assess the performance of the building in real-time (Kučera & Pitner, 2018). Multiple authors such as Iddianozie &

Palmes (2020) and Sinopoli (2010) also use this definition regarding BMS and BAS systems. They state that BMS systems consist of hardware and software in which various open standards are integrated, such as JSON, BACnet (Building Automation and Control Network) as well as REST API. According to these authors, BMS systems can be described as a computer-controlled system that is mainly installed in buildings to enable asset managers to control and monitor the mechanical and electrical equipment of buildings through various sensor data. This includes HVAC, ventilation, lighting, power & fire systems, and security systems. BACnet was specifically designed to meet the communication needs within building management systems, building automation systems, and control applications such as HVAC and lighting, and describes a mechanism by which computerised devices can exchange information with each other, regardless of the specific function that the device performs within the building. Therefore, the BACnet protocol can be used by BMS or BAS systems, complex controllers, and application-specific sensors and controllers. This creates the desired effect of openness for information and data exchange between different device vendors, thus avoiding vendor lock-in (Tang et al., 2020).

2.2.1 How BMS work

Durier (2017) states that sensors linked to the HVAC systems are able to automatically control energy consumption by controlling the building temperature during different peak and non-peak hours of the day. According to Davis (2015), BMS systems prove to be beneficial when they are connected to a web-based wireless monitoring and control system. Data is extracted from the sensors in the BMS using a programmable logic controller (PLC). Subsequently, the sensor data is read from the PLC by means of a supervisory control and data acquisition (SCADA) system. In order to save and transfer the data, the SCADA system transfers data to an SQL database on a server. Finally, a human-machine interface (HMI) retrieves the stored data and presents it to the asset manager (Valinejadshoubi, Moselhi, Bagchi & Salem, 2021). According to Kallio et al. (2020), BMS and building energy management services (BEMS) use sensors to monitor room conditions in real-time and thus create data for indoor environmental quality assessment. In addition, these authors argue that in some cases, these systems have advanced to the point where they are capable of further individualised personalisation, meaning that the available data can be used to better analyse user behaviour and preferences, such as how users move around the building, in order to optimise the use of the space in the building (Kallio et al. 2020). In the years prior to 2003, BMS suppliers such as Johnson Controls, Siemens and Honeywell provided closed BMS systems (Makonin, 2016). This has changed with the advent of standardization and communication protocols such as BACnet. Yet, advanced features are still locked by limited API access. With the growth of the internet of things (IoT), the accessibility to monitoring building systems via the web has grown. Hossein (2019) sees the opportunities in integrating BMS with the web and argues that it is important to set up BMS systems in buildings. As they are essential in the intelligent management of the mechanical and electrical components, thereby effectively managing energy demand.

According to Peng et al. (2017), it is difficult for asset managers to effectively monitor and structure building information due to a large number of specialized applications, each containing their own unstructured data systems and classification systems for all building-specific elements. Another problem with the current IT tools for asset management is that many of these tools require extensive human input, which makes them time-consuming and sensitive to human error (Peng et al., 2017). In addition, these asset management software applications have little or no interactive spatial visualisation capabilities, leaving them only capable to merely load a jpeg or CAD file into the system (Aziz et al., 2016). This makes it nearly impossible for asset managers to find the exact location of the problem. A joint database is needed to collect and store all asset information during the various phases of a building's life cycle. This problem of using cad and jpeg files for visualisation can be solved by using the 3D visualisation capabilities of the BIM model. Thus, there is added value to be found in connecting these data forms. Using the 3D BIM model in combination with the sensor data from the BMS system, it is possible to display the operational problems in an interactive way with more accuracy to the asset manager.

2.3 BIM

The concept of BIM is based on the idea of using digital building models continuously during the entire life cycle of a building, starting with the conceptual phase and ending with the operational phase. BIM is a good solution for the aforementioned problems. BIM improves the information flow between different parties involved in the various construction phases (Borrmann, König, Koch, and Beetz, 2018). A proper distinction can be made between BIM and BMS (see table 3), where BIM mainly provides static data and describes the building with installations. BMS provides volatile sensory data that shows changes in building use over time.

Table 3 difference between BIM and BMS

	BIM	BMS
Meaning	Building information model	Building management system
Scope	Locations of building Spatial 3d representation Device location	Building automation Remote monitoring Remote control
Data volatility	Low (Generally static data)	High (Automatically collected sensory data)
Typical usage	Reference documentation and geospatial visualisation	Online controlling and monitoring of sensory data

By using a BIM method, IT is used many times more during the different building phases. As a result, information isn't stored in drawings anymore. But rather, BIM keeps track of information, stores information and exchanges information through its digital representation. This improves a number of aspects, including the coordination of design activities, the integration and simulation of building information, and the control and transfer of this building information. The intended purpose of BIM is to minimize manual data entry and stimulate the reuse of digital information (Borrmann, König, Koch, and Beetz, 2018).

Although BMS with their human-machine interface provide useful information for the users. In recent years, attempts have been made to use BIM to replace the human-machine interface in BMS systems. Instead of using two-dimensional vector graphics, the 3D representation of the building can be used as an integrated information model for monitoring various building-related performance indicators. This enables an asset manager to perform inspections and monitor building components in the 3D model, which was previously much less accurate in the HMI. This reason for this is that BIM is capable of including information about individual building elements which the HMI was unable to do. So, for any event, whether it is an HVAC failure or temperature loss due to cracks in the building, whenever BIM contains sufficient information at its disposal, it is able to provide asset managers with fast and appropriate information so that they can deliver efficient and effective solutions (Valinejadshoubi, Moselhi, Bagchi & Salem 2020). Despite all the possibilities that the information in BIM offers for any implementation in facility management and asset management activities, there are still many obstacles that stand in the way of BIM solutions (Love et al., 2015). According to Migilinskas et al. (2013), the following problems and obstacles can be identified for BIM concepts. First, a limitation is caused by the incompatibility of the data when it is transferred to other participants. Each time this happens, the non-transferred data has to be retrieved, which leads to unnecessary loss of time and possibly higher costs. There is also the fear of little success, few or major project failures, high initial investment costs, and the learning curve, which keeps people apprehensive about implementing BIM concepts. Finally, a lack of information about strict BIM implementation standards and rules in certain countries and regions pose an obstacle for project participants to implement BIM concepts.

According to Pärn et al. (2017), BIM makes it possible to significantly reduce the time spent on finding semantic O&M data together with the geographical data for the material and building components. With the help of this data, it is possible to provide the IT tools used for O&M with building data, thus making it possible for the asset manager's work to be more precise and less time-consuming. Furthermore, according to Nical & Wodynski (2016), in the context of asset management, BIM, with the help of real-time resource location technology (e.g. RFID), can make it possible to track assets more efficiently. This is particularly important because BIM, with its 3D visualisation capabilities, can increase reliability to a certain degree within the field of asset management.

Nevertheless, according to Faghaly et al. (2018), the intended benefits cannot be ignored when striving for an integrated BIM system with asset management data. The resulting benefits, if properly implemented, will enable an organization to make better decisions, reduce costs, see an improvement in their investments, reduce financial losses, improve health and safety, and lead to better overall building performance and greater comfort for users.

Approaches such as the industry foundation classes (IFC) ensure that it is possible to improve sharing information between two parties by breaking down information and sharing this through a common data format. IFC is a standardized, digital semantic description of the built environment. It is an open, international standard developed by buildingSMART and is used to exchange information between two parties (see, Figure 5).

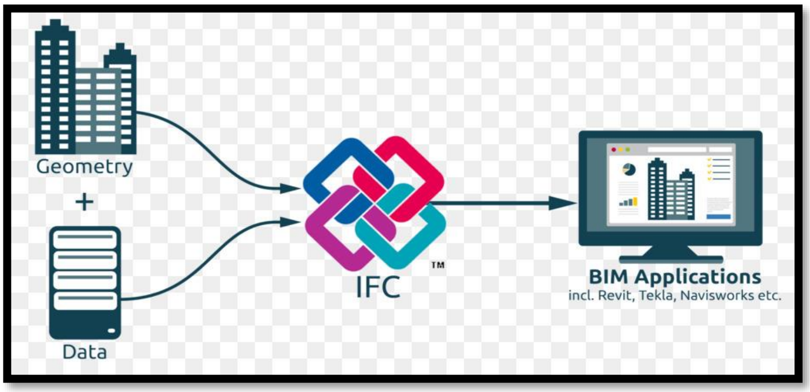


Figure 5 How IFC works as described by Buildingsmart, (2020)

The desired IFC data can be exported in various formats, including JSON, XML, STEP and RDF, and then exchanged over the web, as a file, or stored in centralised or in interconnected databases (BuildingSMART, 2020). Despite the successes that IFC has brought for exchanging data to the industry, there are still limitations. For example, the problem with IFC is that IFC schemes do not contain a solid method for examining and assessing whether the BIM models translated from IFC contain correct and semantically accurate information without unintended geometric translation (Lee et al., 2019). In addition, to some degree, IFC is limited as it can only offer a limited number of building component classes (Werbrouck et al., 2020). Moreover, the EXPRESS and XSD languages are too limited to define semantics, making it difficult to use generic reasoning and query methods (Bonduel et al., 2018). In addition, IFC is not able to link building information stored in the IFC file with data sources on the Web (e.g., making a connection with local building regulations, or geographical information and other such information stored on the Internet). This leads to various possibilities of insufficient integrating and interpretation of O&M processes and elements. Thus, there are still gaps that are not solved by the current exchange standards.

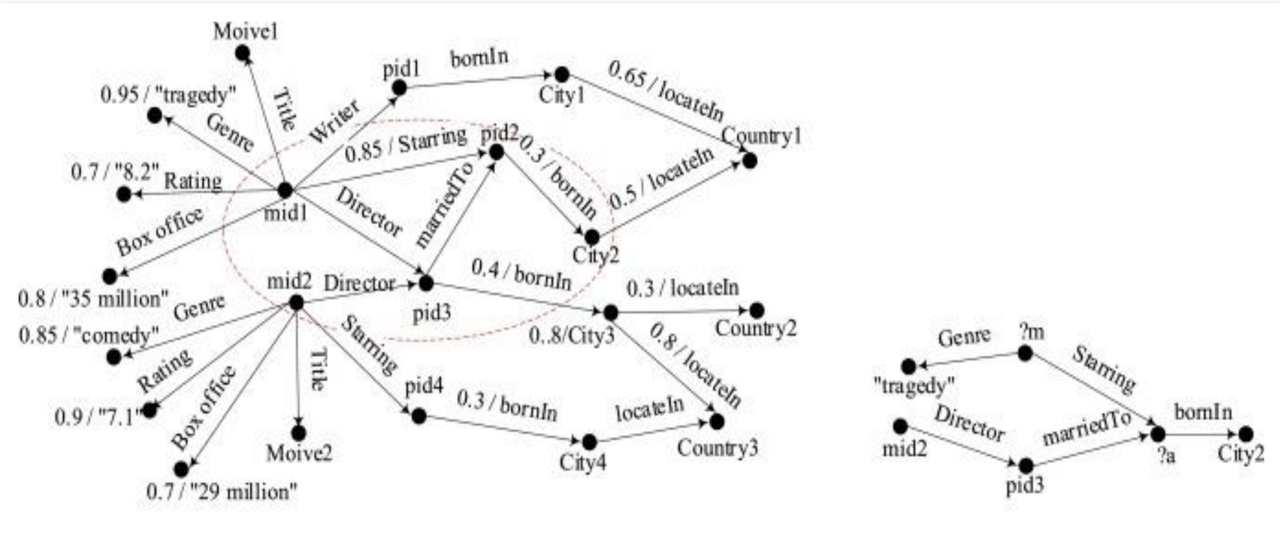
2.4 Principles of Linked Data, ontologies and Linked building data

One way to solve the problems with both the IFC approach and current asset management software systems, while simultaneously providing more openness regarding the exchange of building data, is to move to a web-based BIM solution driven by ontologies and utilizing a linked data principle. The benefits of a linked data approach are extensive. The term linked data refers to a collection of methods for publishing structured data on the Web. In other words, linked data is a way of using the web to create typed links between different data sources. This ranges from heterogeneous systems within a single organization to two individual databases spread across two different locations and organizations. Linked data is expressed in open, non-proprietary formats, meaning that it remains easily accessible for other applications to make use of.

Furthermore, it is modular, as it is well able to be connected to other forms of linked data. This also makes the system scalable (Curry et al., 2013). In addition, linked data is able to bring several benefits to the construction industry through its decentralised integration. According to Curry et al. (2013), it is designed in such a way that it is linked mainly at the lowest level (the data level) and not at the entire system level (the infrastructure level). In this way, it increases the data sharing capabilities of current IT systems, creating more openness and more opportunities to collaborate. This makes the system robust in overcoming the IFC limitations.

To achieve web-based BIM, building data must become web-ready. According to Malcolm, Werbrouck & Pauwels (2020), several initiatives have been launched to prepare building data for the web, such as the Linked Building Data (LBD) initiative. This initiative aims to develop ontologies and terminologies for the built environment using a linked data approach. Linked building data stems from the linked data concept introduced by Berners-Lee (2006)[6]. Linked Data uses Resource description frameworks (RDF) to make statements that link random objects with the world. An example of an RDF database can be seen in Figure 6 (Li et al., 2019). RDF contains an abstract syntax that represents a graph-based data model, as well as a logical semantics that offers a foundation for well-founded interpretations in RDF data. This RDF structure features a basic data representation that is straightforward to parse and alter by applications. As a result, it is only designed to be read by machines and not expressly by people (Berners-Lee, 2004).

2.4.1 Linked Data



(a) A fuzzy RDF data graph *G* inspired by IMDB (b) A query graph *Q*

Figure 6 RDF Graph based upon Li, Yan & Ma (2019) research.

Linked Data (LD), or more commonly known as the "web of data", can essentially be seen as data made available as RDF graphs (Bizer, Berners-Lee & Heath, 2009). Tim Berners-Lee envisioned the concept of Linked Data as an idea of using the web to connect data in order to create a global knowledge database. It, therefore, extends the web by making it possible to share and publish raw data using open standards (Djuedja et al., 2021). These open standards, as stated prior are known as RDF data models, Uniform Resource Identifiers (URI), Simple Protocol and RDF Query Language (SPARQL), or the Web Ontology Language (OWL). Radulovic et al. (2015) and Bizer, Berners-Lee & Heath (2009) described the basic principles of LD as: The first component is to assign URIs to the entities described by the dataset and ensure that these URIs can be read in RDF views via the HTTP protocol. The second component is to

[6] https://www.w3.org/DesignIssues/LinkedData.html

establish RDF links to other data sources on the Web, allowing clients to browse the Web of Data in its entirety by following RDF links. Finally, the third component is to also provide metadata about the published data so that clients can evaluate the quality of published data and can choose between different ways of accessing it.

Linked open data is data that can be freely used and shared and follows a structured design principle for sharing machine-readable cross-linked data on the web[7]. Berners-Lee (2010[8]) suggested a 5-star implementation scheme for linked open data, intending to add a star when proprietary formats are removed, and links created within the data set. One-star data is data that is available on the web, in all formats but with a public licence. This allows users to search, store, change and share the data with each other. Two-star data come into being when one-star data becomes available as machine-readable structured data. This is, for example, the case when an excel file is used instead of an image scan. In addition to everything that can be done within 1-star data, 2-star data allows a user to use software to process and export the data in other formats. Three-star data goes to users who do not need individual software to analyse data, such as CSV formats that store data in text format. Four-star data goes to data formats that use open standards from W3C, such as RDF and SPARQL, to find data and information. Finally, five-star data, which uses W3C standards and linked data principles, links data to other people's data to provide context. This allows users to find more interlinked information from this data, as graph databases become more capable of creating new links from facts. Thus, in its essence, LD ensures that different data sets are connected to each other by means of links and thus form non-isolated data islands (Radulovic et al., 2015).

2.4.2 Ontologies

In order to gather and share these building data, LBD uses a set of vocabularies such as Building Topology ontology[9] (BOT), Product ontology (PRODUCT)[10], property set definition ontology[11] (PROPS) (Djuedja et al., 2021; Bassier et al., 2020a). These ontologies define concepts in a domain and describe the relationships between them. As a result, they formally represent all knowledge about a certain domain and are indispensable for the LBD initiative, as they are the models for the representation of data on the web (Radulovic et al., 2015). An ontology can be seen as the result of a set of shared knowledge organised in such a way that it is machine-readable. This implies that knowledge is based on the conceptualisation of objects, and other entities assuming that they exist in a common area of interest and that there exists some form of relationship between them by which they are interconnected. A more comprehensive form of a description of what an ontology is comes from Studer et al. (1998). They define an ontology as a formally explicit specification of a shared conceptualisation, where conceptualisation refers to an abstract model that identifies relevant concepts. Explicit stands for the fact that the types of concepts used must be defined. The term formal means that an ontology must be machine-readable and, finally shared, represents the capturing and availability of open data. Thus, an ontology is a specification of concepts within a domain of classes, wherein properties of classes describe features and attributes of the class[12]. A major advantage of ontologies is their ability to define semantic models of data. An ontology is able to define links between different types of semantic knowledge (Munir & Anjum, 2018).

According to Quinn et al. (2020), asset management BIM systems mainly represent semantic, geometric, topographic (static) data, while IoT sensory data derived from the BAS has a time-series dynamic structure. This difference makes the integration of these data sets challenging. As a solution, Quinn et al. (2020) propose a linked ontology approach, in which an ontology acts as a proxy for linking the sensory data with BIM. This approach has been explored in various forms, such as in the Semantic Sensor Network (SSN) ontology, previously explored by Kučera & Pitner (2016) or the Building Automation and Control Systems (BACS) ontology by Terkaj, Schneider & Pauwels (2017). According to Neuhaus & Compton (2009), the SSN ontology enables one to describe sensors and is commonly used to describe

[7] What is a Five-Star Linked Open Data | Ontotext Fundamentals
[8] https://www.ontotext.com/knowledgehub/fundamentals/five-star-linked-open-data/
[9] w3c-lbd-cg/bot: Building Topology Ontology (github.com)
[10] Product - Schema.org Type
[11] maximelefrancois86/props: Ontology generation from the property set definitions (github.com)
[12] What is an ontology and why we need it (stanford.edu)

BAS or BMS systems data with semantic tags such as building, room, device, or connect them by giving them attributes such as 'hasPhysicalProperty' and 'sensingMethodUsed'. This allows for it to be linked to elements in BIM models (Kučera & Pitner 2016). The SSN ontology is a more comprehensive approach than one specifically developed for the built environment. In order to connect the sensory data from the BMS to the BIM data, a large amount of semantic data must be manually specified, using scripts, for each data item coming from the BMS (Quinn et al. 2020).

BACS, on the other hand, uses a different methodology. The BACS ontology's goal is to support the modelling of information requirements. This includes four requirements: control behaviour in BAS systems, physical devices of BAS systems and their location in the building, smart appliances, and finally, logical typology in a BAS. The BACS ontology approach reuses different existing ontologies such as SSN, BOT, ifcOWL ontology, and Sensor Observation Sample and Actuator (SOSA) as a way to establish connections. For example, BACS uses BOT to describe different spatial data (e.g. floor, space, element), on which SOSA can then be used to describe sensor data and assign sensor attributes within the IoT network. IfcOWL is then used to read sensor readings as a value (Terkaj, Scneider & Pauwels, 2017). This makes the resulting ontology approach far more suitable than only using SSN as an approach, as it is able to link BAS, IoT, and BIM data more efficiently. IfcOWL is also used in the IFCtoLBD tool (Bonduel et al., 2018). Using the LBD ontologies BOT, PROPS and PRODUCT, the IFCtoLBD tool can convert all current IFC schemas into RDF.

2.5 Semantic Web in other studies

The linked building data approach and web-enabled BIM are both part of a concept what is known as semantic web technology. Semantic web technology is a valuable tool for promoting data integration and enabling complex queries in multiple data sources in order to extend the essence of BIM (Mohamed, Abdallah & Marzouk 2020). Recently, many publications have focused on the added meaning of semantic web technology. Table 4 below gives an overview of the analysis in these studies. The studies focus on the integration of BIM data with asset management and/or FM data and are studies using semantic web technologies, ontologies, or a combination of both. In addition, their ability to improve decision-making during the O&M phase has been investigated.

Table 4 semantic web in other studies

Index	*Article*	*Data sources*	*integration Approach*	*3D viewable*	*Decision support framework*
1	Kim et al. (2018)	BIM data, COBie, maintenance work records, ontologies	Data integration with semantic web technologies	×	With the usage of SPARQL, a user is able to retrieve all stored information about a building element
2	Mohamed, Abdallah & Marzouk (2020)	BIM, Ontology, semantic web technologies and open standards.	An ontological system which integrates as-is BIM information with semantic web technologies	×	Semantic integrated knowledge framework that can constitute a semantic database for building components using BIM
3	Hu et al. (2018)	BIM data, RFID, BMS data with additional maintenance & repair data.	Central web app with different databases for the BIM Data, user data and monitoring data	√	The app supports data enquiring, analysis and statistics, which allow for support in maintenance tasks, routine patrol checking and aided emergency responses

4	Dibley et al. (2012)	Ontologies, Sensory data, IFC and BIM	Integration of various ontologies such as; SUMO, Building ontology, Sensor ontology, and supporting ontologies with IFC, SensorML and OntoSensor infrastructure	×	This research uses a multi-agent software framework: OntoFM (ontology-based) for supporting real-time building monitoring.
5	Shen et al. (2012)	An asset tracking system, Facility operation and management system, BIM server	Decentralized system integration using semantic web technologies	√	The information displayed in a single point of access
6	Chevallier et al. (2020)	Sensory data, BIM data	Sensor and BIM data stored within web databases and linked through semantic web	×	Linked data that is accessible through the web to be used in simulation modelling
7	Howell, Rezgui & Beach (2018)	IoT sensory data, GIS sensory, EPANET data	integrating smart meter and user engagement data with supply-side data. In addition, integrating API and visualization for GIS data, telemetry and asset data. Finally, connect all of the above with semantic web technologies	×	Decision support tool that leverages both the data-driven and knowledge-based programming interfaces of the platform.
8	Quinn et al. (2020)	BAS data, FM-BIM data, IoT sensory data	integrate a building BAS system IoT sensory data with BIM through semantic web technologies	√	Through a linked data framework that uses dynamo to link IoT data and BIM, this allows for advanced data analytics
9	O'Donnell et al. (2013)	Energy performance indicators, Real-time Sensory data, IoT	Scenario modelling through data leveraging from different data sets using a linked data approach	×	Multiple applications that use a variety of analytical tools such as scenario modelling, energy modelling and energy dashboards as decision support to support services

Most studies make use of ontologies as a methodology for integrating different systems, such as Kim et al. (2018), who investigated the integration of building elements and asset management maintenance tasks into a database of the asset management system. Through an ontology-enhanced system, the asset management system was able to improve the acceptance and performance of object-based data management. This ontology system (FMontology) linked data from the IFC to the data stored by the asset management system, after which these had been converted to an RDF using the OWL-based FM ontology. The proposed method will ensure that data inputs required for building management are captured accurately and reliably. This creates an opportunity to improve the financial aspects, accuracy, and productivity of tasks related to facilities management through the use of BIM and the semantic web. Mohamed, Abdallah & Marzouk (2020), have created a framework in their research that focuses on the integration of as-is BIM information with semantic web technologies through the usage of an ontology approach to ensure that sufficient qualitative, formal, and adequate information on existing buildings is available.

On the other hand, Shen et al. (2012) looked at a system-level integration. They used a decentralised system integration methodology to create multiple application platforms for integrating data, where information is displayed in a single point of access. The author has developed a total of three web-based applications. First, Shen et al. (2012) developed a maintenance management system. Second, a wireless sensor/RFID real-time asset tracking system (AeroScout), and third, a BIM server (BIM Octopus). This approach of Shen et al. (2012) entails the creation of a flexible platform that meets multiple integration requirements. In addition, it ensures that systems and their underlying subsystems can respond proactively and reactively to events as they occur, thereby improving their versatility and responsiveness. Parallel to Shen's research, Hu et al. (2018a) created a centralised web-based platform for monitoring maintenance of all mechanical, electrical, and plumbing systems in office buildings. Using QR codes and RFID tags, they managed to extract a logical chain about relevant malfunctions from the BIM-FM model. The RFID tags and QR codes ensure that large MEP components can be catalogued and organised. The interface translated the monitored records into readable table data, which is then injected into the BIM database. This enabled employees to place new work orders on the platform. Finally, the platform was able to present a 3D view of the building through the BIM model.

Contrary to Shen and Hu, Dibley et al. (2012) investigated how sensory-based building monitoring can be done through an ontology framework. The framework consists of multiple interrelated ontologies, which include building ontology, sensor ontology, and support ontologies. They succeeded in presenting the ontologies framework OntoFM visually and also in enriching the current ontology for FM. This was done by means of an IFC taxonomy to ontology converter. In addition to Dibley's et al. (2012) framework, Howell, Rezgui & Beach (2018) developed a decision support tool based on an ontology framework. Their tool aimed at extending current GIS tools and developing a water utility dashboard. This was done using an IoT water network. First, the platform makes a call to the Hypercat API to get a list of pilot sites and databases. Subsequently, the platform a second call for sensor detection and alerts. Then, the endpoints are queried with SPARQL to obtain descriptions of the water network objects. A list of sensors can then be selected, and visual graphs extracted. The platform also used a Google Maps API to visualise the water network in web browsers.

Most of these studies require from the user that he is able to interact with the semantic database through a query processor such as SPARQL, and this is preferably undesirable. Therefore, it is important for this study to run the SPARQL queries in the background. Within the context of decision making, Chevallier et al. (2020) used semantic web technologies to create a digital twin of a building, with sensor networks stored in a triple store on the web. These authors used a common naming convention to link the domain ontologies, from which they could query the sensor data within the BIM model. Other databases could then be connected to the triple store using middleware. This made it possible to get more out of the sensor data through analysis. In addition to Howell, Rezgui & Beach (2018), Quinn et al. (2020), and O'Donnell et al. (2013) have also investigated how to integrate sensory data into a platform using IoT. Quinn et al. (2020) found that BAS systems are excellent systems for monitoring building conditions. However, they were not yet included in the design phase, which ended up having multiple asset management databases during the O&M phase. Therefore, they developed a linked data framework that uses dynamo to link IoT data and BIM, resulting in a framework that allows for advanced data analytics. Their framework enables high-frequency, low latency data transmission by sensor point controllers, dedicated databases, and cloud databases with batch analytics. This framework summarises time series data about lighting usage and occupancy rate, ranging from hourly to by the minute update. In order to visualize the data, Quinn et al. (2020) used Dynamo to effectively map time-series data from a CSV file towards a BIM model, see Figure 7.

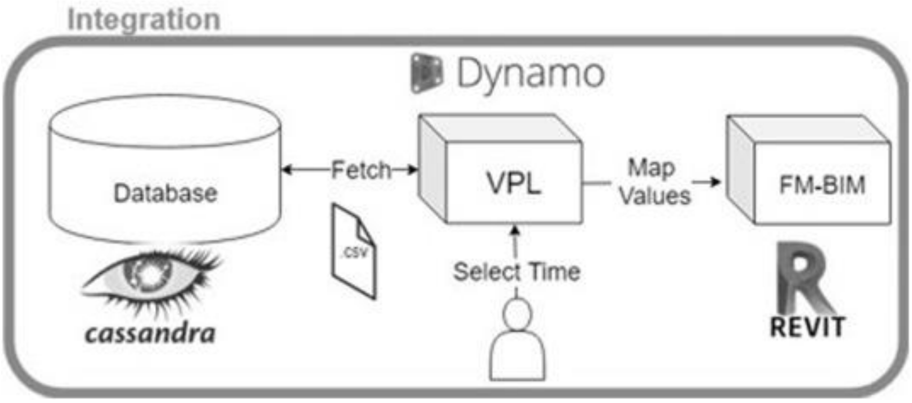


Figure 7 Quinn et al. (2020) integration model

O'Donnell et al. (2013) found ways to use Linked Data and Event Processing as the basis of an approach that aims to help overcome technical barriers to scenario modelling. The main components for their framework consist of 4 components. First of all, the first component, named “Adapters”, take care of creating RDF graphs from multiple formats and data. This results in a linked data cloud, which makes up the second component and is used to represent virtual and actual entities. When the data is made readable by means of URIs, component three, called "support services", can simplify the process of processing linked data. This happens through four methods: entity management service, event processing, data catalogue and provenance service, and search & query services. Finally, the fourth component called, the applications, take care of processing all components 1-3 and present them in dashboards and scenario models to get a broader understanding of the energy consumption of the building, see Figure 8.

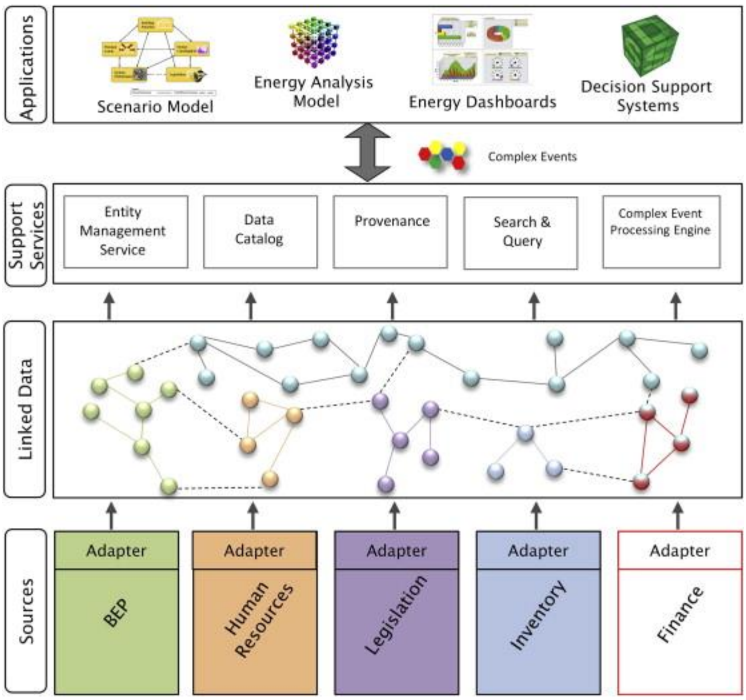


Figure 8 O'Donnell et al. (2013) 4 level framework

All these studies support the possibility of integrating BIM with asset management/FM to support the O&M phase and process the data using a semantic web approach. Yet, whereas Kim et al. (2018) explored the possibilities of integrating especially BIM data with historical maintenance records. In contrast, Hu et al. (2018), Shen et al. (2012) and Quinn et al. (2020), on the other hand, have particularly examined how they could integrate sensor data with BIM through IoT and BMS/BAS systems. Taking into account the common argument that within the asset management field, there is insufficient knowledge of an approach in the field of linked data and BIM, a better approach would therefore be to bring BIM closer to the asset management field by building an individual interface that asset manager can access. Therefore, it can be concluded that the studies of Howell, Rezgui & Beach (2018), Quinn et al. (2020) and Hu et al. (2018) are much more related to the research conducted within this thesis. Hu et al. (2018) focus on a web-based application that uses a BMS system as the main reference to present the condition of the building to an asset manager, so that he can quickly identify defects in the building's MEP system.

Based on the aforementioned findings, it can be assumed that the methodologies used in the studies conducted by Howell, Rezgui & Beach (2018), Hu et al. (2018), Quinn et al. (2020) and Chevallier et al. (2020), will result in a linkable BIM model that consists of static geometric building data on the one hand and dynamic sensor data on the other, and where semantically rich data can be queried for additional information. Central to all related studies is the examination of LBD in the context of building performance indicators such as energy or indoor environmental quality.

2.6 Integrating BIM with sensor Data

BIM is often used to reduce costs and construction time and to improve the productivity and quality of projects (Azhar et al., 2008). As shown in section 2.3, BIM is a robust information model that can be used to virtually represent a building throughout its life cycle (Valinejadshoubi, Moselhi, Bagchi & Salem 2020). During such a life cycle, all kinds of documents are digitized and added to the BIM database. According to Valinejadshoubi, Moselhi, Bagchi & Salem (2020), an accurate BIM information model is a virtual model that represents all building components in detail. Accordingly, these authors argue that the need for accurate BIM models is particularly important for facility managers and asset managers. They enable these users to efficiently monitor buildings by obtaining accurate data from the sensors and visualizing it. Although this desire to dynamically monitor BIM using sensors is there, according to Rio et al. (2013) it is still under development. Wang et al. (2013) have been able to show that the implication of BIM in monitoring systems for asset managers can lead to a much more effective decision-making process and thus allow the asset manager to identify intuitively more potential problems.

In recent years, there have been several attempts to integrate Revit BIM models with sensor data. For example, Wu & Liu (2020) have developed a visual BIM-based energy-saving system. In this system, IEQ and thermal comfort sensor measurements are integrated into the BIM model. On the other hand, Wehbe & Shahrour (2019) used a local database in which they stored sensor data and then linked it to the BIM model. Besides, Natephra & Motamedi (2019) developed a method to integrate sensor data for IEQ with BIM. Using a microcontroller and Dynamo, they captured sensor data and stored it in the BIM information model. Their framework succeeded in integrating geometric BIM data with environmental sensor data and thus measuring IEQ per location. However, Natephra & Motamedi (2019a) hypothesized that it would not be possible without a connecting factor between them. Contrary to all the above attempts, Valinejadshoubi, Moselhi, Bagchi & Salem (2020) designed a method to put together a system that integrates IoT devices into BIM to monitor the IEQ in buildings. They used a microcontroller and smart board to extract sensor data and transfer it to an SQL server embedded in their to be built system. Dynamo was then used to sort, read and transfer the data from the SQL to the BIM model. The framework proposed by Valinejadshoubi, Moselhi, Bagchi & Salem (2020) has been successfully implemented, but the research is limited to measuring thermal comfort and can still be extended to include sensor data on occupancy, lighting and energy. Out of all the above-mentioned studies, the study of Valinejadshoubi, Moselhi, Bagchi & Salem (2020) will be further elaborated.

Valinejadshoubi, Moselhi, Bagchi & Salem (2020) system consists mainly of three main components aimed at performing three specific tasks. Component 1 is the IoT system where a smartboard with a microcontroller is used to communicate the room's air temperature and humidity data. Here, the smartboard is connected to the temperature and humidity sensors to collect the thermal comfort data for an individually set interval. The microcontroller is used to host wireless communication protocols like Bluetooth and Wi-Fi and can receive and send data. The second component consists of developing a database in a MySQL environment, in which the captured sensor data is stored and updated on a regular basis. The microcontroller can be programmed and set to store the sensor data and time of measurement in this database. In their study, they constructed the database from a six-table schema to accommodate the corresponding parameters for temperature and humidity sensors. Finally, the third component is the BIM-based model of the building in which the virtual sensors are incorporated so that thermal comfort can be visualized and monitored. In order to link the physical sensor data stored in the MySQL database with the virtual sensors in the BIM model, nine modules are developed and coded in the software application Dynamo so that the temperature and humidity values stored in the database can be read, sorted, and updated in the BIM model (see Figure 9). Finally, optionally, it can be stored in a cloud by using a cloud-based exchange service application such as Konstru or Speckle.

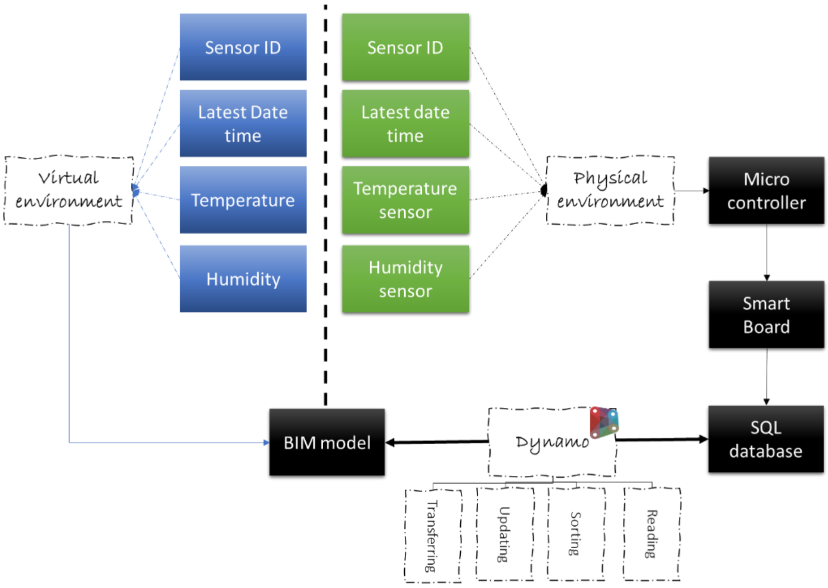


Figure 9 Simplified model of Valinejadshoubi, Moselhi, Bagchi & Salem (2020) framework

However, in contrast to the model shown in Figure 9, for this study, in addition to the humidity sensor and the temperature sensor, further IEQ sensors and motion sensors will be added so that occupancy and comfort parameters can be measured. To connect to the BIM model, it is first necessary to explain how Dynamo can be used to create modules for extracting, reading, and sorting sensor data. The connecting factor between the virtual environment and the physical environment is the sensor ID for each sensor. Valinejadshoubi, Moselhi, Bagchi & Salem (2020) state that the sensor data should be recorded in parameters with the same “sensor_ID” name for both the physical and virtual environment. For the physical environment, the sensor data should be captured in the SQL using a schema and tables containing the following column parameters; “Sensor_ID”, “Record_ID”, “Sensor_Value”, “Recorded_at”. With dynamo, these tables can then be used to transform the “sensor_value” data for the virtual environment into measurements such as; "Temperature", "Sensor_ID”, “Temperature Level", "Humidity Level", "Humidity", "Lux", Occupancy", "LatestDateTime”.

2.6.1 Extracting parameters from the BIM Model

Dynamo works with modules and needs the following to extract the virtual sensor parameters from the BIM model. The module consists of 4 parts, built up from reading out the 'Network Sensor' data from the temperature sensor and virtual humidity sensor. Next, all relevant components in this category are selected and displayed according to the user-defined sensor parameters, such as the sensor ID, temperature, humidity, latest date time and finally extracted from the BIM model and presented in a list (Valinejadshoubi, Moselhi, Bagchi & Salem 2020).

2.6.2 Connecting the BIM model to the database

For the connection to the database, a link is made after extracting the values from the BIM model to the predefined database parameters in MySQL. For example, the sensor ID from the BIM model is linked to the sensor ID from the database. Parameters like the temperature value from the BIM model are linked to the sensor value from the database, and finally, the latest date time and virtual temperature parameter sensor from the BIM model are linked to the recorded air temperature parameter in the database (Valinejadshoubi, Moselhi, Bagchi & Salem 2020).

2.6.3 MySQL or NoSQL database

MySQL is a relational database management system that allows a user to store, modify and retrieve data based on a few queries and commands such as SELECT, INSERT and UPDATE (Fatima & Wasnik, 2016). Besides relational databases, there are also non-relational databases such as a NoSQL database (Jose & Abraham, 2020). Databases such as MySQL store data in an organized manner. The issue here is that as the years went by, newer software releases such as NoSQL databases slowly began to replace MySQL databases. NoSQL databases were, due to their new software techniques, much better able to handle frequently updated data in larger volumes (Jose & Abraham, 2020). An example of this can be found in the NoSQL database MongoDB, that was developed with the intention of ensuring that large amounts of data are processed efficiently. MongoDB, for example, uses JSON (JavaScript object notation) documents and collections instead of tables (Jose & Abraham, 2020). The general fundamental characteristics of a NoSQL database are that, as opposed to relational databases, it is schema-less, and each individual server has its own local storage space, making it dynamically expandable (Fatima & Wasnik, 2016). In addition, Kepner et al. (2016) state that given their diverse structures, each database is powerful for performing a specific set of tasks. For example, according to (Kepner et al., 2016), a NoSQL database may be well suited to performing Internet searches, and a MySQL is better suited to performing transactional tasks.

2.7 Building performance

There are many publications that define the term "building performance”. Building performance is the concept that aims to optimise energy use and the indoor environment in both existing, as well as, new buildings. In addition, it spans across different phases of a building's life cycle in order to deliver more environmentally sustainable, health-friendly, and more productive and better comfort-oriented buildings for the end-user[13]. However, de Wilde (2019) states that the term "building performance" essentially stands for a concept that makes it possible to quantify how well a building suits its own functions. The guiding principle is that performance is related to the building fulfilling its function through behaviour. This makes static building characteristics (e.g. floor area, building volume) not subject to the term because they do not need to perform a reaction to fulfil their function.

Thorough research of the term building performance in relation to asset management shows that the term building performance can be broadened. According to Preiser and Naser (2008), building performance is a process that affects the overall value of a building by measuring how buildings perform in relation to their users. Through its influence on the value of the building, it is thus directly related to asset management. According to Preiser & Vischer (2005), the

[13] Building Performance (tue.nl)

relevance of building performance in asset management is mainly to improve the quality of the building. Then & Tan (2002), on the other hand, break down the concept of building performance in relation to asset management into three phases that are in a loop with each other, consisting of management requirements, asset performance and business drivers. Then & Tan (2002), state that building performance has a strong integration with KPIs that lead to the improvement of the business drivers, which in turn ensure that the asset performance is aligned with the management requirements (business need for building services). As highlighted by Then & Tan (2002), this overall approach bears a profound relationship to the core of this research.

2.7.1 Building performance KPIs

Nevertheless, an asset manager needs to focus on measuring these KPIs as pointed out by Then & Tan (2002). A study carried out by de Wilde (2019) shows that a total of 83 aspects can be identified to measure whether a building fulfils its function.[14] This list was then cross-referenced with other studies to check which aspects were most frequently associated with building performance (see appendix B). A central part of this is improving the overall energy performance and improving indoor environmental quality. Thus, in this thesis, building performance refers to the building's energy and comfort performance. This is mostly because buildings are inefficient consumers of resources. Buildings and the construction sector together account for 36% of global energy consumption and 40% of total CO2 emissions (Maslesa & Jensen & Birkved, 2018). According to Li et al. (2020), by quantifying building performance and focusing on improving energy consumption, as much as 36% of energy consumption can be saved on average (Li et al., 2019). Therefore, by addressing the energy and IEQ aspects in buildings, the highest and most cost-effective long-term CO2 reduction gains can be achieved (Jain et al., 2020). The building energy consumers are all the building systems that lead to increased or decreased use of energy, expressed in kWh. This includes lighting systems, HVAC, cooling/heating components, and power sockets (Li et al., 2020). More efficient energy policies have meant that air can't just flow in and out unchecked, resulting in an increased risk to the IEQ (Demanega et al., 2021; Kallio et al., 2020). IEQ is the concept of all the interactions taking place between physical, chemical, and biological elements indoors, which includes elements such as the overall air quality, the indoor air temperature, relative humidity, lighting strength and noise levels, which could all have an impact on the quality of life (Kallio et al. 2020). Systems that fall under IEQ are, according to Demanega et al. (2021), the heating and the HVAC systems that ensure that the air temperature (°C), humidity levels (%), ventilation speed and air distribution are maintained.

2.7.2 Measuring Building performance

In order to measure these systems and thus the building performance, it is first necessary to find out what methodologies exist for measuring building performance. According to de Wilde (2019), this can be done in four ways. First, through an experiment in the physical world (measurement), in a virtual environment (through simulations) or in the human mind (expert judgement), and finally, stakeholder surveying (evaluation). Experiments study how the system input, output and system state fluctuate. This is done through observation and can, in most cases, be quantified by measuring the as-is state of the building. Simulations measure, for example, the virtual energy performance or daylight penetration of the building; nowadays, this is also done using BIM simulation models. However, Jain et al. (2020) argue that this methodology of simulating buildings during a design phase leads to an undervaluation of building performance, also called the "performance gap", when a building is benchmarked against a baseline value. In fact, their analysis shows that 75% of buildings come out as "underperforming" by using this methodology. These findings are also supported by Homaei & Hamdy (2020) as well as Geraldi & Ghisi (2020), who state that designers often estimate how a building should perform. However, their estimations tend to deviate from the actual energy consumptions. Due to uncertain changes in the building's environment, such as deviating occupant behaviour, climate changes, or economic factors. Expert judgement usually happens through several years of work-related experience and stakeholder surveying results from asking asset manager about their expert judgement.

[14] Overview of Building Performance Aspects and Functions (bldg-perf.org)

Yet, Omar et al. (2017) argue that any of these four forms of measurement should be accompanied using KPIs because they are fundamental principles for effectively measuring maintenance management factors. That's because the assessment of building performance must be done primarily in quantifiable indicators in order to retrace how the performance was compared to the previous year. The problem with this approach, however, is with identifying the right KPIs. There are many KPIs available that range from measuring individual components (e.g. HVAC system, light fixtures, lighting systems), entire systems (e.g. lighting, heating, ventilation, etc.), or entire buildings (Li et al. 2020). In this study, reference is made to the system level where all components are included over a specific floor area. According to Li et al. (2020), energy usage related KPIs measure how efficiently building systems/components provide their services in relation to the amount of energy they consume. Abu Bakar et al. (2015) state that the most commonly used model with the least data input required is the energy efficiency index model, where the kWh is calculated over the m^2 gross floor area.

$$EEI = \frac{energy\ consumption\ (kWh)}{floor\ area(m2)}$$

The energy efficiency index (EEI) represents a ratio of delivered energy to consumed energy. These findings are also supported by Jain et al. (2020) & Li et al. (2020). Jain et al. (2020) use only quantifiable indicators in their study. For the energy performance indicators, $kWh/m^2/year$ is used as the main unit of measurement. Abu Bakar et al. (2015) adds that as baseline values, the energy consumption per m^2 should lie between 200-250 kWh/m^2. In addition to the energy efficiency, it is also possible to use the energy use intensity (EUI) indicator, which measures the cumulative energy consumption as a calculable value over a standard factor (e.g. annual lighting energy consumption over the floor area in m^2). According to Li et al. (2020), this standard factor can differ per KPI and depends on the system. For example, lighting is calculated over the total m^2 floor area, while for cool systems, the conditioned floor area is used instead. This makes it more difficult to estimate energy consumption with this indicator.

The indoor environmental quality performance indicators can be measured in different units depending on the connected system. According to Barthelmes et al. (2019), achieving the right energy efficiency is not only a factor of technical and design solutions. It also depends primarily on the manner in which occupants decide to apply their comfort criteria, and also on how their energy and environmental lifestyles are structured. This means that the way spaces and energy are used depends mainly on how occupants perceive their comfort and how to improve the user perspective of indoor environmental quality and comfort. To decompose this requires breaking down indoor environmental quality into measurable units. These findings are also supported by Nimlyat & Kandar (2015), who state that indoor environmental performance must guarantee at least 80% occupant satisfaction in order to be considered "well-performing". The disadvantage of this measurement is that physical respondents must be asked how satisfied they are and measuring this kind of subjective metric is too time-consuming. It is, therefore, more important to link benchmarks to individual components of IEQ, such as according to ANSI/ASHRAE 55-2017 standard (ASHRAE, 2017), the average temperature should be around 20 - 27 degrees Celsius in order to be considered "comfortable". The comfort measurement of temperature within the building depends on whether the user is able to control the airspeed. If the user is unable to do so at temperatures less than 23 degrees, a default airspeed of 0.20 m/s should be used. At temperatures between 23 and 25.5 degrees, a calculation should be made using the equation down below. Otherwise, the default values should be used.

$$V_a = 50.49 - 4.4047(t_0) + 0.096425(t_0)^2\ (m/s,\ °C),$$

At temperatures >25.5 degrees, a default airspeed of 0.8 m/s should be used. Aside from the temperature and airspeed, humidity is measured in percentages, and air quality, on the other hand, is measured in particle matter (PM) concentration expressed in (µg/m3) and particle size (µm). Nimlyat & Kandar (2015) state that relative humidity is important for comfort because a humidity that is too low exposes the human body to skin dryness and irritation. At an air humidity of <40%, occupants may get a dry throat, and at <20% may affect the blinking movement of the human eye. According to Akanmu et al. (2021), at temperatures between 20 - 26 degrees Celsius, no psychological effects can be observed at high humidity levels of between 60-90%. On the other hand, Korsavi et al. (2020), who conducted research into indoor environmental quality levels at school buildings, suggest as a benchmark for the humidity that the average value should be around 50.9% in warm seasons and around 37.3% in cold seasons to be considered comfortable. These authors also state that for the colder seasons, the average temperature should be around 24.2 and the airspeed around 0.1 m/s. Furthermore, they state that the Co2 ppm should be 1180 to be considered comfortable. At values of 1500 ppm, the air can become fatal to occupants (Nimlyat & Kandar, 2015). Also, the perceived light levels (lux) were set at 467 in the warmer seasons. In the colder season, all these values lie at the following values: temperature (22.8 degrees), lux (527), air velocity (0.06) and CO2 levels at 1310 PPM. Occupancy rate tracks the number of people present in the building and is usually derived from sensors that detect movement and is expressed as number of people per m² (Li et al. 2018). The temperature sensors measure deviations in the heat temperature and are used to control the air conditioning. Excessive humidity inside the building leads to unnecessary condensation, which damages equipment. For monitoring the climate conditions and managing the HVAC systems, a BMS is used as discussed in section 2.2.1. A BMS is also capable of providing temperature and relative humidity data for analysis.

Central to measuring these values are the connected systems. The advantage of a sensor is that it produces large amounts of real-time data and is a huge source of data and information. The potential for integration with web-based tools is therefore considerable. However, Phuoc & Hauswirth (2009) recognize that many of these data sources remain excluded for the time being within their own application structures, making them hardly accessible and difficult to link to other data sources. In order to make the sensor data and its elements web ready, Phuoc & Hauswirth (2009) suggest that that the sensor data should become semantically rich by giving them a triple store notation and save them in the triple store database. After considering all the findings mentioned above, it can be concluded that there are several ways to design a system that integrates BMS sensor data with static building data, see Figure 10. In general, three main integration methods stand out the most. Firstly, there is an integration using an SQL database in which the sensor data is stored via the sensor ID. The way this works is that the sensor data is collected from the asset information management systems and stored in a SQL database, and through an intermediary software such as Dynamo it is then connected to a BIM model. This allows one to create a digital twin, as pointed out by Valinejadshoubi, Moselhi, Bagchi & Salem (2020). A second integration is by using a combination of SQL databases and semantic web technologies as this has been done by Quinn et al. (2020) and Kim et al. (2018). In which they used an interface to retrieve the sensor data from an SQL database and used a self-written ontology, such as the FM-BIM ontology to not only connect it to the static building data but also to retrieve the necessary information from the sensor data. Finally, the third integration method relies exclusively on semantic web technologies to create linked building data. This method uses ontologies to define the sensor data and describes their underlying relationship to the building. As indicated by Malcom, Werbrouck & Pauwels (2020), a common factor, such as the guid, is needed to find the necessary information with the correct accuracy.

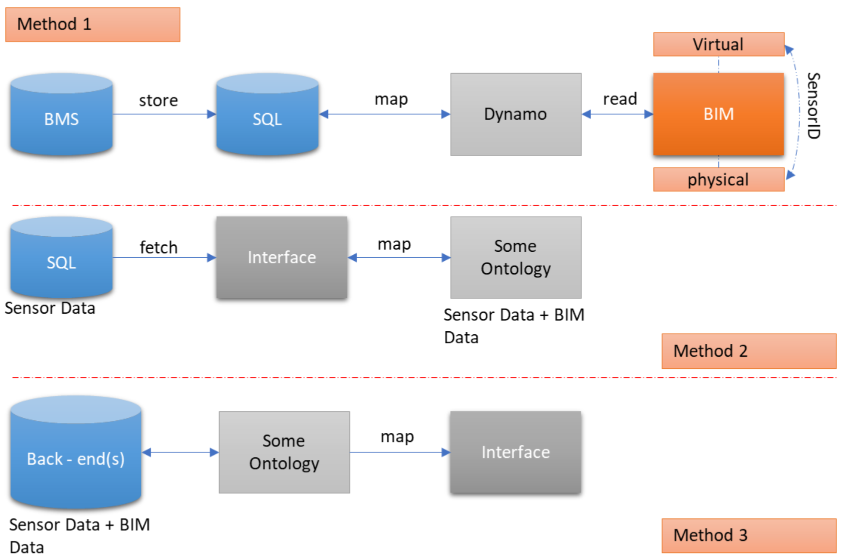


Figure 10 Various integration methods

2.8 Web Applications

2.8.1 Web application architecture

A basic web app structure consists of a browser with a front end, a network, and a web server back end (Maes et al., 2016). In essence, a web browser uses an application programming interface (API) to request web pages from a web server, which often consist of HTML[15], CSS[16] and JavaScript[17] files and renders them into the browser. These web servers can contain databases that handle information between different web systems, such as user data. The front end is also called the client-side and consists of a user interface. On the other hand, the back end communicates with other applications through APIs and uses their endpoints to exchange information (Heilman & Francis, 2007). These exchange formats between APIs are usually JSON files containing object information.

HTTP is an application-level protocol that allows machines to send messages to other devices in a request and response manner. Both request and response messages have a common structure that includes a start line, header(s) and a content body. In a request message, the start line defines the HTTP method of requesting resources from a server. The identification of resources is made using a URI. A REST API[18] allows an HTTP protocol to POST, GET, PUT, UPDATE, or DELETE resources (Rodrigez, 2015). REST APIs are convenient for applications to use as a means of communication with various databases.

Programming languages and frameworks are required for the development of web applications. The most used programming languages in 2021 are Java, JavaScript, Typescript, Python, Ruby, PHP, and Swift. At the same time, for back-end development, Node.JS, Express, Ruby on Rails, Django, and Flask are well-known frameworks that are used most often. According to Stack Overflow[19], the most popular libraries/frameworks for front-end development are React.Js, Vue.Js and Angular. There are not a lot of differences between the different front-end frameworks. React, Vue and Angular are component driven frameworks. Angular's components are known as Directives. Directives are markers on the document object model (DOM[20]) elements, to which Angular can add specific behaviour. The DOM is an interface for HTML documents, and it represents the web page so that programs can adapt the document (web page) in structure, style and content. The page content stored in the DOM can thus be modified by applications using JavaScript. What sets Angular apart from React is that Angular separates the UI part of components into attributes from HTML and the "behaviour" from JavaScript. React, on the other hand, combines the UI and "behaviour" of components. Vue is similar in the way it handles components to React, and it differs in the functionality of the styling. Instead of CSS, Vue uses pre-processors and handles the integration of other libraries like Bootstrap better than React.

In terms of learning, Angular is the hardest to learn, as knowledge of Typescript and model view controller (MVC[21]) is required to develop applications in Angular. On the other hand, React is easier to learn but is less complete than Angular and thus needs help from third-party libraries for developing the app. Vue is the easiest to learn because it is easy to customize and has an overlap with Angular and React in the way they handle components. The downside, however, is that this flexibility of Vue is prone to poorly performing code and thus can lead to problems in the long run[22]. Based on the above reasoning, the application will be built in React.Js because this framework is the most robust of all three developments and has the largest community for support. In addition, a lot of code is already written in React, which makes it easy to integrate certain components during the implementation phase.

[15] https://www.w3schools.com/html/html_intro.asp
[16] https://www.w3schools.com/css/css_intro.asp
[17] https://www.w3schools.com/js/js_intro.asp
[18] https://www.redhat.com/en/topics/api/what-is-a-rest-api
[19] Stack Overflow Developer Survey 2020
[20] https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
[21] https://docs.microsoft.com/nl-nl/aspnet/core/mvc/overview?view=aspnetcore-5.0
[22] Angular vs React vs Vue: Which Framework to Choose in 2021 (codeinwp.com)

2.8.2 GRAPHDB integration with NoSQL and the LBD Server

Central to this study is the development of a web-based BIM application that uses a linked data approach. For this purpose, using a NoSQL Graph database, also called ontotext GraphDB, is essential. This GraphDB is used to store the turtle files generated by the IFCtoLBD tool (Bonduel et al., 2018). GraphDB works with a network of nodes that have underlying relationships with each other and come with unique identifiers in which data is stored, thus extending the functionality of current data solutions (Fatima & Wasnik, 2016). This RDF database stores data in triple stores and retrieves them with semantic queries. A triplestore object consists of a subject-predicate-object structure, e.g. "Space_1(subject) rdf:type(predicate) bot:Space (object)". The advantage of GraphDB is that it is free for use, just like the IFCtoLBD tool, and it is able to access the underlying data using SPARQL queries and then communicate with the user interface using APIs. This is already partially done with the LBDServer developed by Malcolm, Pauwels & Werbrouck (2020). They used two NoSQL databases (GraphDB and MongoDB) for the LBDServer backend while displaying the output using React.JS and Typescript in the web browser.

The LBDServer is designed to bring linked data technologies and BIM much closer to the end-user, as it enables relevant stakeholders to manipulate linked building data from their web browser. In essence, the LBDServer allows users to enrich LBD with SWT ontologies, thus improving the quality of the data (Malcolm, Werbrouck & Pauwels, 2020). The advantage of this is that from the LBD nodes, larger networks of data can be created that are applicable to different industries and disciplines (Malcolm, Wernbrouck & Pauwels, 2020).

The LBDServer uses JavaScript/Typescript syntax to connect a web UI to multiple databases that each have their own purpose for dealing with data (RDF triples, geometry, and images). Using a glTF format file, the tool can present geometric BIM data (Malcolm, Wernbrouck & Pauwels, 2020). The principle is to use IFC to create an export in COLLADA. COLLADA[23] is a standard XML schema whose purpose is to easily migrate 3D files between applications. The LBDServer tool uses IFCconvert to convert the IFC file into COLLADA (KhronosGroup 2019[24]). IFCconvert is a property of IFCOpenshell. This COLLADA file acts as an intermediate to finally be converted using COLLADA2GLTF into a GL transmission format (glTF[25])(KhronosGroup 2019). Such a glTF file is constructed in a JSON format that divides the 3D model into mesh objects so that the content can be presented in a web-friendly manner and has as property, various properties, a mix of geometric and texture asset data, and matrix locations (KhronosGroup 2019). To read the JSON file, the LBDServer uses Xeogl[26], an open-source library, to visualize 3D models (Malcolm, Wernbrouck & Pauwels, 2020). All geometric data is linked to all relevant mesh objects in the glTF files through their IFC GlobalID. Similarly, to how sensor data is given a sensor ID to connect physical and virtual sensor objects. Thus, a connection through this GlobalId can be searched for with the geometry in the Xeogl viewer and the IFC schema (Malcolm, Wernbrouck & Pauwels, 2020). However, these authors argue that it is important to preferably use ontologies such as bot:hasComplexGeometry / omg:hasGeometry between geometric data and semantic data so that it can be connected to the underlying RDF graph triplestore.

2.8.3 Integration of sensor data with web-based BIM

The current LBDServer is capable of displaying BIM models in a web browser and of querying and reading data with the help of a SPARQL query. It is from this server that an integration with the building performance KPIs will be sought. The integration needed being mainly on the visual and virtual representation of sensors data in the BIM model and linking this sensor data simultaneously semantically on the back end to the MongoDB database. In this way, a link can be made from MongoDB to the actual physical sensors because equivalent parameters such as sensor ID and sensor values are converted into temperature, energy consumption and humidity. Information can then be displayed in objects on the web using a SPARQL query and APIs. Essential is to make the sensor data semantically

[23] https://www.khronos.org/collada/
[24] https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/figures/gltfOverview-2.0.0b.png
[25] https://www.khronos.org/gltf/
[26] https://xeogl.org/

rich. To this end, the sensors must be described using a combination of ontologies, such as the BOT and PROPS ontologies. This makes it possible not only to describe the sensors but also to add properties to them. This, in turn, allows for querying the graphDB database in which this semantically rich data is stored by means of a SPARQL query in the browser. This can be done by using the IFCtoLBD tool in combination with the LBDServer that transforms the file into a TTL to be used in GraphDB.

2.8.4 Interlinking the findings

All in all, the various findings in the literature can be brought together to create the following model (see Figure 11).

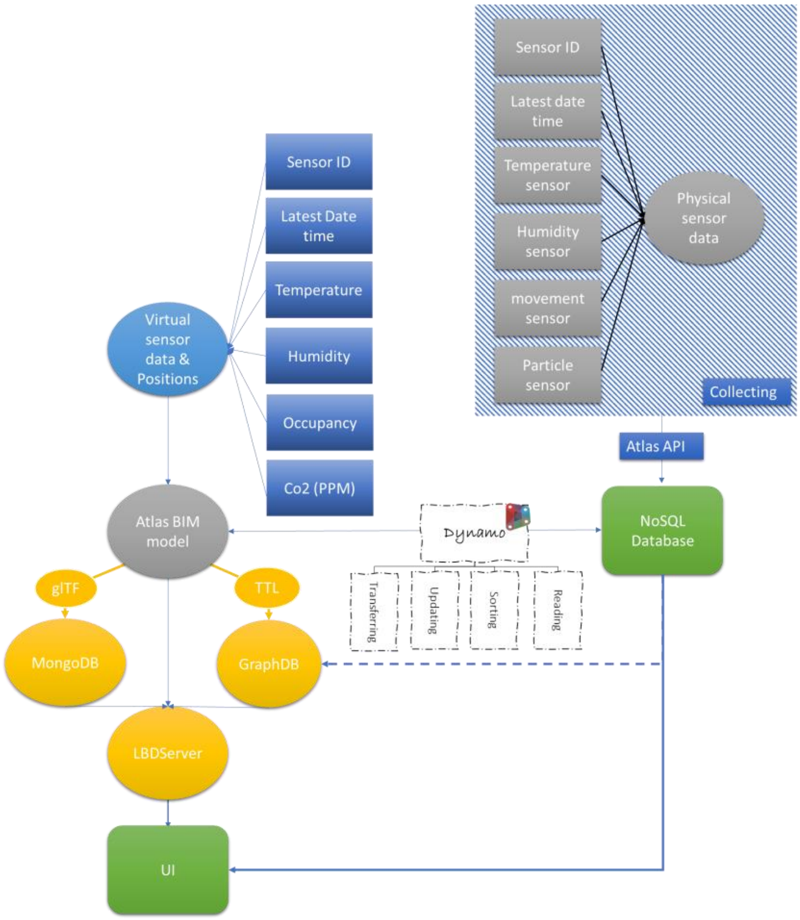


Figure 11 Interlinking the findings in a model

The model is an enhancement to the model of Valinejadshoubi, Moselhi, Bagchi & Salem (2020), but differs from it in several ways. First, the MySQL database is replaced by a NoSQL database, in this case consisting of MongoDB, because it has the advantage of accommodating more volumes of time-series data. In addition, this model also includes the components from the LBDServer, which will be reused for building the user interface. Furthermore, various sensor data is collected from the Atlas living lab server and stored in the NoSQL database, especially the sensor ID and timestamp are important to create the digital twin. This NoSQL database is connected to the BIM model in the back end, as the glTF file containing the static building data is also stored in MongoDB. The sensor ID is added in the glTF file as well as in the TTL file so that a connection can be made between the BMS dynamic sensor data and the BIM model. The user interface should handle this without any action required from the asset manager. These sensors are made semantically rich according to the BOT and PROPS ontologies. The properties of the sensors are described by means of the PROPS ontology, and these are connected to the relevant space to which they belong by means of the BOT ontology.

2.9 Literature conclusion

In this literature review, extensive research was conducted on the topics of asset management, value delivery, asset management IT systems & BIM, semantic web technologies, linked data and finally, building performance and web applications. The purpose of this literature review is to provide an answer to the first five research questions.

- Q1 What is Asset Management, and how is it currently related to BIM environments.

First, an attempt was made to decompose the term asset management within BIM environments. The literature review revealed that the term asset management can be described as a joint effort of an organisation to realise/recover value from an asset. Whereby value realisation occurs when a balance is sought between costs, risks, opportunities and increasing the building performance. The realisation of value refers, with regard to this research, to a building that fulfils its functional value. The functional value of a building depends on the property owner but can be categorised as economic, cultural, operational, symbolic, and social value. Asset management is, after all, a web of activities in which the asset manager searches for a connection between the property and the organisational objectives of the property owner. These organisational goals can vary from achieving sustainability goals to reducing costs. Regardless of which objective an organisation attaches to an asset, an asset manager usually aims to improve the operational functionality of the building by optimising its performance. At present, BIM is hardly used within asset management, as it is considered more suited for the design and construction than the O&M phase. In recent years, by using asset information management, more attempts are being made to integrate BIM with asset management, especially with the tools that are primarily aimed at monitoring and managing the building.

- Q2 How is building data managed within and outside a BIM model (cfr. Digital Twin)

Several asset management systems can support this goal, including CMMS, EAMS, BMS, and BAS. However, these systems have their shortcomings. In the case of CMMS, high-quality data was lacking within these systems. BMS and BAS solve this to a certain extent. A BMS is a computer system installed in buildings to primarily control and monitor mechanical and electrical systems. In comparison, a BAS is an intelligent system comprising both hardware and software. It allows systems including HVAC, lighting, security, and other systems to connect and communicate with each other on a single platform. BMS and BAS systems are good for monitoring and collecting data around building systems. While these systems are all very good at solving problems seen in CMMS and EAMS systems, they are not without their own shortcomings. Each system comes from a vendor with its own data structure, which hinders integration between the different systems often found in buildings. In addition, they often require manual human input. This leads to human error margins. Furthermore, they do not have 3D interactive models that can be used to address specific sensors and/or defects. The concept of BIM solves the above problems to some extent. The use of a BIM methodology leads to higher use of IT in the architecture, engineering, construction and operation (AECO) industry and increases the coordination of design activities, integration of different systems, and the transfer of building information. BIM makes it possible to significantly reduce the time spent on finding semantic O&M data together with the geographical data for the material and building components. However, BIM is not equally suited to the monitoring and measurement of building performance aspects. A combination of BIM and BMS sensor data is, therefore, necessary to solve the above problems.

- Q3 What is IEQ, occupancy & energy consumption, and how can building performance in terms of energy use be quantified and evaluated.

The sensor data found in BMS systems represents output values that are related to the building performance. Building performance is the concept that aims to optimise energy use and the indoor environmental climate in existing, as well as, in new buildings and across different phases of a building's construction, in order to deliver more environmentally sustainable, health-friendly, and more productive and better comfort-oriented buildings for the end-user. The term is used to relate several concepts to each other. However, the focus of this term in relation to asset management is on the performance of the building in relation to energy use and comfort. Asset managers strive for the satisfaction of end-users as a result of good asset management. This is mainly achieved when the user feels most comfortable. A focus will therefore be placed on the indoor environmental quality. Indoor environmental quality consists of measurable values such as CO2, temperature, airspeed, and humidity. The values to be aimed for depend on the season and can fluctuate between 1180 and 1300 PPM for the CO2 values, between 0.1 and 0.3 m/s for the airspeed. The temperature should be between 22.8 and 24.2 °C, and the air humidity between 37.3% for cold seasons and 50.9% for warm seasons. Too high or too low below these values and the comfort of the occupants may be perceived as uncomfortable. In addition, occupancy plays a major role in the utilisation of the building. For this reason, it is important to include these two (IEQ & occupancy) aspects in the development of the web application.

- Q4 Which underlying relationships between BIM and building performance can be established?

BIM is commonly used to reduce costs and construction time and to increase the productivity and quality of projects. In addition, the literature study showed that BIM is a robust information model that can be used to virtually represent the building throughout its life cycle. An accurate BIM information model thus enables asset managers to efficiently manage the building throughout its life cycle using sensor data. However, this development in which BIM models are linked to sensor data is still in the development phase. There have been several recent attempts to integrate BIM with sensor data. The most common integration is by linking the physical sensors with their properties using their sensor IDs to the virtual BIM model in which the sensors are also modelled. The virtual sensor is assigned the same sensor ID as the physical sensor. This is only possible when the physical sensor data is stored in a SQL database that is then read, sorted, updated, and transferred to the BIM model through an intermediary programme (e.g. Dynamo). To create a digital twin, a number of parameters are needed in addition to the sensor ID. It is important to register the timestamp of the measurement as well as the measurable unit, which differs per sensor. This can be noted as degrees Celsius, humidity in %, or the number of people per m² and must also be included in the database. This database can be built from a MySQL database as Valinejadshoubi, Moselhi, Bagchi & Salem (2020) have done, but also from a NoSQL database such as MongoDB. A NoSQL database has the advantage over a MySQL database in that it can handle a much larger volume of data. This is especially the case with time-series data that is continuously updated at different intervals. Once this live connection is made between the physical and virtual world, the BIM model is enriched with the sensor values and can now be seen as a digital twin.

- Q5 How can static building data (geometry, element types, properties) be combined with real-time building data (sensor readings, space use, etc.) to form a digital twin of the building?

To solve problems found in both BIM and BMS sensor data, a solution has to be found in combining these two systems. However, this leads to a number of problems where dynamic sensor data can hardly be linked to static spatial data. In this context, linked data can act as a connector, linking dynamic time-series data to these static geometric data through the use of ontologies. Linked data refers to a collection of methods for publishing structured data on the world wide web. This ranges from heterogeneous systems within a single organisation to two individual databases spread across two different locations and organisations. Linked Data uses RDF to make statements that link random objects with the world. RDF is a suitable format for publishing data on the web so that other applications can use or reuse semantically rich information in accordance with their own needs (Berners-lee, 2004). Linked building data subsequently is the result of converting and structuring building data into a set of RDF graphs with the use of semantic web technologies. This is done using a set of vocabularies. These ontologies describe and define concepts and their underlying relationships. The advantage of using ontologies is that a connection can be made between semantically rich geometric data and dynamic time-series sensor data. There are various ontologies available, each intended to describe certain components. For example, the BOT ontology describes the core topological concepts of the building, while the PROPS ontology describes the properties of the building and sensors. The SSN ontology is developed specifically for describing sensors. It is also possible to combine different ontologies to describe building elements and sensors. This is often done in practice by different researchers to describe BMS sensor data with BIM data. Although the SSN ontology is more comprehensive to describe the sensors, it is also an approach that is not only intended for the built environment. As a result, custom links need to be created for the SSN ontology. This means that the SSN ontology approach requires a large amount of semantic data to be defined manually using scripts for each point in the BAS and BMS systems. Because of this, it is convenient to use the PROPS ontology and BOT ontology to connect the sensors to the BIM model. Using the BOT ontology, spaces can be defined, and the sensors can be placed in the spaces using the bot:containsElement attribute. In combination with the PROPS ontology that defines and inserts the properties of the sensors, a link can be formed between the BMS sensor data and the BIM model. This connection is achieved by using the bot:containsElement and bot:hasGuid attributes. In addition to the bot:containsElement, which is used to indicate which sensors can be found in each space, the bot:hasGuid can be used to establish a link between the same guid found in the BIM model and the guid found at bot:hasGuid in the semantic rich turtle file.

3 Methodology

In this study, quantitative and qualitative research is carried out to answer the question as to how a web-based BIM application can enable asset managers to optimize the building performance of utility buildings. Subsequently, with the help of React.JS, semantic web technologies, the LBDServer and NoSQL, the linked building data asset management dashboard was built at the front and back end.

3.1 Data collection

For the literature review, a review was first made of the term's asset management, value creation, and what tasks asset managers perform. Subsequently, previous studies in the field of the semantic web from 2010 to 2020 were examined in order to analyse how the semantic web works and how linked building data can be created. This revealed several possibilities for creating a digital twin. First of all, the study by Quinn et al. (2020) shows that by using a combination of ontologies and SQL, an integration between the time series sensor data and their ID could be realised, which required Dynamo to store and map the sensor data. In addition, an integration based purely on ontologies is also possible, in which the sensors and their spaces are described and then queried for semantic information. Finally, a direct connection can be realised in which the sensor data is integrated directly into MongoDB where all building data is also stored as was done by Valinejadshoubi, Moselhi, Bagchi & Salem (2020).

With an aim to identify what had to be improved, the building performance indicators were then analysed based on the research conducted by author Nimlyat & Kandar (2015), Korsavi, Montazami & Mumovic (2020), and de Wilde (2019). Finally, various frameworks and methodologies were examined, in which a combination of BIM, sensor data, the semantic web was used to examine how the web application should be built. Initially, it was intended to use the framework of Valinejadshoubi, Moselhi, Bagchi & Salem (2020), as they chose a structure that focused on similar building performance indicators and on utility buildings. However, this kind of integration is difficult to achieve without an API. For this reason, an integration based on the BOT and PROPS ontologies was chosen to develop the application. For all literature, scientific articles were searched on science direct and the TU/e Library.

The architectural requirements for the web application were drawn up based on this literature study. Because asset managers will use the web application, five semi-structured online interviews were conducted with several asset managers, including the asset manager of the TU/e and four external asset managers. This resulted in the user requirements for the user interface. The interviews lasted 30 minutes on average and consisted of 16 questions focused on six main topics, namely, asset management, current asset management IT systems, linked data, decision support, building performance, and sensor data management. The interviews were semi-structured because this allows for a structure within the interviews to some degree, while keeping the possibilities open for new ideas. Since only the user requirements were taken from the interviews, a summary of the interviews is added in section 4.1. During the interview's notes were made, which are added in appendix D.

In addition to the literature research, open-source and free to use data resources were also used in this research. For Atlas, a sensor dataset from the living lab was used, and a BIM model for the eighth and ninth floor was provided by a former student in Revit. The sensor data is retrieved from the living lab server as a historical dump of data in an excel format and contains the following sensor parameters; HVAC related sensor values such as CO2 levels, temperature, airflow and airspeed, and registered movements expressed in Boolean values that were counted, the applied framework is shown in Figure 12.

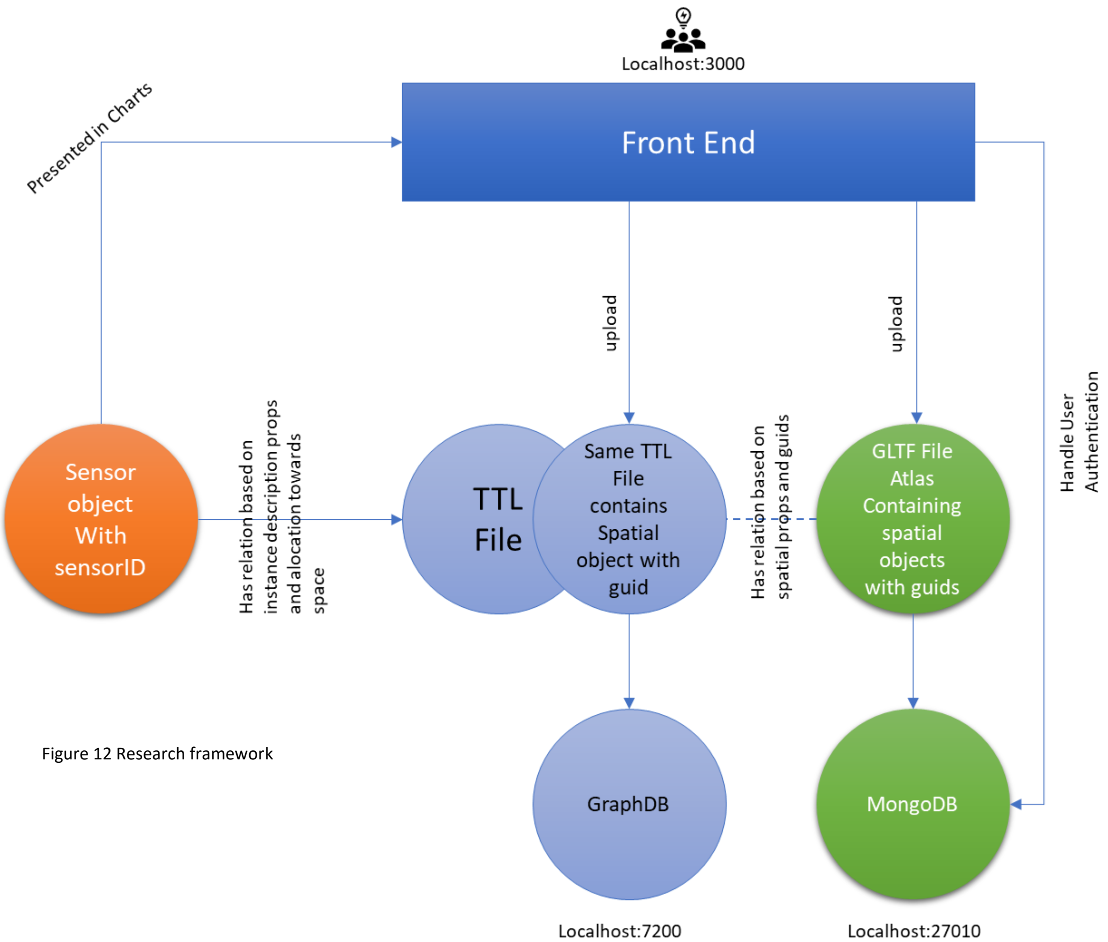


Figure 12 Research framework

3.2 Research

First, a global framework was set up that could be presented to the asset managers during the interviews; this UI framework is made with Figma[27], a free to use web tool. Subsequently, the semi-structured interviews were then conducted, after which the user requirements could supplement the final UI. The interviews lasted 20 - 30 minutes in total and were conducted online through Teams. First of all, questions were asked on the topic of asset management to gain insight into what asset management and value delivery means to each asset manager. Second, questions were asked about current asset management IT systems and asset information management to gain insights into which limitations they see within their current ways of working. Third, questions were asked about sensor data management, with the aim of gaining insight into how sensor data is collected and managed independently of the literature studied. Subsequently, questions on linked data were asked to see which use cases they would find for linked data within asset management. Finally, questions were asked about decisions and KPIs, and how decisions were made based on the input the sensor data would give, and which KPIs they used, what the benchmark values were and how they were managed in practices. All this, in order to get a better insight into the possibilities for improving the existing services of asset managers, the approach was to incorporate the user requirements in the web application in an intuitive way. During these interviews, notes were made, which were used as a user requirement list for the framework. These were then documented in an overview table. All requirements from the interviews have been combined with the requirements from the literature.

[27] https://www.figma.com/

A component framework was drawn up from this table, explaining how to deal with the various components and data flows. Then the framework for the web application was completed, which was created using the software draw.io[28]. For a period of 1-2 months from January to February 28th, 2021, sensor data was recorded and measured for the 8th floor of Atlas. The sensor data was recorded, processed in a JSON format, and stored locally during this research. This sensor data was linked through the sensor ID and their location to the spaces in the BIM model. Afterwards, by re-using the pre-written code from the LBDServer, the BIM viewer component was re-used in the new UI along with the critical sensor data from the sensors. Two documents have been made available for this purpose the BIM model in both IFC and glTF format and the LBD turtle file of Atlas. During the research, both formats were not suitable for use. First of all, the spaces in the original glTF file were not included. Therefore, the file had to be re-generated. To achieve this, use was made of the tools blenderBIM and IFCtoLBD, more on this in section 6.1.1.

3.3 Atlas living lab and handling the sensor data

During the study, extensive research was carried out within Atlas over a two-month period into how the user interface works and whether the sensors are linked properly, and whether it displays the correct values and information on the web application. For the case study, the Atlas living lab server was used. Atlas was renovated in 2018 and officially put into operation on the 21st of March 2019. The entire objective of the renovation was to deliver the most sustainable educational building in the world. Atlas has thus received a BREEAM sustainability assessment. During the renovation works, smart building systems were installed on each floor, controlling temperature, lighting, and daylight to meet the sustainability objectives and requirements. The university also seized the opportunity to set up Europe's largest living lab server during the renovation work so that in the future, students and staff could use data for their research.

A living lab is a digital environment in which scientific experiments can be conducted on people in an enclosed environment. The objective for the Atlas living lab is to support over a long period research that focuses on analyzing user behaviour patterns in terms of social interactions and energy use within the building. The building collects and analyses sensor data from the 4th to the 11th floor and does so for the office, study, meeting, and classroom spaces. The Atlas living lab infrastructure consists of; an IP based network infrastructure that is part of the TU/e internal network, dimmable Philips LED lights that are IP addressable, sensors such as the luminaires and motion sensors that capture information anonymously, and control software from Philips Connected office that monitors the luminaires and sensors. These are controlled by a smart energy-saving lighting (SEL) system. These SEL sensors measure, in addition to the lighting and movement also the temperature and CO2 values. The control software can also call the hardware through an API in order to read the sensor values. The living lab also has a server that extracts research data from the systems and runs the applications. This server and all other connected equipment also have a backend database that stores data. This entire infrastructure is enclosed in the Research Environment Atlas Living Lab (REALL) and is accessible to external researchers with an API key or by direct exports given by the TU/e.

The server could not be addressed directly. However, historical sensor data was used in this research to collect data from the SEL to read data on the temperature, motion, air intake & exhaust, and CO2 in ppm. The motion sensors have a short range of 2 to 3 meters around the desks and 3-5 meters for intensive walking movements. In addition, the sensors also provide information on energy consumption, expressed in kWh per m². In this study, all of the above sensor output values recorded per hour, with the exception of the energy output values, were used to develop the "proof of concept". This mainly involved a calculation of the observed value. These were benchmarked against the established baseline values and displayed in the web application.

[28] https://app.diagrams.net/

4 System Design and Component Framework

This chapter explains the use case, system architecture and data collection of the application. It also incorporates the user requirements from the semi-structured interviews with the asset managers. These are the foundation for the use case development. Subsequently, the use case is used to develop the system framework and to determine how data flows should be handled.

4.1 Data collection and User requirements

For the web application, semi-structured interviews were conducted with five asset managers who have experience in managing utility buildings. For reasons of anonymity, the names of the respondents and the university names have been left out. The asset managers manage five university campus areas in the Netherlands from here on out named university one till five. In appendix D, an overview of the interviews can be found. The asset managers have been offered a total of 6 topics with 16 questions. The purpose of the interviews was to gain insights into possible user requirements associated with possible use cases. First, all corresponding answers between the asset managers were collected and then the requirements were identified. The requirements were determined based on what was usable for the application. Thus, on the basis of what values could be used, whether there were KPIs to make certain calculations or certain preferences that would result in an improvement of the asset management activities. As a result of the chosen approach, the user requirements primarily resulted in new target values to measure the building performance indicators. The analysis also provided insight into the use of KPIs and the value of linked data & link building data for asset management. However, the last three points mentioned could not be included in the application as a user requirement. Nevertheless, they are interesting points to consider for a different study. Table 5 below shows the matching statements per question indicated by the index and which user requirements are associated with them.

Table 5 Summary interview correlations and user requirements

Topic	***Q Index***	***Responses that are consistent among asset managers***	***User Requirements***
Asset management	**A1**	Activity patterns, being able to make modifications, sustainability and organisational level, life cycle costing, technical performance, the process for the user, indoor climate, and user satisfaction	None
	A2	linked to business goals, quality of the indoor climate and wellbeing of occupants. Making asset understandable. The functionality of the building	**Focus area IEQ, occupant satisfaction**
	A3	Inefficient data usage, usage patterns of the buildings (occupancy),	None
Current AM IT Tools	**B1**	Yes, it will be the future, but currently not made for the O&M phase. Potentially suitable to link to usage and occupancy patterns. Potentially be used to predict the future value of property	**link to building usage and occupancy**
	B2	No fully integrated systems and trouble keeping data up to date	**Develop an integrated system with up-to-date data**
Sensor Data	**C1**	Occupation, client experience, better climate control, system data for insight in intensity usage, user satisfaction, usage patterns	**Occupation, client experience, better climate control, system data for insight in intensity usage, user satisfaction, usage patterns**

	C2	Ask yourself what you want to achieve with this data, how is the quality of the sensor data, does it interact well with each other, what do you want to analyse. Sensor readings must also reach the end-users.	**Enable end-users to see sensor data, and judge the sensor data quality that is coming in.**
Building Performance	D1	Find a balance in the number of measurement indicators. In addition, this is client specific. Efficient use of the building. Energy measurement but also operational safety and continuity.	None
	D2	500/600 ppm for the CO2 levels and 800 PPM is already too high, nominal air velocity values, the building can't be 100% controlled, between 20-23 degrees the Atlas building can't heat or cool, <20 it's possible to heat and >24 it's possible to cool the temperature.	**500/600 ppm CO2 should be aimed for, and 800 is too high, nominal air velocity values, the building can't be 100% controlled, between 20-23 degrees can't heat the rooms or cool, <20 it's possible to heat and >24 to cool.**
	D3	No KPI indexes were used. Most of the assessments come from experience	**No KPIs**
Linked Data	E1	1. Yes, locating each other more easily provides benefits. 2. Yes, align user satisfaction with assets towards organisational goal. 3. Yes, predictive value when connecting with the schedule of students.	**Aim at occupancy and User Satisfaction**
	E2	1. Yes, by carrying out benchmarking with other assets of other universities. **2.** Yes, for user flow and occupancy that would allow us to increase user cooperation and form clusters,	**Access towards other data sources through linked data would provide added value for asset management.**
	E3	Linking to a digital school timetable would add a lot of value for the asset managers. Consider the end-user and the data quality, and there is a lot of gains to be made from the occupancy	**Link occupancy towards a digital school timetable in order to gain better insight in the usage patterns**
Decision Support	F1	• What do you want to measure? • Why do you want that? • How is the data quality? • What can we do to improve what is occurring at this event?	None
	F2	This would lead to more fact orientated decisions, however, most decisions come from experience. Asset managers make less decisions based on data, due to insufficient data being around or the data quality is still lacking. In addition, they don't trust in technology over their own experience.	None
	F3	No KPIs	**No KPIs for decision making**

A number of salient issues emerge from the analysis, with two main themes highlighted by all respondents. Firstly, all respondents aside from university five have indicated that they do not use KPIs (F3 & D3) for their measurements. Instead, they use a subjective assessment which stems from their work experience (D3 & F2). This corresponds to the method described by de Wilde (2019), in which expert judgement is used to assess the building performance over the use of KPIs. Therefore, KPIs are not taken into consideration for developing the web application because the interviews, in addition to the literature, also resulted in too few insights into measurable KPIs for the building performance indicators. On the other hand, each respondent would like to make more decisions based on facts, preferably from sensor data (F2). However, they indicate that the current systems are not ready yet and that it is also possible that the sensor data does not output the correct values (F2 & C2). This is, for example, the case in university two on the 11th floor on their main building, for the lighting and some motion sensors. The lighting systems on this building are installed behind the workplaces, which means that the workplaces are not sufficiently illuminated, which may lead to additional complaints by the end-users. In addition, some of the movement sensors are placed too close to each other, which results in a double value reading for one individual passing by (see appendix D). These aspects are also recognizable to the other respondents. The sensor data should, therefore, not only provide quantifiable output but this output should also be assessed for quality and accuracy.

It is striking that the implementation of BIM still lags behind compared to the other phases of the AECO sector (B1). The asset managers note that the BIM models are mostly a design and engineering tool and usually do not reach the exploitation phase. Nevertheless, the desire to use BIM for O&M is present for all five respondents. Secondly, all respondents unanimously agree that the most important building performance indicator is the occupancy and usage patterns of the end-users (C1 & B1 & A1 & A2 & E3 & E1 & E2). Once the UI developed in this research provides more insight into the occupancy patterns of the building, this will not only be an added value for the asset manager but also for the end-user (E3 & C2 & C1). After all, the end-user will be able to find his/her workplace more. For the asset manager, more insight is provided into which spaces are used efficiently, thus enabling them to make better use of their space. In addition, all the asset managers consider the building performance indicators of occupancy rate & indoor environmental climate as the most important indicators (A1 & A2 & C1), which also corresponds with the literature findings. However, the respondents do consider it important that these indicators are communicated to the end-user so that they understand the building better and in return improve their user satisfaction. If this is not the case, problems can arise, as was the case with the delivery of the main building on university 2, where an early delivery of the building led to a serious impact on how the end-users understood the building. Certain functionalities of the building were insufficiently communicated to the end-user. The result is that 1% of the end-users complain about various aspects of the building (see appendix D), due to them not understanding how the building works. An important aspect of asset management is linking organizational goals to the property, which can still be challenging at times. The respondents indicated (E1 - E3) that, by means of linked data, it would be possible to link the satisfaction concerning the asset to the organizational goals of the universities.

Moreover, all respondents unanimously agreed that when linked data is used to connect to the digital school timetable, unprecedented added value is created for universities (E3 & E2). After all, they will be better able to meet and anticipate the occupancy requirements. This will also lead to opportunities for future analyses with regard to what the assets might cost next year. There is also added value for the asset manager when he can benchmark his objects against other university buildings or the standard norm using linked data and an integrated online data source on university buildings.

All in all, the requirements are quite clear. The UI must give priority to displaying the occupancy rate of Atlas. For this, all sensor data must first be evaluated. The asset manager must be able to measure end-user satisfaction. For the asset manager, end-user satisfaction is pivotal. This end-user satisfaction stems from how he or she understands and experiences the building, based on its internal climate and its effect on their well-being. For this reason, the following values will be used for the sensor data for IEQ. An interesting field of research would be to investigate how satisfied users are with the building under certain sensor conditions. However, this is currently beyond the scope of this research. For this reason, user satisfaction will not be included in this research.

For the calculation of the IEQ sensor data, 500/600 ppm CO_2 will be used as the target value and 800 as the maximum. In addition, for the temperature, cooling is taken into account when the temperature is above 24 degrees and heating when it is below 20 degrees. Between 20 and 23 degrees, the windows are used for cooling. A pleasant temperature depends on the individual but is usually the average room temperature of 21 degrees. The air velocity values used are the standard nominal values for air velocity. For humidity levels, the aim is to remain between 40-50% to be considered optimal (see section 2.6.2). If the sensor output values remain within the specified bandwidth, the air quality can be assessed as optimal. The value can fluctuate enormously for the occupancy rate, as the number of people expected to come to the building is unpredictable. All in all, the following values can be put in a summary shown in table 6 below.

Table 6 Overview of the final values to be used in the development of the UI

Sensor Category	***Sub Type***	***Literature values as described by (Nimlyat & Kandar, 2015; Li et al. 2018 ; Akanmu et al. 2021 ; Korsavi et al. 2020)***	***Interview values***
Indoor environmental quality	CO2	1180 – 1310 PPM	500/600 (target) and 800 PPM Max
Indoor environmental quality	Temperature	22.8 – 24.2 degrees Celsius	21 degrees Celsius
Indoor environmental quality	Air velocity	0.1 and 0.2 m/s dependent on temperature levels	Same as literature
Indoor environmental quality	Humidity levels	40-50%	Same as literature
Occupancy	Occupancy	Room/Area specific expressed in amount of people/m²	Same as literature

4.2 Use Case

The use case for the web application was developed starting from the main objective of this thesis, in combination with the encountered literature, as well as the available living lab data and the user requirements as described in section 4.1. This use case uses the user requirements and the literature to map the system functional requirements and also identifies the necessary steps an asset manager must take to reach the final goal.

The use case scenario is described using Visual Paradigm's Unified Modeling Language (UML) schema. A UML is often used to make object-oriented analyses and designs of IT systems. The UML includes not only the stakeholders in the use case scenario but also all external entities that interact with the web application. However, the asset manager is still the main actor because he initiates events within the application that cause interactions with external databases and servers. This makes the Atlas living lab server, the LBDServer and the triples stores with the RDF triple secondary actors. Table 7 describes the use case specifications, which can then be used to develop the component architecture.

Table 7 use case

Scenario	*A web interface that navigates through routing on a single page and displays graphs of critical sensor data without continuously re-rendering the page.*
Primary actor	The asset manager.
Secondary actors	1. Atlas Living Lab server. 2. LBD Server. 3. TripleStore Database.
Pre-Conditions	1. The Atlas living lab server is accessible to the application. 2. The data from the living lab server is stored in a MongoDB NoSQL database. 3. A connection is made with the sensor ID in the BIM model and the NoSQL database. 4. The LBDServer components for processing the glTF and TTL formats are reused. 5. The TTL file is converted to RDF and imported to a repository in a triple store. 6. The geometric glTF file is stored in MongoDB. 7. The LBD components for rendering the glTF file are reused in the UI.
User Requirements	1. Display occupancy rate above other indicators. 2. Assess the quality of sensor data output. a. Co2 500/600 PPM b. Temperature between 20-23 degrees and 21 for optimal user experience c. Humidity between 40-50% d. Lighting 500 lux. e. Air velocity nominal values

Description	1. The asset manager accesses the application in the web browser. 2. The application requests components via the LBDServer, which, on the one hand, process the geometric data of the building and make it ready for use in a web browser. On the other hand, it also calls on the components that convert files into data connected to the building's triple store database. 3. The application requests the sensor output data from MongoDB 4. The application renders a tile component with a nested graph component, that contains the data that it receives from the sensor data stored in the triple store. 5. The asset manager is able to make tickets about building components, such as critical energy output found in the Philips lighting system with sensor ID X1.1 6. The Asset manager can interact with the graph and BIM model to get information about building elements and critical sensor data.
Extension	6.1. The asset manager can interact freely with the UI and is able to re-render the graph component, depending on the sensor that he selects in either the BIM model or button component (e.g. graph displays motion sensor data output instead of IEQ). 6.2. The asset manager is able to return to the initial state of the component by switching between IEQ, Occupancy, energy use overviews. 6.3. The overviews are linked to a set of three buttons, and each button is for a single KPI. 6.4. When a sensor is selected, the application makes a request with the NoSQL database, which on its own accesses the Atlas living lab server.
Exceptions	• If possible, no SPARQL Queries are conducted on the front-end UI • No RDF Graphs are displayed on the front-end UI
Results	The asset manager is able to display the triplestore visualization of the BIM model in a web UI and is able to access data from the Atlas living lab within a graph.

The use case initially creates the foundation for the development of the component architecture. Figure 13 shows a representation of the use case in a UML diagram. The UML use case diagram showcases how the primary and secondary actors interact with each other. Given the available data from the living lab server, the expected sensor output data will contain data on temperature, CO2 levels, and occupancy. Some of the information that the BIM model can retrieve is geometric data such as floor area, height, length, and width and building element structures. However, it should also be possible to access the virtual sensors through references, sensor ID, and location.

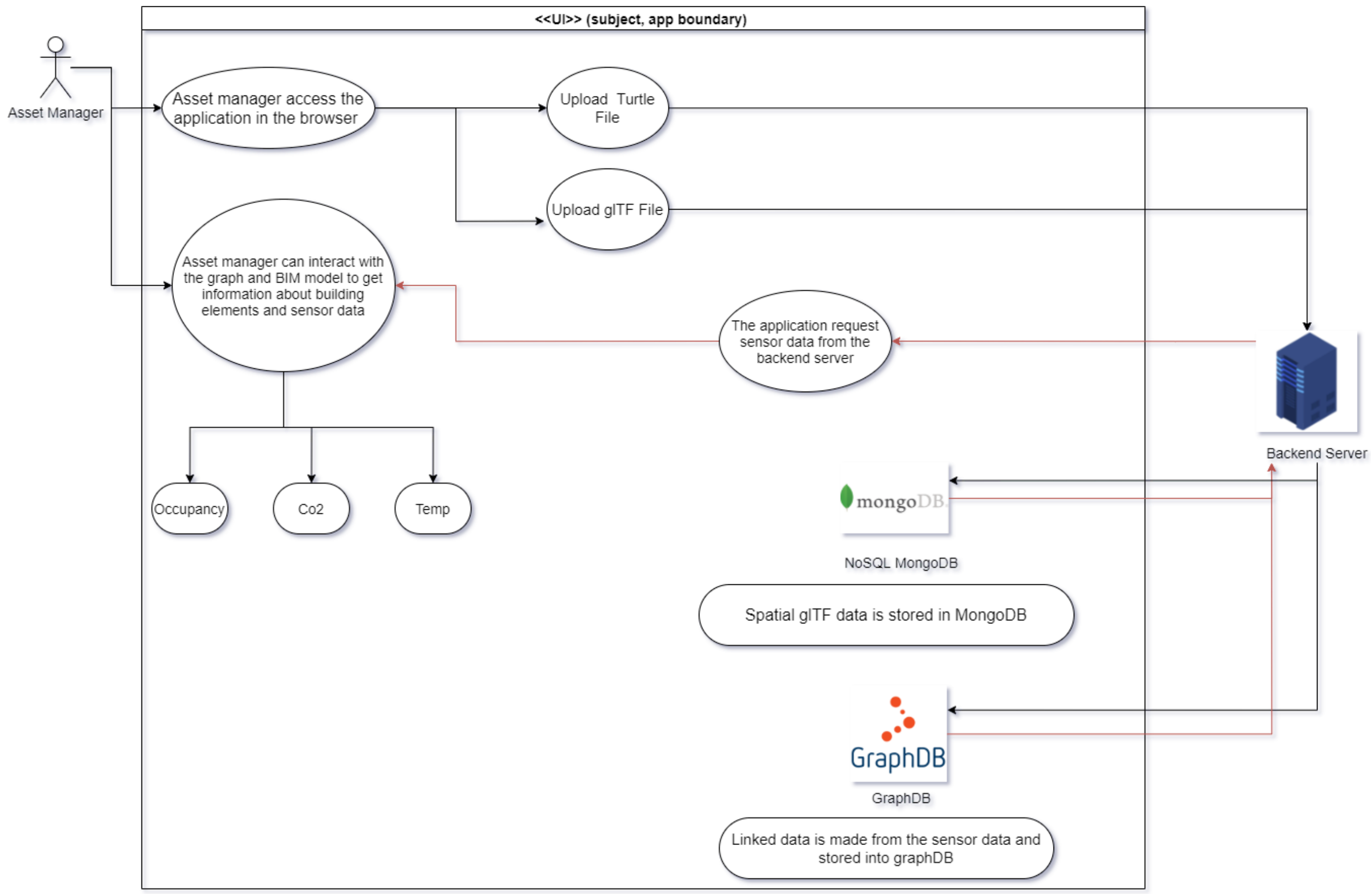


Figure 13 UML Use Case

4.3 Component Architecture

The component architecture is the conceptual model of the prototype application. In this framework, the use case requirements from the UML are translated into individual component placed in the framework. The framework shown in Figure 14 is composed of a four-layer framework consisting of a front-end UI that is connected to the underlying back-end controller- and service layer by the model layer. Each layer will be explained in detail in section 4.4. The purpose of this framework of components is to display the asset manager's user interactions and illustrate how the underlying operating mechanisms handle this. The model, therefore, consists of data flowing down from the user. These data flows are often triggered by changing events and state handlers in the dashboard, such as when a user uploads a file or initiates a search on the sensor data. According to W3schools[29] and the developer docs from React.Js[30], both "state" and "props" are JavaScript objects which contain some information that can be used to influence how the output of a component should be rendered. However, the distinct difference is that props are passed to a child component in the same way that parameters are passed to functions. In contrast, 'state' is only managed within the component in the same way that variables are declared within a function. In addition, W3schools adds that when the state object changes, the component then re-renders and is displayed again. The model also consists of data flowing upwards and being presented in the display rendering component. This is data coming from the GraphDB database, the sensor data from Atlas, and the geometric data that comes from MongoDB a is presented in the BIM model viewer component.

[29] React State (w3schools.com)
[30] Component State – React (reactjs.org)

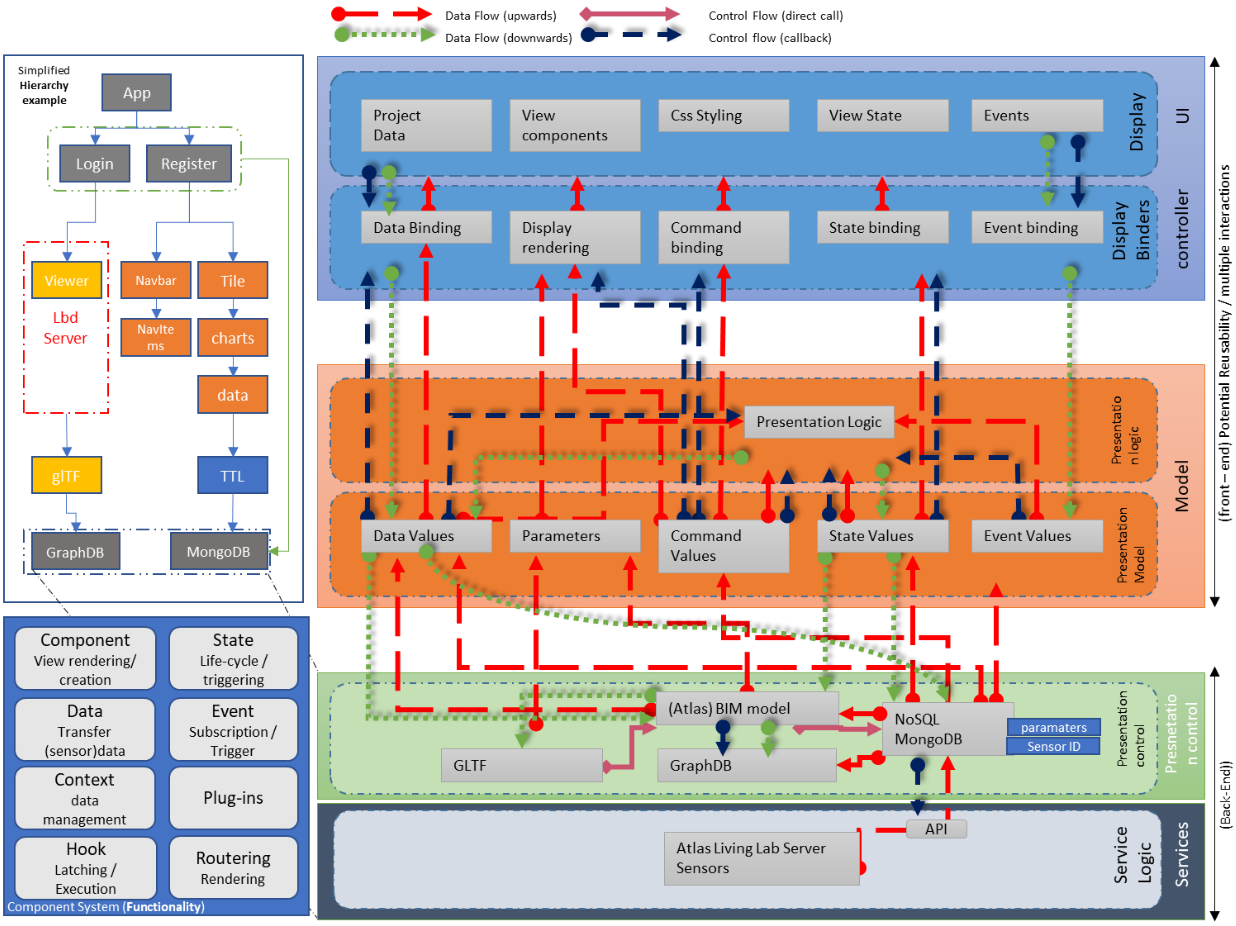


Figure 14 Component Architecture based upon ComponentJS[31] framework

4.4 UI

4.4.1 Display

The UI component handles data flowing from and towards the underlying model component by observing the model components parameters, state values, data values and given command values. In addition, the UI component translates the user interaction events into the model component by changing the setState/useState[32] and event values. The UI component consists of two layers: the display layer mainly meant for the end-user and the display binding layer that is meant to capture the user interactions and bridge them to the model component. In React, this layer mainly consists of the App component, which houses underlying components in a tree structure, see appendix E.

4.4.2 Display binders

The display binding layer houses various components which handle an event, state, and the initial display rendering. The initial display rendering component renders the default app component based upon the presentation models parameters value, state values and data values. This component is rendered continuously along with the changing state, data, and event values. The state binding component observes the initial state values in the presentation model and updates the state values in the UI-display layer. The data binding component is vital to this research. Its main

[31] https://componentjs.com/architecture/ui-component-architecture.pdf
[32] Hooks API Reference – React (reactjs.org)

purpose is to observe the initial (sensor) data values in the presentation model layer. Each time the data is updated, it triggers the components in the UI display layer to re-render and adjust accordingly. The command binding component holds a similar structure as the state binding component.

4.5 Model

The model component consists of two sub-layers, namely the presentation model and presentation logic layer, which house several components related to the default state, data handling, and how to handle functional logic within the web application. The model component holds no dependencies and is essentially a derivative of the UI component. It holds the parameters, data, state, and event values that both the display and controller components observe and interact with. Its primary function is, therefore, to construct a connection from the UI with the controller layer. Previously, it was assumed that within React, data is passed from top to bottom (from the parent component to the child component) through the use of props. However, since the model layer has to communicate in both directions, this can lead to problems.

Therefore, within React, objects such as " Context[33]" and hooks such as " useContext " are used to pass props without navigating through each layer of the app tree. Hooks are functions within React similar to React's own lifecycle methods. They are meant only for functional components where React state can be updated. Thus, context allows an application to set and share values between different components without using props through each level of the component hierarchy. Context thus enables an application to establish and share values between different components without having to use props through each level of the tree. The useContext hook is a default React hook that accepts the context object and returns the current context value. A component that calls useContext will always re-render when its context value changes. This is especially important when the asset manager uploads files and requests sensor data from the back-end.

4.5.1 Presentation logic

The Presentation Logic Layer includes all presentation logic elements. It is responsible for all decisions that have to be made based on the presentation model layer. It is within this layer that the “useContext” and “useState” hooks mainly use the initial context and initial state values stored in the presentation model layer to update components.

4.5.2 Presentation model

As previously shown, the presentation model layer includes all default data, parameters, state and event values aspects, all of which are stored in the initial context. The primary purpose of this layer is to record all default information from the UI layer. A base value is needed that the user can always fall back on or that he can adjust.

In this layer, the parameter values are all values that configure the display rendering aspect. It also contains the components originating from the LBDServer, which render the geometric BIM data. Likewise, the styling of the web application for the sensor data in the graph components and tile components are processed within this parameter component. The command values in this layer are all values that model external actions for the UI layer, like content reset or deleting a ticket. State values in this layer are all the values that handle for the UI component, errors, displaying of objects, enabling objects, and other similar state objects. The data values components that are processed in this layer receive all (sensor) data and geometric data from the controller layer. These are then displayed in the UI. These data value components are several different components that each have a specific functionality with regard to the application. For example, the viewer component has the purpose of displaying the BIM model after the user has uploaded a glTF file. In addition, the sensor values will be displayed in graphs. Therefore, the sensors will

[33] Context – React (reactjs.org)

first have to be captured in a data component and then used in a graph component, so that they can be displayed in the UI.

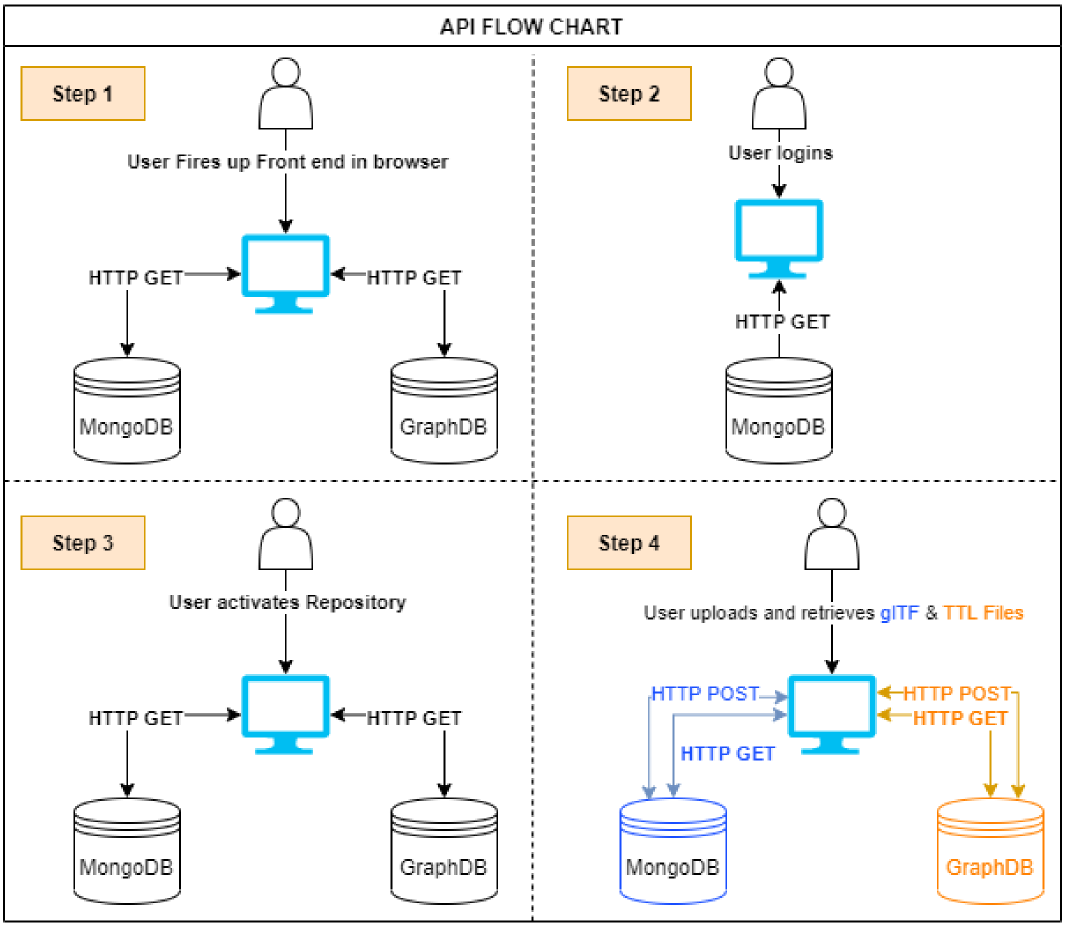
Finally, the default event values are modelled in the UI component with no directly related data. This happens on behalf of the event component. The components in the controller layer observe these events. Then, when a button is pressed or when the viewer is clicked, these components will initiate a call to the service layer to request data about the sensors or the spatial representation of Atlas. This completes the data flow of the web application. For example, when React renders components, they are rendered using lifecycle methods, which tell the application that when certain events happen, what actions the components should perform. Going back to the previous example, when the glTF files are uploaded by the user, the viewer component is refreshed to display a model instead of a blank page.

4.5.3 Data Handling and API Post/Get Requests

Based on this component framework, the web application lends itself well to manage user interactions and store and retrieve data. Initially, the user is asked to launch the application in the browser (e.g., Google Chrome, Edge, Firefox) during start-up. This triggers the back end (MongoDB & GraphDB) through an HTTP request to connect to the front-end. Subsequently, the user is then able to register. This launches an HTTP "POST" request, which stores the user's data in MongoDB. This allows future projects to be linked to the user's profile by saving them in a user-owned repository. On a second visit, the user can log in, which triggers an HTTP "GET" request that retrieves the data stored within MongoDB, including the user repository, which contains his/her previous work.

For the project, there are also several HTTP "POST" and "GET" requests. First of all, the user must create a project "repository". This automatically initiates an HTTP POST request in the background. After a successful HTTP request, the user must upload the geometrical data stored in the glTF and TTL format to the back end. In the interface, the user is given an option to upload both the glTF file and the TTL file by means of an upload button. Likewise, in the background, a POST request is processed by Axios, which saves the glTF file in MongoDB. Immediately after the upload, a GET request is triggered, retrieving the geometrical data stored in the glTF file from MongoDB, parsing it, and displaying it in the Viewer component. In addition, the user has to upload the TTL file of the building, which contains the semantic data, in which the sensor data is stored with all the "sensor IDs". The web application then posts an HTTP "POST" request to the GraphDB backend, which also immediately triggers a "GET" request, thus forming a direct connection between the geometric object displayed in the viewer component and the data contained in the semantically rich TTL file. This connection is formed solely on the guids. Each building element should have a unique guid that can be found in both the TTL file as well as in the glTF file. In Figure 15 below, the above steps taken by the APIs are represented in a flow diagram.

Figure 15 API flow diagram



4.6 Conclusion

This chapter attempted to answer the question, "What system architecture is needed to build a decision support system from BIM data and building performance data?" As a result, the focus of this chapter was primarily on developing the component framework for the linked building data asset management dashboard. This is divided into three parts, the user requirements & use case, the application structure, and finally, the data flow handling.

A total of five interviews were conducted with five experienced asset managers. These interviews revealed that no KPIs were used for assessing the building performance. Therefore, they were not included in the development of the dashboard. In addition, all respondents indicated that the sensor data may contain error margins and should be assessed for quality. The most important sensor data for the asset manager are the building's occupancy rate and the quality of the indoor environment for the end-user. These requirements together form the user requirements, which were included in the use case. In addition to these requirements, the web application had to meet seven pre-conditions, including the reuse of LBDserver-derived components for processing and displaying the glTF and TTL files, connecting the sensor ID to the TTL file and connecting the sensor data to the dashboard UI.

Starting from these requirements and use case, it was decided to achieve the following as a result: The asset manager is able to display the triple store visualization of the BIM model in a web UI and is able to access data from the Atlas living lab within a graph. Hereby the asset manager must be able to upload the BIM model and connect to the sensor data without being bothered with the whole process of converting individual files. The proposed framework and use case can be assessed for strengths and weaknesses.

The chosen approach for the framework has its strengths, particularly in reusability and the way data flow is handled. By dividing different parts of the dashboard into components, each with its own set of tasks, the usability is increased because the number of dependencies is reduced. For example, the data values components in the presentation model layer handle all data flow coming from the presentation control layer, such as the sensor data, BIM viewer and the information stored in the TTL file. These are separate components, so the sensor data is stored in individual components that first capture the sensor data and then display it in graphs in another component.

In addition, React requires a fairly strict way of handling data, with properties needing a top-to-bottom approach. By using React's own functions, methods and Hooks such as “Context”, “useContext”, and “useState”, it is possible to apply a proper form of state management and data flow handling. The disadvantage is that for future new developers, these are rather difficult concepts to learn. A new developer needs knowledge of state management, React Hooks and context management. An alternative is to use the React-Redux library, which is covered in detail in section 5.1.6. React Redux handles large and partly all state management and context management data flow. However, React-Redux is also difficult to master at first. In the long run, however, it is more useful than React context.

5 Proof of concept

This chapter describes in detail the steps taken in the development of the prototype linked building data asset management dashboard, from here on out 'the dashboard'. These steps include the application layout, the internal mechanism for retrieving and processing data and how this data is exchanged internally within the components. The complete application code can be found on GitLab[34]. For a complete overview of the entire framework of the front- and backend components, it is highly recommended to consult appendix E & F.

5.1 Prototype development

The prototype development implements the component framework as described in section 4.3. This section describes the dashboard's user interface and how it communicates with the API endpoints of GraphDB & MongoDB through the model, controller, and service layers.

5.1.1 Application Layout Front – End

The first step in implementing the component framework is determining the user interface layout while considering the requirements as described in section 4.2. This requires the asset manager to be able to view the occupancy and indoor climate of the Atlas building in particular. In addition, the asset manager should be able to see a 3D viewer in the UI after uploading the glTF file. Finally, the asset manager must be able to upload turtle files containing the linked building data to the backend. The front and back end should perform the storing and parsing of these files in the background, invisible to the user. From this point, the dashboard must store, process, and retrieve the files through API calls to MongoDB and GraphDB.

Moreover, React works with a component hierarchy "Tree" structure to render a virtual DOM (see appendix E). A virtual DOM is, as the name suggests, a virtual representation of the real DOM. React constructs a virtual representation of the application and compares the changes with the actual state of the application (the real DOM). When React detects a change, it only updates the components where the changes are detected. This increases the performance of React considerably. The primary purpose of the component hierarchy is to deliver a set of responsibility for individual components and the relationships between them. This provides for a healthy balance in structure and flexibility. However, it creates a dependency on the data flow, which means that it is only possible for data (properties) to flow downwards from the parent component towards the child. This is often referred to as unidirectional data flow[35]. A child component should never update the parent data but only use it as a read-only. So, a solution will have to be devised on how to handle the different data streams.

Thus, considering all the above functional requirements, the decision was made to divide the layout of the dashboard into the following parts. Each component renders an individual piece of the dashboard. React has two component structures, namely, class-based components and functional components. The most significant difference between the two components is the syntax and state management. Functional components are JavaScript functions that return a syntax similar to HTLM called JSX. Class-based components is a JavaScript class that extends "React.Component[36]" and contains a render method inside.

[34] ISBE / AtlasDigitalTwin · GitLab (tue.nl)
[35] Unidirectional Data flow in React – TA Digital Labs
[36] React.Component – React (reactjs.org)

For the structure of the dashboard, reusable tile components will be generated that can hold various other components. For example, the sensor data will be rendered in graph components, and these will then be rendered in the tile components. Also, the user must be able to log in and link projects to his/her profile. By using Figma, it is possible to create an initial UI in which the various components can be visualised. A simple representation of the hierarchical structure of the component framework and the initial user interface designed with Figma can be seen in Figure 16. Figure 27 before section 5.1.6 shows a view of the complete front-end framework.

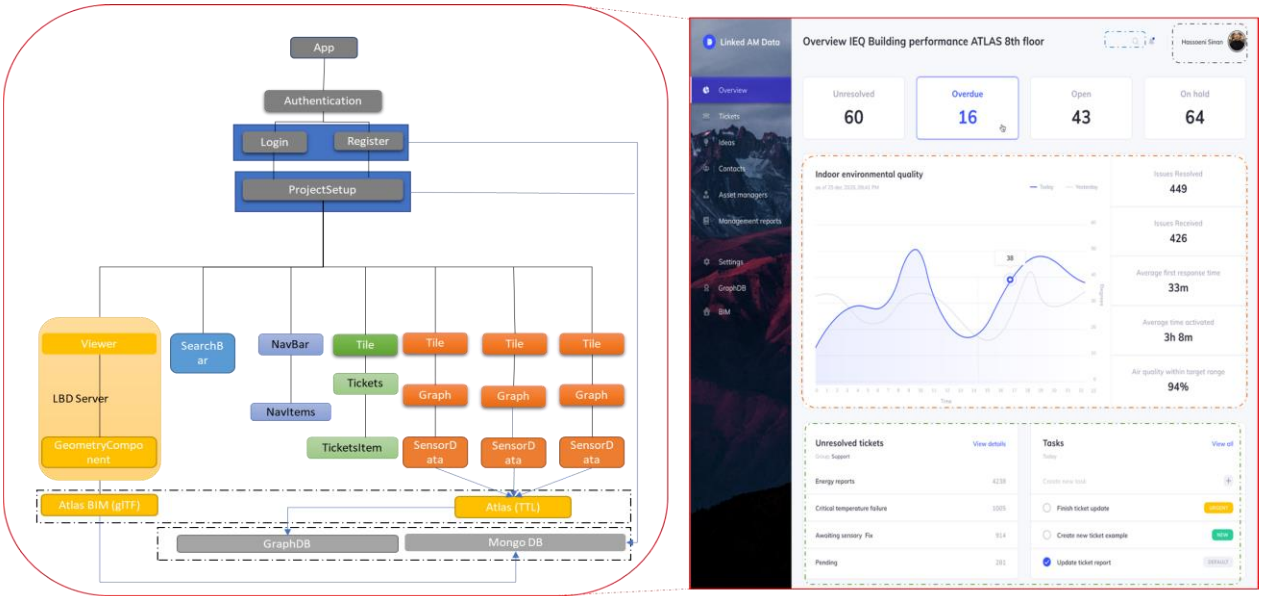
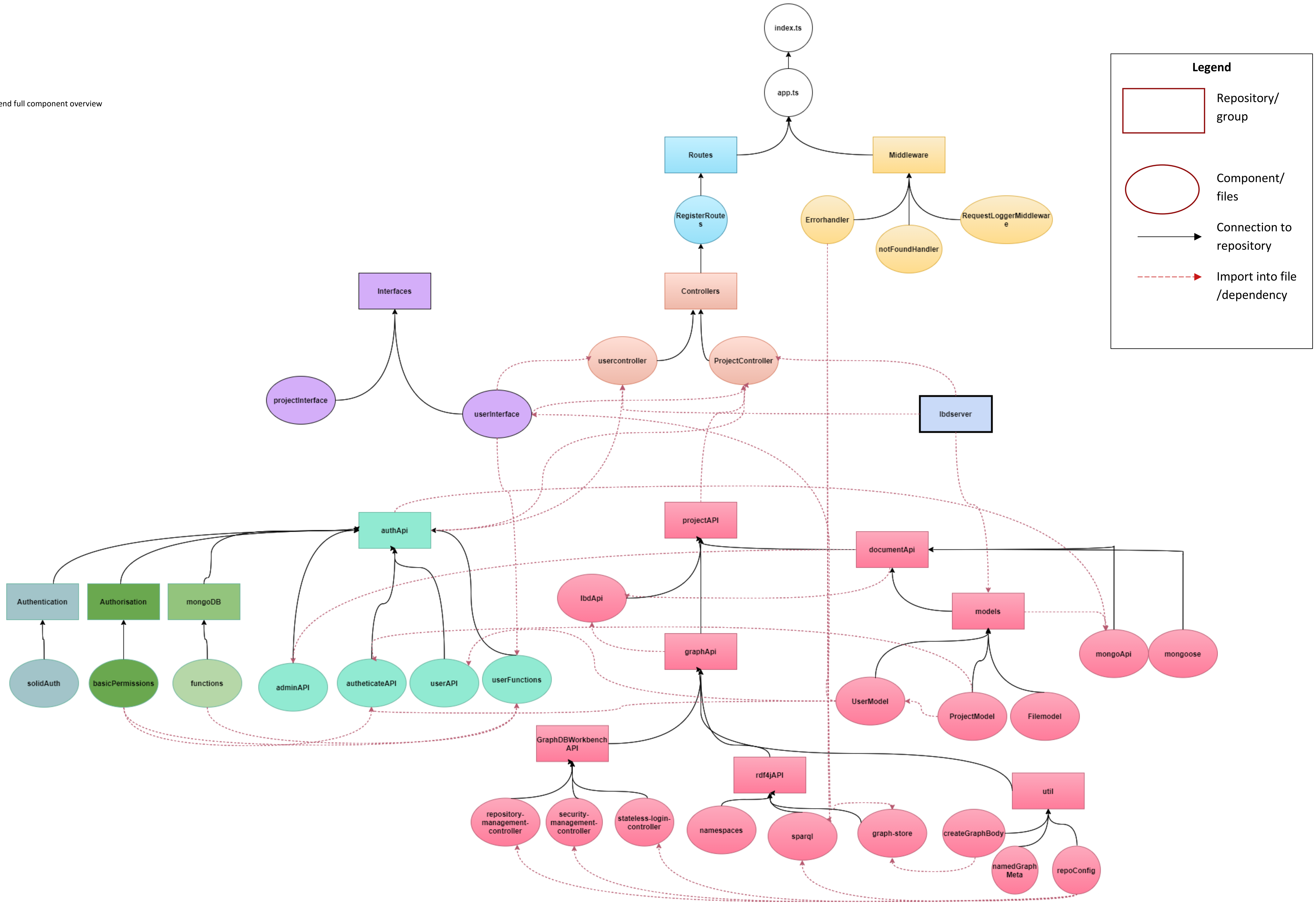


Figure 16 initial UI and simplified structure Linked building data asset management dashboard

5.1.2 Application Layout Back – End

Figure 17 shows a full view of the back-end. A legend is attached in both this representation of the back-end and the front-end. In contrast to the front-end, the back-end does not consist of a strict hierarchical structure. Different components can therefore be imported without having a parent-child relationship. All imports are marked with a red dotted line, or direct imports are marker by black arrows in the back-end. All square blocks in both the front-end and back-end mark self-named groups or repository directories. Each colour in the back-end and front-end indicates a group of repositories and the underlying files or components that are part of them. The ellipses represent the individual components under these repositories. In the Front-end framework in Figure 27, the parent-child relationships are represented by the arrows. Unlike the back-end, the front end has a context component. All components into which the context file is imported have blue dotted relationship lines.

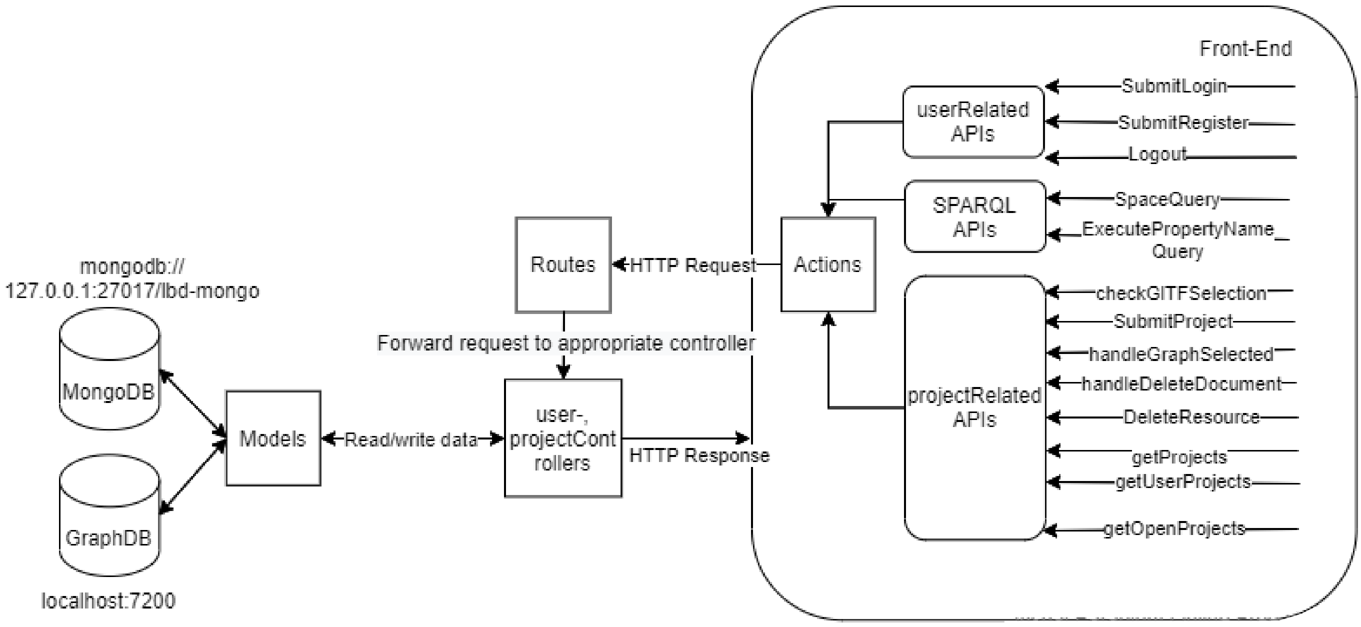
Figure 17 back -end full component overview



5.1.3 Front to back-end API handling

The dashboard must process data coming from the users in multiple ways. In React.js, there are several possibilities to process an API request. This can be done by programming "fetch[37]" requests or by using external libraries which handle most of the requests. Using one of these external libraries called "Axios[38]", asynchronous API HTTP protocol POST and GET requests are processed more efficiently. The dashboard uses Axios to communicate with the end-points of MongoDB and GraphDB. When a user creates a profile, Axios communicates with MongoDB in the back-end and stores this data. The same principle applies when the user creates a project and uploads the glTF files. In both cases, Axios communicates with the back-end MongoDB through an API endpoint. The same principle also applies when a user uploads a TTL file only in this case the user communicates with the GraphDB back-end. The key factor here is the way in which the back-end is connected to the front-end. All the routing is handled in the “routes.ts” file found in the back-end. The controllers handle each request to the backend. The backend knows two types of controllers; a userController which handles all the user related requests, and a projectController which handles all project, user and document related requests. The controllers derive their requests from the models that are created in the projectAPI and authAPI directories, see Figure 17. The routes help with forwarding all HTTP API protocols for creating, retrieving, deleting, updating the user/project data, and help with processing the files uploaded by the end-user in the UI to the appropriate controllers (see Figure 18).

Figure 18 (right to left) how the frontend gets connected to the backend through routing, controllers and models.



The back-end uses components derived from the LBDServer to handle all communication protocols. Each of these builds up the server for the back-end in the “index.ts” file. This server is responsible for storing and organising all incoming data and ensures that everything on the client-side works as it should. For this, the server uses various software tools such as Express.js, Cors and Dotenv. Express.js is a routing and middleware Node.js web framework that has few functionalities. In essence, Express[39] is a series of middleware function calls. Middlewares are functions that access, request (req) and response (res) objects. It is partly because of middlewares that responses coming from users, such as uploading a form, can be handled. Routing refers to how an application endpoint (URIs) communicates

[37] Using Fetch - Web APIs | MDN (mozilla.org)
[38] axios/axios: Promise based HTTP client for the browser and node.js (github.com)
[39] Installing Express (expressjs.com)

with the client request. This is done by using Express methods such as app.get to perform an HTTP "GET" request and is essential for the API handling. Dotenv[40] is a module that stores environmental variables in a .env file, which helps with more efficient application functionality. First, a port variable is defined to connect the server to the browser. For this purpose, Port 5000 is included in the .env file as a variable. It is from this port connection that the front and back end are connected with each other.

In the app.ts file, the routes are also required for navigating the API handling requests to the correct endpoint. For this purpose, the app.ts file uses the RegisterRoutes function from the routes.ts file. The primary API endpoints are defined as “/” for the authRoutes and “/lbd” for the projectRoutes. A web service endpoint is a URL where the service can be accessed by the client application. In this case, the front-end is the client application, and the back-end is the service application. The endpoints in question are marked with "/", “/lbd”, “/register” or "/login", and these consist of small interfaces between the API and the client application, see Figure 19.

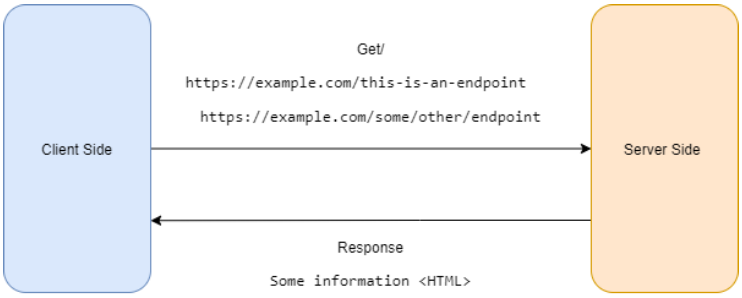


Figure 19 Client / API end point handling

5.1.4 back-end: Project Spatial data (glTF) API handling

In order to upload the files to MongoDB, parameters must first be defined to which the object must adhere. For this purpose, a mongoose projectSchema (found at ProjectModel.ts) function was first defined, which specifies that each project folder should be given an id, and URL string parameters, see Figure 20. In addition, the files themselves must receive a main buffer[41], a URL string[42] and a project string variable. The fileSchema can be found in the FileModel.ts component. MongoDB schemas[43] are JSON objects that allow the back-end to specify the content of documents and what form that content should take.

[40] dotenv - npm (npmjs.com)
[41] An Overview of Buffers in Node.js | www.thecodebarbarian.com
[42] Stata Guide: String Variables (mwn.de)
[43] Document Schemas — MongoDB Realm

Figure 20 File and ProjectSchemas

```
import * as mongoose from 'mongoose'

export default
const fileSchema: mongoose.Schema = new mongoose.Schema({
    main: {
        type: Buffer,
        required: true
    },
    url: {
        type: String,
        required: true
    },
    project:
        {
            type: String,
            ref: "Project",
            required: true
        }
}, {
    timestamps: true
})
const File = mongoose.model<mongoose.Document>('File', fil
eSchema)
```

```
import * as mongoose from "mongoose"

export default
const projectSchema: mongoose.Schema = new mongoose.Schema
({
    _id: {
        type: String,
        trim: true
    },
    url: {
        type: String,
        required: true
    }
}, {
    _id: false
})

const Project = mongoose.model<mongoose.Document>('Project
', projectSchema)
```

With the above parameters defined, the following group of functions is set up in the mongoApi component; uploadDocument, deleteDocument, getDocument, deleteProjectDoc, createProjectDoc, pushProjectToCreator, deleteProjectFromUser, and findAllProjectDocuments. Figure 20 shows the upload and getdocument functions.

```
async function uploadDocument(id, data) {
 try {
  const bucket = createBucket();
  const _id = v4();
  filename = `${process.env.DOMAIN_URL}/lbd/${id}/files/${_id}`
;
  const readable = new Readable();
  readable._read = () => {};
  readable.push(data);
  readable.push(null);
  const stream = bucket.writeFile({ filename}, readable);
  console.log('stream', stream)
  console.log('_id', _id)
  return (filename);
 } catch (error) {
  error.message = `Unable to upload document; ${error.message}`
;
  throw error;
 }}
```

```
function getDocument(id, fileId) {
 return new Promise((resolve, reject) => {
  try {
   let file = ''
   const filename = `${process.env.DOMAIN_UR
L}/lbd/${id}/files/${fileId}`;
   console.log('filename', filename)
   const bucket = createBucket()
   const readStream = bucket.createReadStream
({filename})
   readStream.on('data', (chunk) => {
    file += chunk.toString("base64");
   })
   readStream.on('end', () => {
    resolve(file)
   })} catch (error) {
   error.message =
`Unable to get document with id ${id}; ${error.m
essage}`reject(error)}})}
```

Figure 21 Upload and getDocument functions

The function uploadDocument is an asynchronous API function that uploads documents if the values from the function meet the set parameters defined in the various MongoDB schemas found in the "Models" components. Once this is the case, the file is uploaded to the following domain: "https://lbdserver.org/".

The getDocument, on the other hand, is an API function that returns a promise[44] (see Figure 22), which is called to check if the function meets the set parameters when retrieving the documents. If the conditions in the getDocument function are met, the opposite of the uploadDocument function is executed. The file is returned from the MongoDB database.

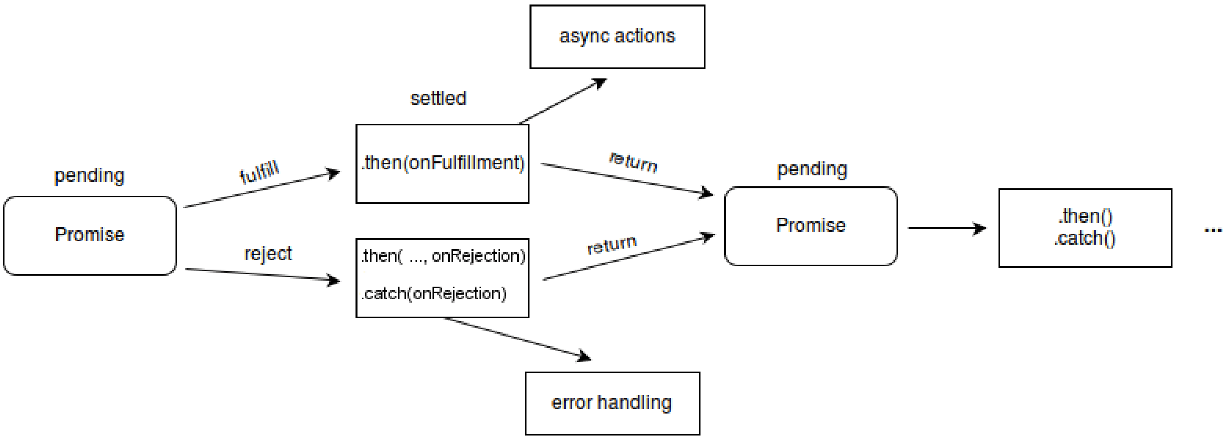


Figure 22 Promise API source: Mozilla (2021)

A Promise is a replacement for a value that is not necessarily known when the promise is made. It allows the application to assign handlers to the final success value or failure reason of an asynchronous API operation. This enables asynchronous methods to return results in the same way that synchronous methods do: so rather than delivering the final value immediately, the asynchronous method provides a promise to deliver the value at a later time. (Mozilla 2021). Figure 23 below shows a flow diagram of the API calls that come from the front-end and are sent to the back-end. It also shows which primary files these API calls go through, and which functions are called.

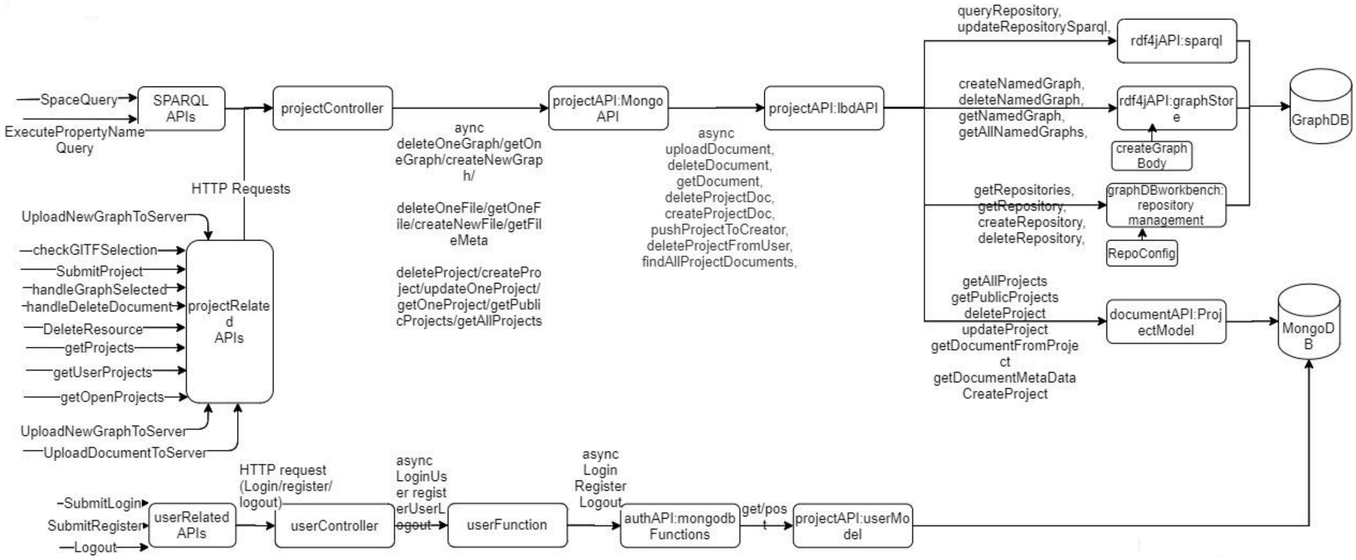


Figure 23 full API overview and the components they pass through

[44] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

5.1.5 back-end GraphDB Connection

In the back-end, there is a folder structure called "graphAPI" that handles all API operations for creating, updating, and retrieving the RDF files, see Figure 17 or appendix F. It follows the developer REST API documentation structure as described in the GraphDB documentation[45]. This developer documentation is divided into two parts: the GraphDB Workbench API and the RDF4J API. The GraphDB Workbench API manages all security and repository management controllers, while the RDF4J API manages all RDF graph stores, namespaces, contexts, SPARQL queries and general repository management. All documents are uploaded to the GraphDB URL localhost:7200/ as defined in the .env file, see Figure 24. Central to the connection with GraphDB is the creation of a repository using the GraphDB Workbench API and uploading the graph data file to this repository. Several functions have been written for this purpose. For creating a repository, the repoConfig function found in the repoConfig.js file and the createRepository function found at the repository-management-controller.js file are of importance. For uploading the graph data file, the createNamedGraph (graphStore.js) and defaultBody (createGraphBody.js) functions are pivotal. Both the defaultBody and the repoConfig functions specify the parameters that must be met before a repository can be created and what conditions the RDF file must meet before it can be uploaded.

```
PORT=5000
TOKEN_SECRET=replace-this-secret-with-something-else
MONGODB_URL=mongodb://127.0.0.1:27017/lbd-mongo
GRAPHDB_URL=http://localhost:7200
DOMAIN_URL=https://lbdserver.org
GDB_ADMIN=admin
GDB_ADMIN_PW=admin
FRONT_END_URL=http://localhost:3001
```

Figure 24 .env file

The createRepository function receives the properties title & id from the repoConfig function. Figure 25 shows a scaled-down view of the repoConfig function. Next, using Axios, a POST request is performed on the URL localhost:7200/rest/repositories, creating a repository with security authentication.

```
async function createRepository(title, id) {
 try {
   let repoconfig = repoConfig(title, id);
   const formData = new FormData();
   formData.append("config", repoconfig, "config");
   const url = `${process.env.GRAPHDB_URL}/rest/repositories`;
   const headers = {"Content-Type": `multipart/form-
data; boundary=${formData._boundary}` Authorization: `Basic ${btoa(
     process.env.GDB_ADMIN + ":" + process.env.GDB_ADMIN_PW)}`,};
   const response = await axios.post(url, formData, { headers });
   console.info(`Created repository with id ${id}`)
   return(response.data);
 } catch (error) {
error.message = (`Failed creating repository; ${error.message}`)
   throw error}}
```

```
exports.repoConfig = function(title, id) {
  return `#
  # RDF4J configuration template for a
GraphDB Free repository
   @prefix rdfs: <http://www.w3.org/20
00/01/rdf-schema#>.
   @prefix rep: <http://www.openrdf.or
g/config/repository#>.
   @prefix sr: <http://www.openrdf.org
/config/repository/sail#>.
   @prefix sail: <http://www.openrdf.or
g/config/sail#>.
   @prefix owlim: <http://www.ontotex
t.com/trree/owlim#>.
   @prefix shacl: <http://rdf4j.org/confi
g/sail/shacl#>.
   [] a rep:Repository ;
     rep:repositoryID "${id}" ;
     rdfs:label "${title}" ;
```

Figure 25 creating a repository

[45] Note: for in more in depth documentation an installation of the GraphDB free version is needed, this can be found in: https://graphdb.ontotext.com/documentation/free/devhub/workbench-rest-api/location-and-repository-tutorial.html

It is a given that every repository must also receive semantically rich RDF files. The defaultBody acts as a container for the parameters and properties that are passed to the createNamedGraph function (see Figure 26). The context, baseURI and data are of importance in this connection. When the user uploads a Turtle/ttl file, it is put into an empty JSON object array, parsed and with a POST request posted to the URL http://localhost:7200/rest/data/import/upload/the repository id/text.

Graph-store

```
async function createNamedGraph(repositoryId,{ context, base
URI, data },token) {
 try {
   const body = JSON.stringify(defaultBody(context, baseURI, da
ta));
   const url = `${process.env.GRAPHDB_URL}/rest/data/import
/upload/${repositoryId}/text`
   console.log('url', url)
   const options = {
    method: "post",
    url,
    maxContentLength: Infinity,
    maxBodyLength: Infinity,
    headers: {
     "Content-Type": "application/json",
     Authorization: `Basic ${btoa(
      process.env.GDB_ADMIN + ":" + process.env.GDB_ADMIN_
PW
     )}`,
    }, data: body,
   } const response = await axios(options);
   console.info(`Created named graph ${context}`)
   return parse(data);
 } catch (error) {error.message = (
    `Failed to create named graph ${context}; ${error.message}`
   ) throw error}}
```

createGraphBody

```
defaultBody = (context, baseURI, data) => {
  return {
    name: context,
    status: "NONE",
    message: "",
    context,
    replaceGraphs: [],
    baseURI,
    forceSerial: false,
    type: "text",
    format: "text/turtle",
    data: data,
    timestamp: Date.now(),
    parserSettings: {
      "preserveBNodeIds": false,
      "failOnUnknownDataTypes": false,
      "verifyDataTypeValues": false,
      "normalizeDataTypeValues": false,
      "failOnUnknownLanguageTags": false,
      "verifyLanguageTags": true,
      "normalizeLanguageTags": false,
      "stopOnError": true
    },
    xRequestIdHeaders: null }}
module.exports = defaultBody
```

Figure 26 Graph-store and createGraphBody

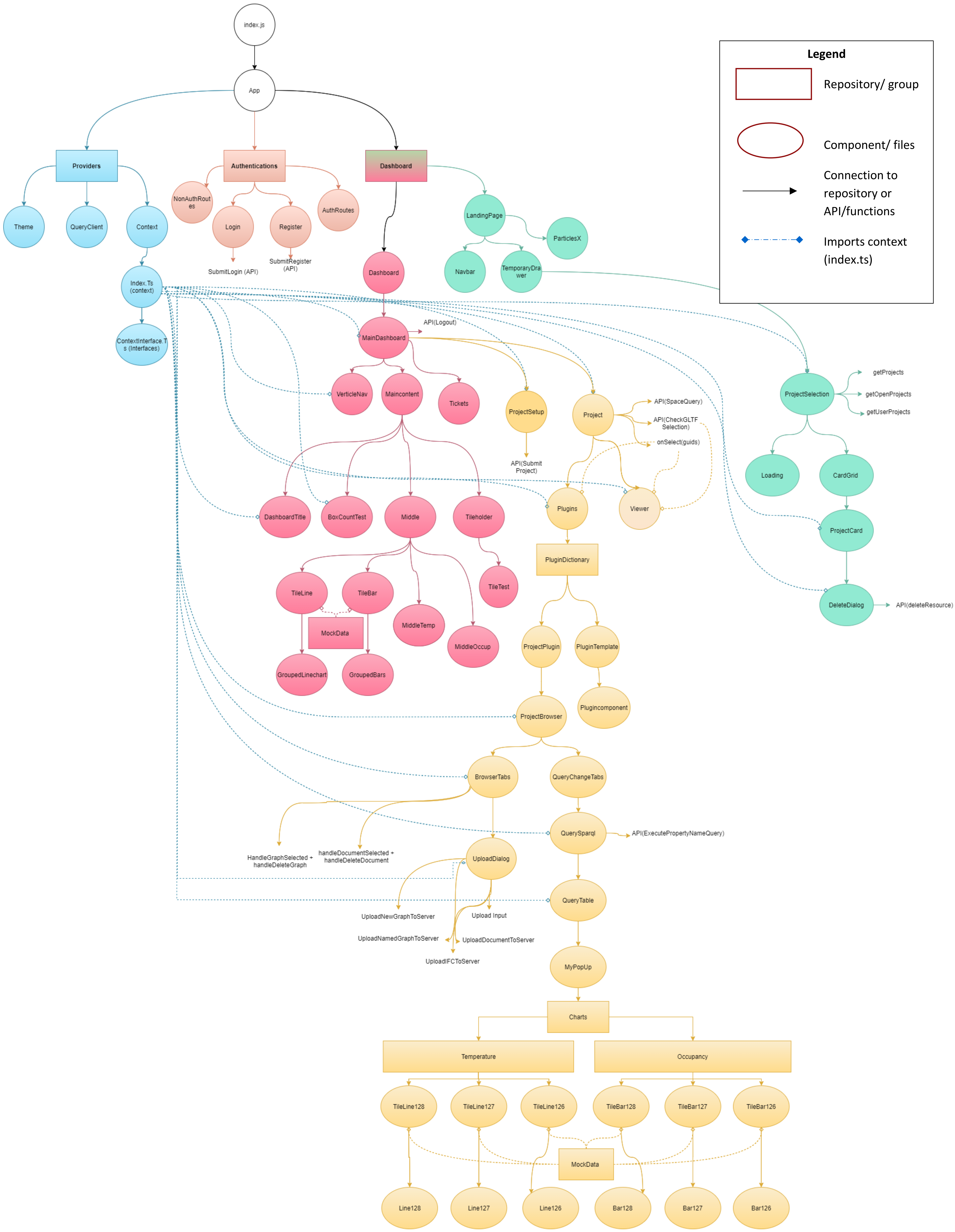


Figure 27 Front-end hierarchy structure

5.1.6 Front End: file uploading handling spatial data (glTF) & TTL File

For the front-end, four components play a significant role in uploading and processing the glTF & TTL files and in creating a GraphDB repository. These are, BrowserTabs.js, UploadDialog.js, ProjectSetup.js, ContextInterface.ts and the context index.ts file. In the context file, the initial state instances are defined. This can be an empty array, an object, or a string of instances. This file and its mechanism follow a similar principle as the React-Redux principle. React components can be seen as functions. Although it is possible to write an extended function, it is better to split the code into smaller functions designed to perform a specific set of tasks. Similar to what happens in the back-end with the defaultBody function that passes properties to the createNamedGraph function. It is within this context file that the interfaces from the ContextInterface.ts are called. Interfaces are part of one of the core principles of Typescript and are specifically intended to verify which shape values hold. Similarly to the back-end, it is common to divide react components as "container" components that are responsible for collecting and managing data. One can think of display components that display data in the UI based on the properties they receive from other components. The React-Redux library[46] helps to reduce the code size by providing various functions such as Redux "store" and Redux "connect". The context index.js file fulfils the "store" function, which stores the current store state that can be extracted and updated with data received by the projectSetup and BrowserTabs components. Figure 28 shows the "store" of the application used as the initial state. Figure 29 shows the redux principle and how this context fulfils this principle and thus influences the application.

Figure 28 Initial "Store" State and Interfaces

```
import {createContext} from 'react'
import {IContext} from "./interfaces/contextInterface"

export const initialState: IContext = {
 user: null,
 currentProject: null,
 states: [{"project": {}}],
 plugin: "project",
 selection: [],
 subject: [],
 predicat: []
}
const initialContext = {
   context: initialState,
   setContext: (context: IContext) => {}
}
const AppContext = createContext(initialContext)
export default AppContext
```

```
import {IReturnProject, IReturnUser} from "lbd-server"

interface CurrentProject extends IReturnProject {
  activeGraphs: string[],
  activeDocuments: string[]
}

interface IContext {
  user: IReturnUser | null,
  currentProject: CurrentProject | null,
  states: IPluginState[],
  plugin: string | null,
  predicat: IPredicat[],
  subject: ISubject[],
  selection: ISelection[]
}
interface IPluginState {
  [x: string]: any
}
interface ISelection {
  guid: string,
  [x: string]: any
}
interface IPredicat {
  p: string,
  [x: string]: any
}
interface ISubject {
  p: string,
  [x: string]: any
}
export {
  IContext
}
```

[46] Getting Started with React Redux | React Redux (react-redux.js.org)

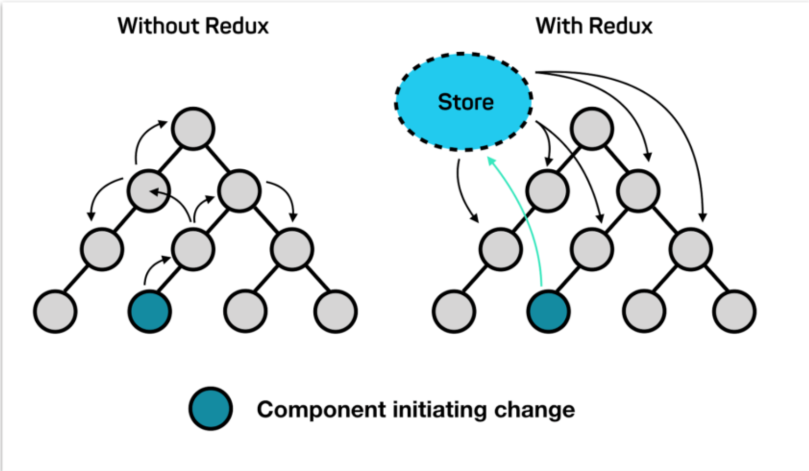


Figure 29 visual representation of the Redux store principles, which is fulfilled by the context file. Source: CodeCrunch (2021)

The repository is set up in the ProjectSetup.js file. This file contains the function submitProject (see Figure 30), in which the initial "store" state is updated with the input values provided by the user and then uploaded to the back-end, which handles the POST request, as described in section 5.1.5. The submitProject function is divided into three parts and uses the "React Hooks" system as support. Within the submitProject function, two hooks are used. Firstly, the UI uses the “useContext” hook, which contains the context index.ts properties and “useState”, which contains a Boolean value. These two hooks allow the dashboard to create a repository by updating the initial state and context files.

The first part of the submitProject function contains a variable named "result" that uses the user-defined repository name, description, label, and openness as parameters to be transferred to the backend. This variable contains the createProject hook coming from the lbd server. The second part fulfils the part of updating the initial "store" state, in which the context is updated with the hook setContext. A repository is created in the "store" containing the currentProject object, an empty array called activeDocuments which is essential for uploading the glTF file, and an activeGraphs empty array. The latter is of importance when uploading the TTL file. In the third part of the function, the project is activated with the setProjectCreated hook.

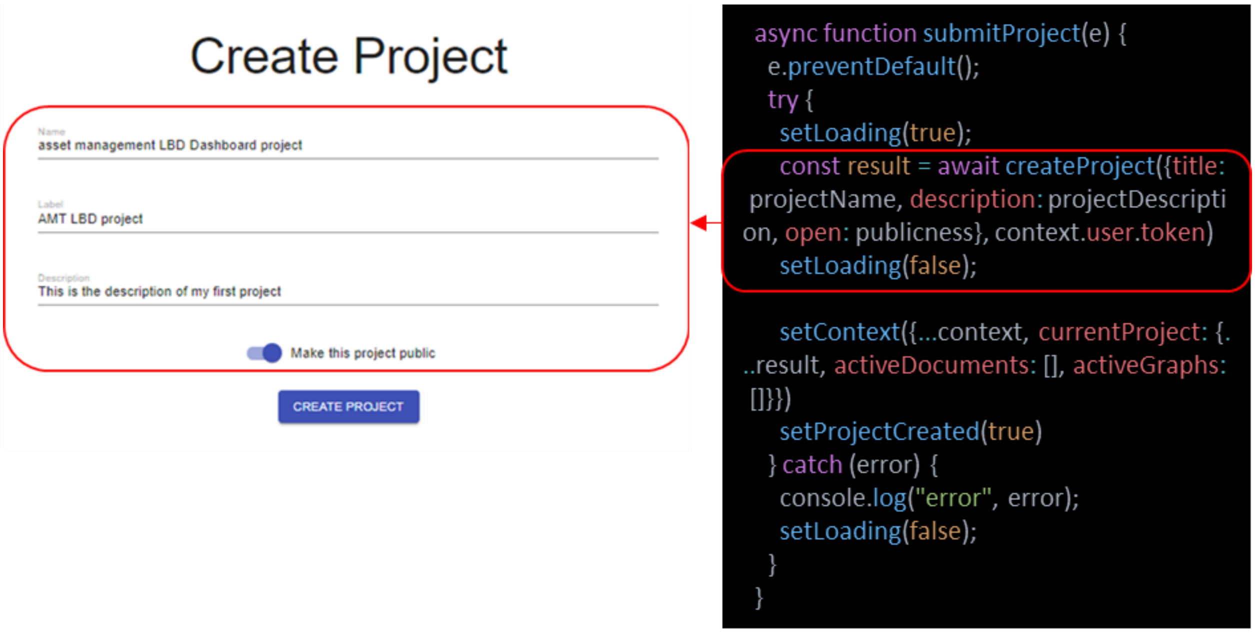


Figure 30 Front-end: Creating Repository

In the UploadDialog.js file, the various functions (uploadNewGraphToServer, UploadNamedGraphToServer, UploadIFCToServer and UploadDocumentToServer) handle the upload process of the documents in a similar way as the submitProject function. Figure 31 shows the uploadNewGraphToServer and uploadDocumentToServer functions and explains them in detail. The first part of this function consists of a variable that collects the form input in a response variable and based on the type of file it receives. This can be a glTF or TTL. It forwards these files to the appropriate API endpoint URL. The second part of this function consists of a variable "currentProject", which sets the response variable object context.currentProject to the appropriate file content. Once the await function returns successfully, the third part, "setContext", handles the update of the initial "store" context object called “currentProject” with the data obtained from the API call.

```
async function uploadNewGraphToServer() {
  try {
    const response = await uploadGraph(
      { label, description },
      context.currentProject.id,
      context.user.token);
    const currentProject = context.currentProject;
    currentProject["graphs"][response.uri] = respo
nse;
    setContext({ ...context, currentProject });
    return;
  } catch (error) {error.message = `Unable to create
new graph; ${error.message}`;
    throw error;
  }}
```

```
async function uploadDocumentToServer() {
  try {
    const response = await uploadDocument(
      {label,
       file: fileToUpload,
       description,
      }, context.currentProject.id, context.user.token);
    const currentProject = context.currentProject;
    currentProject["documents"][response.uri] = respon
se;
    setContext({ ...context, currentProject });
    return;
  } catch (error) {
    error.message = `Unable to upload document; ${erro
r.message}`;
    throw error;
  } }
```

Figure 31 uploadNewGraphToServer & uploadDocumentToServer functions

5.1.7 LBDServer: 3D viewer component

The 3D viewer component found at "GeometryComponent.js" is unlike the previous files set up as a class-based component. Therefore it does not make use of React Hooks to update and render components. Instead, this component uses the constructor method with a this.state function, in combination with various lifecycle methods, to achieve similar results. It also uses an external plugin from Xeokit-SDK[47]. Most of the written code is based on the documentation originating from Xeokit. The component uses lifecycle methods such as ComponentDidMount to render parts of the Xeokit code such as the viewer, NavCubePlugin, scene, and camera. The component is updated with ComponentDidUpdate, when it recognises that a user has uploaded a "glTF" file and thus updated the initial "store" state. Figure 32 showcases the before and after update of the component.

Figure 32 ComponentDidMount & ComponentDidUpdate lifecycle methods on Geometry component

[47] xeogl - Examples

5.1.8 Sensor Data Display component

In an ideal situation, in accordance with the literature and the methodology of other researchers such as Quinn et al. (2020) and Valinejadshoubi, Moselhi, Bagchi & Salem (2020), the sensor data would be captured directly from the Atlas Living lab in MongoDB via an API GET request, and then fetched to the front-end using Axios. However, the Atlas living lab API is still under development and is currently unavailable for this research. For this reason, the approach to the sensor data has shifted from processing the data through an API to capturing the data in a JSON format and displaying it in the charts. Each chart is enclosed in a "Tile" container component and is displayed in the dashboard.

For the charts, an external library "react-chartjs" is used. The JSON file consists of a variable with an array of objects. There are three objects which define the sensor data: “Sensor”, “Date”, and “Measurement”, in which the sensor ID, timestamp and values are included. Next, this data is imported into the TileBar or TileLine components, which pass the data to the GroupedBars and GroupedLineChart child component. Each chart is given a label "occupancy" or "temperature", and the data passed as a property. For the development of the charts, ChartJS[48] own developer documents are used. Using the Javascript .map method, it is possible to map over this sensor data and then pass it as props to the variables room and labels. This, in turn, allows for the data and labels to be passed as props to the child component to be displayed in the charts. Figure 33 shows how each chart is constructed. The result of the above is shown in Figure 34.

Figure 33 Constructing Charts with sensordata

```
const SensorData = [
    {
        "Sensor": {
            "Type": "Occupancy",
            "Space": 892,
            "Room": 8.128,
            "SensorID":  "11NR008LT-
001PIRTM",
        },
        "Date": {
            "Month": "January",
            "Day": 1,
            "Timestamp": '2021-01-
01 00:02:00',
        },
        "Measurement": {
            "Value": 20
        }
    },
```

```
export default class TileLine extends Component {
  state = {
      data: room128TempDataJanuary,
      labels: Timestamp1stJanuary,
      room126:
room126TempDataJanuary
      room127: room127TempDataJanuary
      AvgRoom128Jan: AvgRoom128Jan,
      AvgRoom127Jan: AvgRoom127Jan,
```

```
buildChart = () => {
      const myChartRef = this.chartRef.current.getCont
ext('2d')
      const { data, room126, room127, labels } = this.pr
ops;
      if (typeof myLineChart !== 'undefined') myLineCh
art.destroy()
    myLineChart = new Chart(myChartRef, {
         type: "line",
         data: {
             labels: labels.length === data.length ? labels :
 new Array(data.length).fill("Data"),
             datasets: [{label: "8.128", data: data, fill: false
, border
             Color: "#05464F" }, {label: "8.126", data: roo
m126, fill            : false, borderColor: "#28B8AB" }
```

[48] Chart.js | Open source HTML5 Charts for your website (chartjs.org)

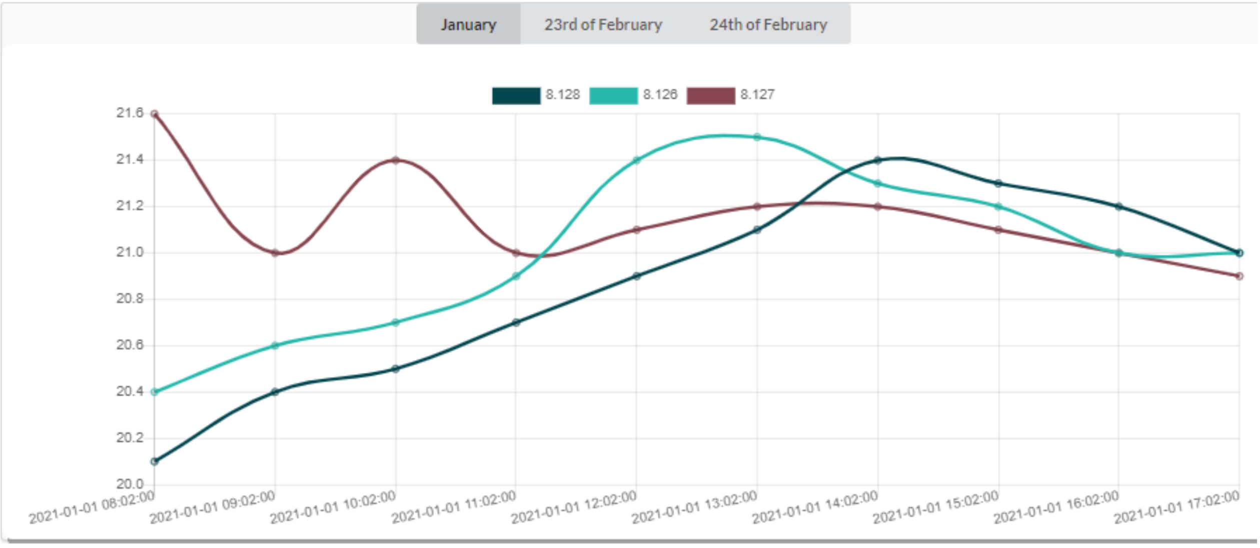


Figure 34 Example Line chart Component

5.1.9 Evaluating the Asset management Dashboard

The prototype is evaluated based on its functionality and whether it works as originally designed. Several design criteria were important here. First, the user must be able to upload files, create projects and a user profile. In addition, the user must be able to read sensor values for the occupancy in the building and internal climate. For this, an alpha test phase was carried out, which investigated whether the correct information could be displayed, the user could follow the correct application routes, and if the user could successfully create a profile and upload files. Table 8 below plots the initial criteria and reflects on what was achieved and what had to be changed.

Table 8 Reflection on initial dashboard goals

Initial requirements	*Success*	*Reason*	*Component*	*Workaround*	*Workaround Component*
The asset manager is able to access the application in the browser	Yes	N/A	App.js/ Index.js	N/A	N/A
The atlas living lab is accessible to the application	No	There was no API	N/A	Excel dump data is stored in JSON files	Mockup.js
Data from the living lab server is stored in MongoDB	No	There was no API	N/A	Excel dump data is stored in JSON files	Mockup.js
A connection is made with the sensor ID in the BIM model and MongoDB	No	No sensors were modelled in BIM model	QueryTable.js/ Querysparql.js/ Index.ts(context)/ contextInterface.ts	Manual input of sensor Id and semantic information in TTL file, the query uses the TTL file to display information in the viewer	Backend / QueryTable/ QuerySparql/ Index.ts
LBD server components for processing the glTF & TTL files are reused	Yes	N/A	Back-end + BrowserTabs.js	N/A	N/A
TTL file is processed and stored in GraphDB	Yes	N/A	Back-end + BrowserTabs.js	N/A	N/A
The glTF file is processed and stored in MongoDB	Yes	N/A	Back-end + BrowserTabs.js	N/A	N/A
The viewer component is reused from the LBDServer	Yes	N/A	LBDViewer.js	N/A	N/A

Display the occupancy sensor data and IEQ sensor data	Yes	N/A	groupedBars.js/ BarChartDash.js/ groupedLine.js/ LineChartDash.js	N/A	N/A
The application renders a tile component with a nested graph component	Yes	N/A	Tilebar.js / TileLine.js	N/A	N/A
The asset manager is able to make tickets about building components	No	insufficient time	Tickets.js	A component was made but remained static. It needs to be uploaded to the backend	TableParent.js / TableChild.js
The asset manager can interact with the graph and BIM model to get information about building elements and critical sensor data.	Yes	N/A	QueryTable.js	N/A	N/A
The asset manager can interact freely with the UI and is able to re-render the graph component, depending on the sensor that he selects in either the BIM model or button component	Yes	N/A	MyPopUp.js	N/A	N/A
The overviews are linked to a set of three buttons. Each button is for a single KPI.	Partially	No KPIs were used	MiddleTemp/ MiddleOccup/ MyPopUp	Buttons were made to select the date and read sensor output values	MiddleTemp/ MiddleOccup/ MyPopUp
If possible, no SPARQL Queries are conducted on the front-end UI	No	Too many dependencies	QuerySparql.js	Too many dependencies that make it impossible to separate SPARQL from the text input	QuerySparql.js
No RDF Graphs are displayed on the front-end UI	Yes	N/A	Back-end	N/A	N/A

In view of the above criteria, it is possible to look back on the development of the dashboard with success. Out of the 16 initial requirements, it was possible to implement 11 successfully. The dashboard operates in accordance with the established design criteria. However, during the alpha test phase, some limitations occurred. The dashboard has several design and functionality limitations. Firstly, it was not possible to connect to the Atlas API. This was due to the fact that no API was available for implementation. For this reason, no asynchronous API handlers were included in the dashboard that retrieves and store sensor data in MongoDB. Instead, the historical sensor data is captured in JSON files and inserted as a data file in the MockUp.js component.

In addition, it was not possible to establish a link between the obtained sensor IDs and the sensor IDs of the virtual sensors in the BIM model. One of the reasons for this is that they are not drawn in the BIM model, so they cannot be highlighted or queried. As a solution, the spaces are taken as a reference because these physical spaces do contain the sensors. In addition, the sensors are manually entered into the TTL file so that it is possible to perform a SPARQL query for these sensors. In turn, this uses the complementary guids of the spaces to highlight the spaces in the BIM model and, in turn, to present the sensor data in the QueryTable.js component. Several components were adapted and added for this purpose, including the QuerySparql.js, QueryTable.js, context (index.ts) and ContextInterface.ts components. All in all, it can be assumed that if the BIM model had the sensors at its disposal, it would have been possible to select them or execute a SPARQL query for them, highlighting these sensors.

Due to lack of time, it was also not possible to create a dynamic ticket creation component. This is currently too static in the application and only works as a visual representation. However, by creating a parent and child table component that can each make an API HTTP POST and GET request, it is possible to realise a ticket system for the asset managers that can dynamically create tickets based on the BIM model or the QueryTable and send them to the backend.

Finally, it was not possible to move the SPARQL queries from a text input field to automated buttons. This is partly because the SPARQL capabilities require specific inputs to highlight parts of the building. In addition, there is a large dependency behind the QueryTable that requires certain parameters to be passed to work. This dependency can be found in particular in the text input fields where a ?p or ?guid is required as a query input parameter in order to display the query table properly. Without these parameters, the query table cannot be displayed. Finally, it is possible to display sensor data in the dashboard based on three different dates, but these output values only follow the sensor data provided. No KPIs are included. Furthermore, the dates are set based on the data contained in the MockUp JSON file and do not consist of dynamic time-series data as desired initially. Furthermore, it was not possible to make the dashboard responsive so that it could be used on a mobile phone. Thus, the use remains for computer users only. Nevertheless, it is possible to use the dashboard as intended.

5.2 Conclusions

This chapter describes how the prototype was developed and then tested, what its limitations are and what its success is. The designed app layout focuses on the requirements as defined in the component framework from section 4.3. The focus was to develop both a front-end and a back-end. It is important for the user to be able to upload a BIM model and sensor data. The dashboard has been developed with an eye on future developments. For example, tabs have been included to create tickets about failing systems. In addition, multiple lines of code are included but not displayed in the UI. These lines of code are meant to migrate to the Redux system and further optimise the context and state management operations. The back-end handles the communications with the GraphDB store and MongoDB databases. For this, the front-end uses a .env environmental variable to communicate with the back-end. Next, the projectSetup, GeometryComponent, BrowserTabs, GroupedBars/GroupedLine, and Tile components enable various operations to be performed. The projectSetup component takes care of the communication with the back-end so that a project repository is created for the user. In addition, the BrowserTabs component ensures that the user is able to upload files in glTF and TTL format. The GeometryComponent helps when a glTF file is uploaded to display the 3D object, this was made possible by using the Xeokit library. Finally, the Tile components and the groupedBar- and groupedLinechart components ensure that the sensor data is displayed correctly. For this purpose, the chartjs library is used. The connection between the sensor data and the BIM model is made by evaluating whether the sensor ID property is present and matches the BMS data. The evaluation of the dashboard was carried out about its design and intended functionality. This revealed that the dashboard does not work optimally on several design and functionality principles. Out of the 16 requirements, five were not successfully implemented, two of them are due to the fact that there was no API available to build on, and there are some problems with the way the dashboard should execute queries. These are mainly dependency problems, which have resulted in the dashboard not being able to migrate from a text input field to a SPARQL on buttons.

6 Results: Case Study Atlas decision support

This chapter evaluates the dashboard in terms of its practical usefulness. For this purpose, the dashboard will be evaluated for its applicability to the Atlas case study. First of all, in this chapter, the sensor data will be evaluated for usability. Then the sensor data is processed in the TTL file and analysed. In addition, an analysis of the building performance of Atlas will be carried out based on the sensor data. Finally, this analysis will be reviewed, and an evaluation of Atlas on whether it meets the building performance requirements will be carried out.

6.1 Sensor Data Evaluation

The initial plan was to use an API to connect the sensor data coming from the Atlas Living Lab server to the dashboard. However, the API system architecture is still under construction and cannot be used at the moment. For this reason, another possibility has been chosen that uses historical sensor data. This concerns sensor data from 148 sensors, divided over 24 measuring moments per day for each sensor, whereby the total measurement was carried out over a period of 58 days, starting on 1/1/21 and ending on 28/2/21.

The historical sensor data from Atlas was provided in two Excel formats called "11RoomBSMHourJan & 11RoomBSMHourFeb". These spreadsheets are divided into one sheet for each day that contains all sensor data for that day. Figure 35 shows a simplified representation of a data sheet for 30 January 2021.

Timestamp	11NR008TE-001TRL	11NR008TE-003TRL	11NR008TE-004TRL	11NR008TE-005TRL	11NR008TE-010TRL
30-1-2021 00:02	21.8	21.8	21.4	21.8	20.8
30-1-2021 01:02	21.8	21.7	21.4	21.7	20.7
30-1-2021 02:02	21.6	21.6	21.2	21.6	20.5
30-1-2021 03:02	21.4	21.4	21	21.4	20.4
30-1-2021 04:02	21.3	21.3	20.9	21.3	20.4
30-1-2021 05:02	21.1	21.1	20.8	21.2	20.4
30-1-2021 06:02	21	21	20.8	21	20.4
30-1-2021 07:02	20.9	21	20.8	21	20.3

Figure 35 Simplified version of the data spreadsheet

The sensor data coming from the Atlas Living Lab was released in an Excel format. However, these data files are less than ideal to be displayed in the web browser. Nevertheless, this research needs to transform the sensor data. To achieve this, the sensor data must first be evaluated and dissected to better understand how it can be transformed.

The 148 sensors can be divided into seven categories. Namely TRL (temperature), CPA (adjustment on wall module), PIRTM (occupancy), CO2, FLW (flow), VTVRS (control signal supply of air), and VAVRS (control signal extraction of air). Each sensor ID has a similar structure, see Figure 36.

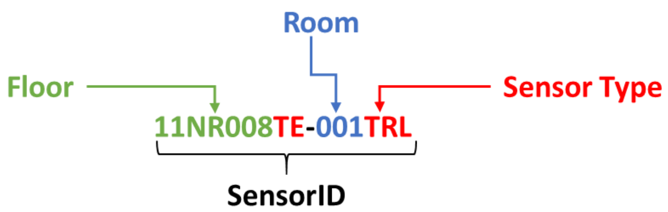


Figure 36 Deconstructing the SensorID

The sensor ID plays a central role in this research. However, there are some problems with the structure of this sensor ID. Namely, the "room" as it is referred to in the sensor ID does not match the actual physical room number or in the TTL file. Room 001 refers to room 8.128 in reality and space 892 in the TTL file. For this reason, all sensors have been inventoried. A complete inventory can be found in appendix G. The inventory shows that the sensor data is spread over 42 rooms. Each room has a temperature, CPA wall module and occupancy sensor by default. This does not apply to the larger common areas 8.201, 8.326, 8.324, 8.323, 8.140, and 8.445. These areas also have a CO2, FLW, and control signal sensors for the intake and exhaust of the air.

Since the data needs to be implemented directly into the dashboard, the Excel format needs to be transformed into a CSV or JSON format. However, there are several problems with this approach, which can be found mainly in the structure of the Excel sheet. All data is spread across different spreadsheets and needs to be bundled into a single-sheet CSV file. This requires a programming approach in order to gain the ideal format. As a solution, a python script was used with pandas & NumPy, which bundles all Excel sheets based on a number of parameters.

Nevertheless, the CSV format is less than ideal. Using a CSVtoJSON[49] file converter, it turns out that, due to the original structure of the overall data file, the values read in by the sensors are misclassified. Namely, they are immediately adjacent to the timestamp. In addition, the term "timestamp" is next to the name sensor ID, which should hold the string value sensor ID. Figure 37 shows a fragment of the extracted JSON file compared to the desired format.

Figure 37 Transforming the JSON file

Intitial format

```
{
    "Timestamp": "11NR008LT-
001PIRTM",
    "2021-01-01 00:02:00": 2,
    "2021-01-01 01:02:00": 2,
    "2021-01-01 02:02:00": 2,
    "2021-01-01 03:02:00": 2,
    "2021-01-01 04:02:00": 2,
    "2021-01-01 05:02:00": 2,
    "2021-01-01 06:02:00": 3,
    "2021-01-01 07:02:00": 3,
    "2021-01-01 08:02:00": 3,
```

Desired format

```
[
{
    "Sensor": {
        "Type": "Occupancy",
        "Space": 892,
        "Room": 8.128,
        "SensorID":  "11NR008LT-001PIRTM"
    },
    "Date": {
        "Month": "January",
        "Day": 1,
        "Timestamp": "2021-01-01 08:02:00"
    },

    "Measurement": {
        "Value": 3
    }
},
```

With 148 sensors generating output 24 hours a day for a period of 58 days, this would mean that the file would have to be manually modified 206,016 times. For this reason, it was decided to select a few sensors for the study and use them in the dashboard. Since the interviews with the asset managers showed that the occupancy and indoor climate sensors are considered the most important sensors to be read, work will continue with these sensors.

Multiple JSON files were generated, one for the occupancy and one for temperature sensors. The temperature sensors were chosen because, of all the IEQ sensors, they are the only ones that appear by default in all rooms. For each file, three sensors were selected, namely sensors 001, 003, and 004 (PIRTM/TRL). This is a measurement for rooms 8.128, 8.127, and 8.126. In addition, the time period considered is not 24 hours. Instead, it was chosen to measure the sensors from 8:00 a.m. until 5:00 p.m., as these are the main peak hours. The reason for this is that the occupancy sensor data records movement during the evening hours, even if the building was closed from 9:00 p.m.

[49] CSVJSON - CSVJSON

until 6:00 a.m. These values should read 0, as is the case for the Flow and CPA sensors, because during these hours, no occupants can be present in the building, see Figure 38. For this reason, only the peak hours are used. So from this point on, benchmark values will have to be used to find out the difference between the actual number of people in the rooms and the default values read from the data source files, more on this in section 6.3

Flow & CPA Sensors


```
[
  {
    "Timestamp": "11NR008FT-013FLW",
    "2021-01-01 00:02:00": 0,
    "2021-01-01 01:02:00": 0,
    "2021-01-01 02:02:00": 0,
    "2021-01-01 03:02:00": 0,
    "2021-01-01 04:02:00": 0,
    "2021-01-01 05:02:00": 0,
    "2021-01-01 06:02:00": 0,
    "2021-01-01 07:02:00": 0,
    "2021-01-01 08:02:00": 0,
```

```
{
    "Timestamp": "11NR008TE-003CPA",
    "2021-01-01 00:02:00": 0,
    "2021-01-01 01:02:00": 0,
    "2021-01-01 02:02:00": 0,
    "2021-01-01 03:02:00": 0,
    "2021-01-01 04:02:00": 0,
    "2021-01-01 05:02:00": 0,
    "2021-01-01 06:02:00": 0,
    "2021-01-01 07:02:00": 0,
    "2021-01-01 08:02:00": 0,
```


Occupancy Sensors


```
{
    "Timestamp": "11NR008LT-
001PIRTM",
    "2021-01-01 00:02:00": 2,
    "2021-01-01 01:02:00": 2,
    "2021-01-01 02:02:00": 2,
    "2021-01-01 03:02:00": 2,
    "2021-01-01 04:02:00": 2,
    "2021-01-01 05:02:00": 2,
    "2021-01-01 06:02:00": 3,
    "2021-01-01 07:02:00": 3,
    "2021-01-01 08:02:00": 3,
```

```
{
    "Timestamp": "11NR008LT-
003PIRTM",
    "2021-01-01 00:02:00": 2,
    "2021-01-01 01:02:00": 2,
    "2021-01-01 02:02:00": 2,
    "2021-01-01 03:02:00": 2,
    "2021-01-01 04:02:00": 2,
    "2021-01-01 05:02:00": 2,
    "2021-01-01 06:02:00": 3,
    "2021-01-01 07:02:00": 3,
    "2021-01-01 08:02:00": 3,
```


Figure 38 Inconsistencies in Data

6.1.1 Sensor Data TTL upload

The sensor data must be made semantically rich. The connection from the JSON file to the front-end stands. However, the TTL file of Atlas must be provided with the correct sensors. This is done by describing the sensors in the TTL file, providing them with a sensor ID and then linking them to the appropriate space. Because spaces 8.128, 8.127, 8.126 are involved, the sensors are inserted in spaces 892, 1023, and 1144.

A full description of the temperature and occupancy sensors can be found in appendix H, and a sample can be found in Figure 39. The sensors are described according to the BOT ontology. Each instance has been given several fundamental properties that can be queried with a SPARQL query. Every instance of sensors has been given its own GUID, its own tag, and its own sensor ID that matches the sensor ID from the original Excel sheet.

Figure 39 Connecting the Sensors to the Turtle File

Description of sensor

```
inst:TemperatureSensor_11NR008TE_001TRL
a bot:Element ;
rdfs:label "Signify Smart Energy Savings Sytems : Signify
Temperature Sensor: 001TRL"^^xsd:string;
rdfs:comment""^^xsd:string;
bot:hasGuid"B1300825-17FD-5274-000A-
ED70F7C0A5CB"^^xsd:string;
props:tag"001TRL"^^xsd:string;
props:reference"Signify Temperature Sensor"^^xsd:string;
props:reference"Signify Temperature Sensor"^^xsd:string;
props:level "Level: Ceiling 8th floor"^^xsd:string;
props:sensorID"11NR008TE-001TRL"^^xsd:string;
```

Sensor placement in Space

```
inst:space_892
    a bot:Space ;
    bot:hasGuid "1456e859-2e9e-4e70-9341-
f073adb2dbb0"^^xsd:string ;
    props:hasCompressedGuid "0KLkXPBfvES9D1y7Eji
jkm"^^xsd:string ;
    bot:containsElement inst:CPASensor_11NR008TE
_001CPA, inst:TemperatureSensor_11NR008TE_001TRL
;
```

The general idea is to link each sensor to the space in the TTL file. In Figure 39, space 892 (room 8,128) has a unique guid "0KLkXPBfvES9D1y7Eji", which is linked to that same element as it is defined in the glTF file. This allows for a connection between the dynamic sensor data and the semantically rich TTL file. A SPARQL query or a selection in the front end displays which sensor is selected and to which space this sensor belongs. However, there are a few problems with this current approach. The glTF file as provided by the TU/e does not contain any spaces. Therefore, it is not possible to perform a SPARQL query to display the sensors for the selected spaces. In order to solve this problem, the original IFC file has to be transformed. This can be done by using a software tool from Blender called BlenderBIM[50]. A direct export from BlenderBIM to Collada can be realized. Subsequently, using a method similar to that of Werbrouck, Malcolm and Pauwels (2021), one can use the Khronos group Collada2GLTF[51] tool to create a new glTF file in which the spaces are included. However, this method has a certain number of limitations. For example, data is lost during the conversion from BlenderBIM to collada. The IFC name is transferred under the meshes in the glTF file instead of the guids, so the rooms are not provided with the necessary guids. This required a manual insertion of the guids under the correct rooms, see Figure 40.

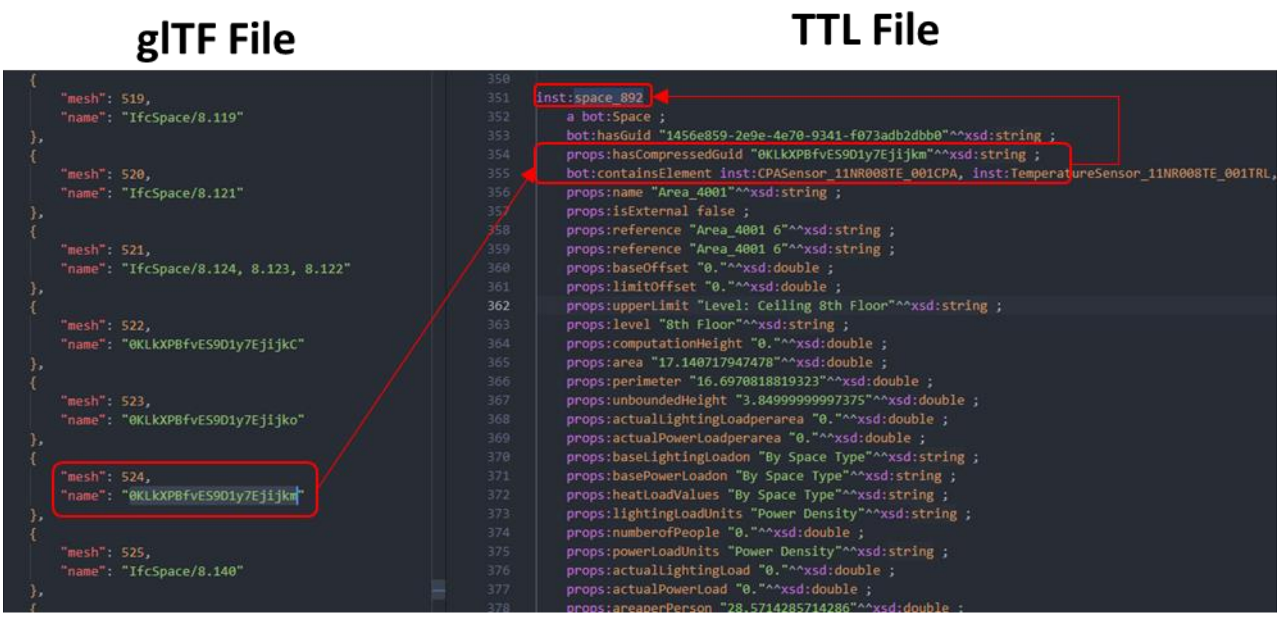


Figure 40 glTF connection with TTL through guids

6.2 Dashboard & GraphDB SPARQL

A current functionality in the dashboard is the execution of a SPARQL query in the front end. This is mainly done in the "QuerySparql.js" component, which sends out a POST and GET request to graphDB and MongoDB by means of an asynchronous API call. GraphDB then checks whether the token, parameters and repository are correct and returns values from the TTL file. The viewer and the TTL file are connected to each other by means of the guids. Each element in the building is given a guid, and these elements are then described with their guids in the TTL file.

[50] (blenderbim.org)
[51] GitHub - KhronosGroup/COLLADA2GLTF: COLLADA to glTF converter

This component and especially the executeQuery function contained within it require a number of parameters, including an “initialQuery”, as well as the context component where the “initialState” is stored. The executeQuery verifies the user performing the query and uses the token associated with the user to find the correct project directory. It then creates a variable result that stores the response of the query. The response consists of the id, graph, token and query. Next, the response is captured within the context component´s parameter called "selection" by pushing this value into the array. This ensures that the user is able to retrieve the values from the query and allows for the dashboard to highlight the objects in the Viewer, see Figure 41.

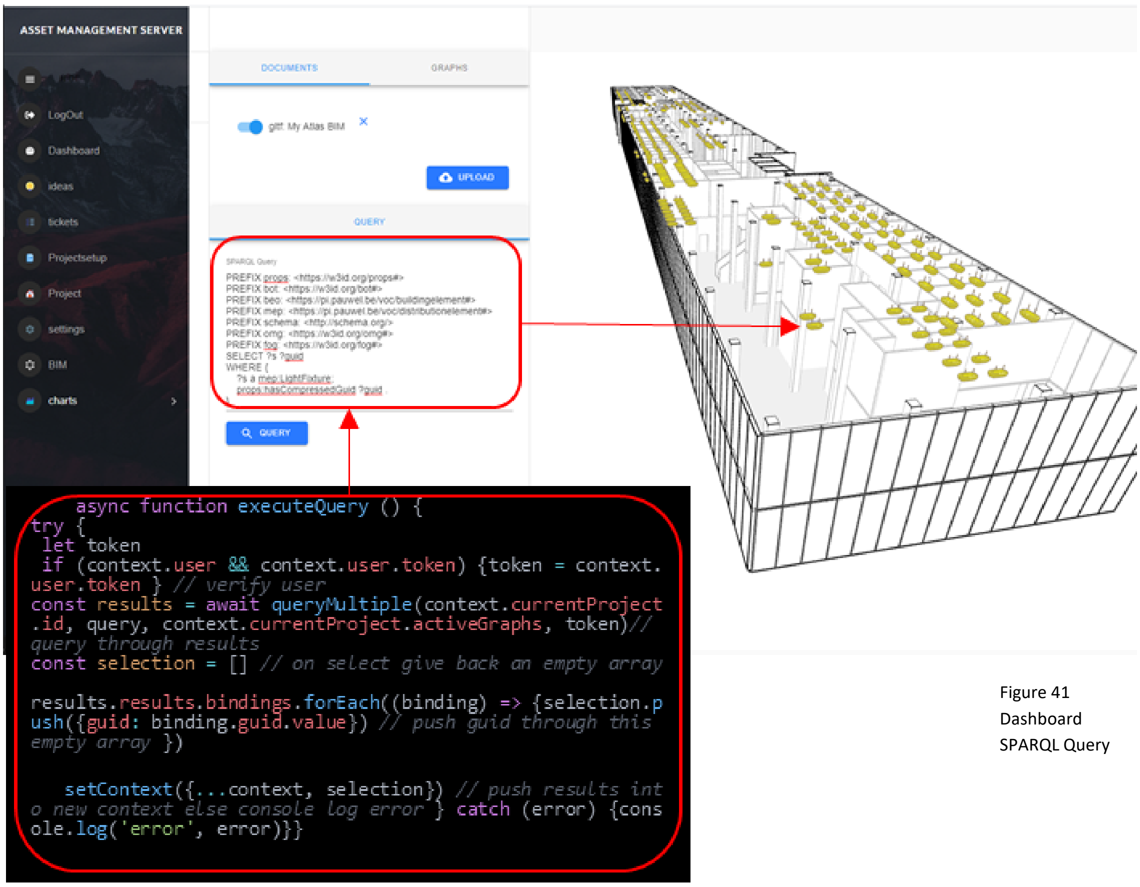


Figure 41
Dashboard
SPARQL Query

In addition to the SPARQL capabilities in the dashboard, it is also possible to SPARQL query from http://localhost:7200 directly in the GraphDB database, see Figure 42. Hence, an evaluation of the sensor data is required, this is possible by executing the following query using the PROPS ontology on the PROPS: SensorID property.

Figure 42 SPARQL Query and Results

The results show that the sensors have been successfully inserted into the TTL file. This has established the final connection, meaning that the sensors are now assigned to a space, and it is possible to exploit the additional knowledge now associated with these spaces.

By using the visual graph in GraphDB, it is possible to see what other relationships the spaces have. Figure 43 shows an example of Space_892 (physical space 8.128). In this example, the space is located on the 8th floor and in addition, the space has a total of eight sensors, divided into four light sensors and a single conductance sensor. A previous researcher added these sensors. Finally, three sensors have been inserted for this space during this research. These are the CPA wall module sensors, the occupancy sensors, and the temperature sensors, all of which are highlighted in blue.

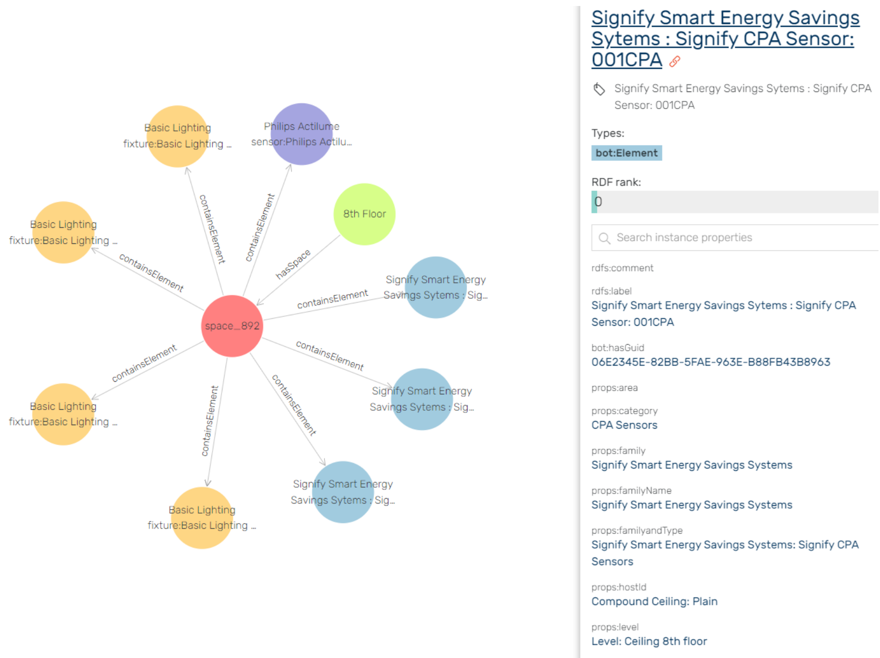


Figure 43 Visual Graph seeing the connections with other elements

6.3 Assessing the building performance of Atlas

The very last requirement in this thesis is to find out whether an asset manager can evaluate the building performance based on the dashboard. In this thesis, this means whether the asset manager is able to measure the building comfort and energy performance, with an emphasis on comfort. After all, the interviews with the asset managers have shown that for them, building performance means user satisfaction as a result of the comfort provided by the asset. The indoor environmental quality is leading in this regard. This consists of measuring the air temperature, relative humidity, CO2, and lighting strength in contrast to the occupancy. Given the provided data, an assessment can be made for the temperature, air supply, occupancy, and CO2 in the building. However, the choice has been made in the dashboard only to show the temperature values and occupancy. Yet, for a full assessment, the entire Excel will still be examined, but only to support the assessment based on the dashboard.

Based on the findings of Korsavi, Montazami & Mumovic (2020), a distinction can be made between an assessment in the summer and an assessment in the winter. This will cause the target values to differ. The sensor data given concerns the months of January and February, which means that an assessment based on the winter will follow. For this, the target values for the CO2 levels must lie either below or between 1180 and 1310 PPM, for the temperature, this concerns values below or between 22.8 and 24.2 degrees Celsius, the airspeed at temperatures below 23 degrees amounts to 0.2 m/s (ANSI/ASHRAE 55-2017). Finally, under normal circumstances, the occupancy rate should be calculated over the m² (Li et al., 2018). However, due to the corona pandemic, a minimum amount of people are allowed within Atlas. Because of this, there is a predefined maximum number of people allowed in each room. For rooms 8.128, 8.127 and 8.126, this amounts to two persons at all times. In addition, based on the interviews carried out, it can be concluded that some target values for Atlas can be adjusted. Most notably, the CO2 levels should be below 800 PPM, and the temperature should fluctuate between 20-23 degrees with an ideal temperature value of 21 degrees Celsius.

For the analysis, three dates are used in total, which are also included in the JSON file in the dashboard. These are the 1st of January 2021, 23rd of February 2021 and the 24th of February. These dates were chosen because the building was closed on the 1st of January due to the Christmas holidays, so the building was mostly empty, and most of the sensors should not have been triggered during this time. This makes it possible to use the 1st of January as the baseline value for the analysis. This means that the readings of the 24th of February and 23rd of February can be compared to those of the 1st of January.

By using Excel as a basis for analysing the sensor data, several things stand out immediately. First of all, for the temperature and CO2 data, there are no remarkable differences between the readings on the 1st of January and the other two dates. However, there are certainly significant discrepancies in the readings for the occupancy, flow sensors, and air regulatory sensor data. For the complete analysis, see appendix I. A remarkable feature of the occupancy rate is that default values are read by all sensors on the first of January. From 23:02 until 05:02, all sensors read a value of 2, and from 06:02 until 22:02, it is three see Figure 44.

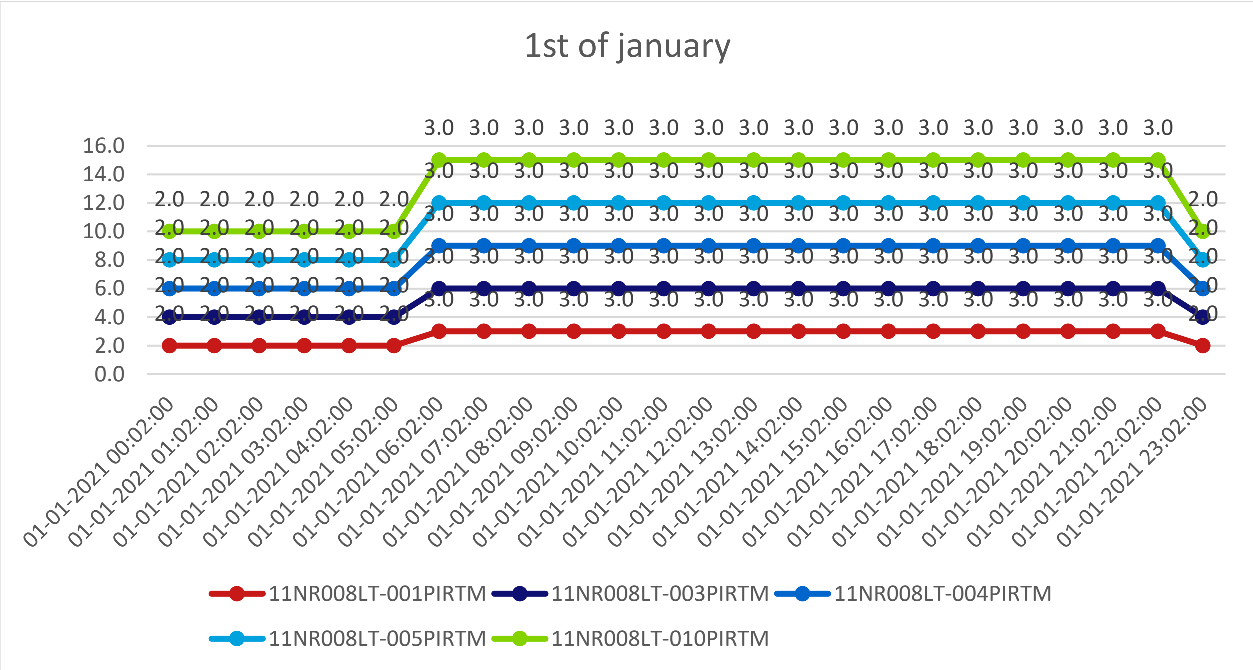


Figure 44 Discrepancies in occupancy sensor readings

Thus, these values indicate that there are people present at the time, which is highly unlikely. On the other hand, for the other two dates, the values fluctuate between the default values seen in Figure 44 and five. In view of the current circumstances regarding corona, where only two people may be present in each room, it can be concluded that the values read for the occupancy rate are only accurate when they are subtracted from the base value for the rooms. So, if the sensor data indicates 5, and the default value at that time is 3, then it indicates that there are two people in the room. However, in an ideal situation, the sensors would output a value of 0 when nobody is present. This would also simplify the direct entry into the dashboard because, with the current state, it must be taken into account that there is a deviation in the sensor data and how this should be presented in the graphs.

The same assessment and calculation are also required for the flow and air supply/exhaust sensors. Namely, the flow sensor with sensor ID 11NR008FT-040FLW read a value of 28 till 30 by default, while all other flow sensors read a value of 0 by default. The air supply and exhaust sensors read respectively for the sensors 11NR008RS-013VT2Rs, 11NR008RS-013VT4RS and 11NR008RS-013VTVRS 100 as default values and for the sensors 11NR008RS-013VT3RS 87 and 11NR008RS-038VTVRS 88.9 as default values. On normal days when people are present in the building, the air supply and exhaust sensors have values between 30 and 50 for all sensors. This means that the difference is reduced by 50 - 70. For the flow sensors, all sensors go from low values such as 0 and 30 to at least 64 up to 270, depending on the sensor location. A small sample of the value differences in the sensors is shown in Table 9.

Table 9 Differences in values between Sensor Values

	Flow Sensors						Air supply and exhaust sensors				
	Timestamp	11NR008FT-013FLW	11NR008FT-038FLW	11NR008FT-039FLW	11NR008FT-040FLW	11NR008FT-301FLW	11NR008RS013VT2RS	11NR008RS-013VT3RS	11NR008RS-013VT4RS	11NR008RS-013VTVRS	11NR008RS-038VTVRS
January 1st	01-01-2021 10:02:00	0.0	0.0	0.0	30.0	0.0	100.0	87.3	100.0	100.0	89.9
	01-01-2021 11:02:00	0.0	0.0	0.0	30.0	0.0	100.0	87.3	100.0	100.0	89.9
	01-01-2021 12:02:00	0.0	0.0	0.0	30.0	0.0	100.0	87.3	100.0	100.0	89.9
	01-01-2021 13:02:00	0.0	0.0	0.0	30.0	0.0	100.0	87.3	100.0	100.0	89.9
	01-01-2021 14:02:00	0.0	0.0	0.0	28.0	0.0	100.0	87.3	100.0	100.0	89.9
	01-01-2021 15:02:00	0.0	0.0	0.0	30.0	0.0	100.0	87.3	100.0	100.0	89.9
23rd of february	23-02-2021 10:02:00	67.0	114.0	82.0	116.0	135.0	22.2	34.3	27.6	29.1	39.0
	23-02-2021 11:02:00	64.0	114.0	82.0	116.0	134.0	22.2	34.3	27.6	29.1	39.0
	23-02-2021 12:02:00	66.0	114.0	82.0	116.0	139.0	22.2	34.3	27.6	29.1	39.0
	23-02-2021 13:02:00	68.0	113.0	81.0	115.0	136.0	22.2	34.3	27.6	29.1	39.0
	23-02-2021 14:02:00	67.0	117.0	82.0	115.0	139.0	22.2	34.3	26.6	29.1	40.3
	23-02-2021 15:02:00	64.0	118.0	82.0	116.0	136.0	22.2	34.3	26.6	28.3	40.3
24th of february	24-02-2021 10:02:00	67.0	115.0	82.0	267.0	132.0	21.9	34.1	26.9	29.0	39.2
	24-02-2021 11:02:00	64.0	114.0	80.0	113.0	136.0	28.0	37.9	31.0	30.8	39.2
	24-02-2021 12:02:00	66.0	115.0	82.0	115.0	134.0	28.0	37.9	31.0	30.8	39.2
	24-02-2021 13:02:00	66.0	116.0	82.0	115.0	136.0	28.0	37.9	31.0	30.8	39.2
	24-02-2021 14:02:00	66.0	117.0	84.0	115.0	138.0	28.0	37.9	31.0	30.8	40.5
	24-02-2021 15:02:00	67.0	269.0	80.0	115.0	136.0	28.0	37.9	31.0	30.8	56.1

Taking into account all the above information, an individual assessment can now be made for rooms 8.128, 8.127 and 8.126 based on the comfort of the user. In these rooms, there are no CO2, flow or air supply and exhaust sensors. These sensors are only found in large common areas. However, it can be assumed that the sensor values read from these sensors apply to the entire floor area.

By filtering on the number of people present in each room, the data file can be reduced significantly, see table 10. Next, the number of people per room is classified based on the time that they were present. This makes it possible to read the CO2, flow and air regulation sensor values. It is striking that the number of people present over a long period of time was higher during the peak hours on the 23rd and 24th of February, from 10 a.m. until 2 p.m. In all cases, two people were present in all rooms. For room 8.127, there were only two people present on the 23rd of February at 16:02 for the entire day. As a result, the assessment for this room has the highest error rate because only a single value is read at one moment. For rooms 8.128 and 8.126, an average temperature of 23.3 & 23.4 was recorded on the 24th of February and 22.8 & 21.9 on the 23rd of February. These values are two degrees higher than the ideal values that should be used according to the asset manager of the TU/e. However, according to the literature, the temperatures met the target values for the winter period on both days. Thus, the temperature can be considered as "comfortable for the user". The CO2 values averaged in all days between 430 and 433 PPM, far below the maximum target value of 800 set by the TU/e itself and the 1180 derived from the literature. So, as far as the CO2 is concerned, the space is experienced as comfortable.

PIRTM 001 (8.128)					
TimeStamp	Occupancy	Temperature	CO2	Flow	Air (supply/exhaust)regulatory sensor
24-02-2021 10:02:00	2.0	23.2	436.0	67.0	21.9
24-02-2021 11:02:00	2.0	23.4	435.0	64.0	28.0
24-02-2021 13:02:00	2.0	23.2	432.0	66.0	28.0
Average 24th		**23.3**	**433.0**	**65.8**	**26.4**
23-02-2021 10:02:00	2.0	22.0	434.0	67.0	22.2
23-02-2021 11:02:00	2.0	21.9	435.0	64.0	22.2
23-02-2021 12:02:00	2.0	22.1	431.0	66.0	22.2
23-02-2021 13:02:00	2.0	22.1	426.0	68.0	22.2
23-02-2021 14:02:00	2.0	22.5	438.0	67.0	22.2
23-02-2021 15:02:00	2.0	22.6	427.0	64.0	22.2
23-02-2021 16:02:00	2.0	22.9	430.0	64.0	22.2
23-02-2021 17:02:00	2.0	23.5	428.0	64.0	22.2
Average 23rd		**22.7**	**428.5**	**64.7**	**24.4**

PIRTM 003 (8.127)					
TimeStamp	Occupancy	Temperature	CO2	Flow	Air (supply/exhaust)regulatory sensor
23-02-2021 16:02:00	2.0	23.7	430.0	117.0	34.3

PIRTM 004 (8.126)					
TimeStamp	Occupancy	Temperature	CO2	Flow	Air (supply/exhaust)regulatory sensor
24-02-2021 11:02:00	2.0	23.5	435.0	80.0	27.6
24-02-2021 13:02:00	2.0	23.5	432.0	82.0	27.6
24-02-2021 16:02:00	2.0	23.3	429.0	82.0	26.6
Average 24th		**23.4**	**432.0**	**81.3**	**27.3**
23-02-2021 06:02:00	2.0	21.7	433.0	82.0	30.2
23-02-2021 07:02:00	2.0	21.7	432.0	82.0	31.2
23-02-2021 08:02:00	2.0	21.8	428.0	84.0	23.0
23-02-2021 10:02:00	2.0	22.1	434.0	82.0	22.2
23-02-2021 11:02:00	2.0	22.2	435.0	82.0	22.2
23-02-2021 13:02:00	2.0	21.6	426.0	81.0	22.2
Average 23rd		**21.9**	**431.3**	**82.2**	**25.2**

Table 10 analyzing the building performance for each room

6.3.1 Cross-referencing the analysis with the Dashboard.

Based on the analysis from the Excels, it can be concluded that the rooms can be considered comfortable. However, the values as they are formed from the above analysis should also follow from the dashboard. This can be done by conducting a SPARQL query in the dashboard that pre-selects the rooms and has the following as its parameters:

```
PREFIX props: <https://w3id.org/props#>
PREFIX bot: <https://w3id.org/bot#>
PREFIX inst: <http://linkedbuildingdata.net/ifc/resources20201208_005325/>
select * where {
inst:space_892 ?p ?guid .
} limit 10000`
```

This query contains a pre-condition for the dashboard to pass before it can return a query table. The pre-conditions are that the query must include *?p* and *?guid* as its parameters. Done correctly, this SPARQL query will highlight space 8.128 in the dashboard and return a query table containing the semantically rich properties of the space. A smaller view of the entire query table can be seen in Figure 45. The added sensors described in section 6.1.1 can also be found here. It is important that the correct sensors are returned. In the above SPARQL example, the dashboard should return the sensors with the following IDs 11NR008TE_001TRL, 11NR008LT_001PIRTM and 11NR008TE_001CPA. From Figure 45, it can be observed that the dashboard displays the correct output.

Figure 45 Query table output SPARQL

It shows the suitable sensors associated with the correct query and thus the proper graphs. By clicking open the temperature sensors, it shows the following graph see Figure 46. It corresponds identically to the graphs shown in the Excel analysis. This also applies to the occupancy sensors. This enables an asset manager to draw the same conclusions from the dashboard as he or she would from analysing the Excels. This completes a vital pre-condition for this thesis.

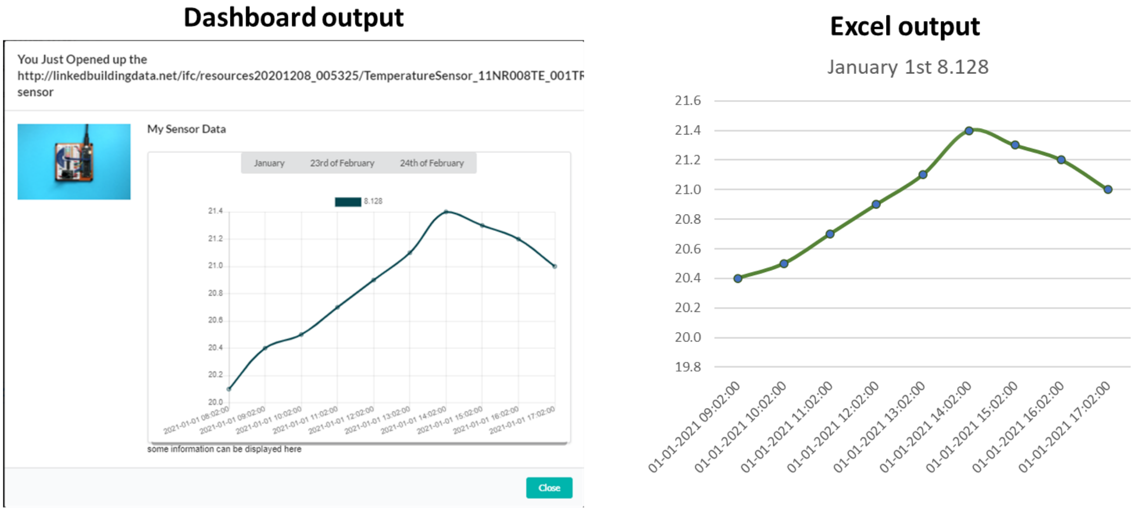


Figure 46 Cross-referencing the sensor graphs from the dashboard with the excel

6.4 Conclusion

In conclusion, several things can be noted. First of all, concerning the sensor data. Since it was not possible to establish an API connection with the living lab, the desired dynamic structure for the sensor data fell away and had to make way for historical sensor data. As a result, the format in which the sensor data is delivered is less than ideal. The format consists of a two-part excel structure in which 148 sensors display sensor data over a period of 58 days and a 24-hour interval in 58 spreadsheets. The problem with this is that the sensor data is not bundled and not easily machine-readable. To solve this, first, the spreadsheets from the files were bundled and transformed into a CSV format, which in turn was transformed into a JSON format using a CSV to JSON converter. This made the format suitable for use in the dashboard and made it possible to gain more insight into what problems there are within the sensor data structure. The JSON file still has errors after the conversions, mainly because the original source file was written in a not so ideal format.

For this reason, it was decided to convert the converted JSON file into a hardcoded JSON format that can be used directly for the graphs. Finally, the sensors were described using the BOT and PROPS ontologies and were inserted into the TTL file under the spaces. This makes it possible to use a SPARQL query to check whether the sensors have been made semantically rich and whether they each have a subject, predicate object description.

The reason for this connection stems from the fact that the spaces are provided with suitable guids and are modelled in the BIM model. As a result, a connection to the BIM model can be established on the basis of the matching guides found in both the glTF and TTL file. Contrary to the individual sensors, which are not modelled in the BIM model. As a result, they have not been assigned any guids in the glTF file, only in the TTL file, which means that this matching connection is missing. Therefore, these individual sensors cannot be highlighted in the BIM viewer. However, for each space on the eighth floor, a description called bot:containsElement has been inserted. It is under this ontology description that all the corresponding sensors are inserted for each space. In concrete terms, the TTL file describes that each room has a certain set of sensors.

Using the connection in the TTL file, it is possible to execute a SPARQL query in the front-end for the individual spaces within Atlas. The SPARQL query has the requirement that it must have at least as parameters *?p* and *?guid* to return a query table. This is a less than ideal condition to have, and it is inconsistent with the initially set requirement described in section 4.2. The preferred method is a button driven predefined SPARQL query. At present, the asset manager must have knowledge of SPARQL in order to use the dashboard to its full extent. Nevertheless, the dashboard shows that it can successfully execute a query and display the correct sensor data.

The sensor values are read out for three dates. First of all, January 1st 2021, is used as a baseline date. In addition, values for February 23rd and 24th are also readout. Therefore, an assessment can be made that is as complete as possible. From the sensor data provided, it can be concluded that the sensor readings are not accurate for all sensors. For the occupancy, flow and air regulator sensors, there are significant differences in the values found. This applies especially to the occupancy sensors, as every space indicates that people are present while this is not possible for the relevant times.

Nevertheless, it is possible to assess the different spaces as to whether they meet the building performance criteria, as defined in the literature and the interviews. The assessment in both the dashboard and Excel shows that the building meets all the building performance criteria, which mainly focuses on whether the building offers comfort to the users. For example, Atlas, with an average value of 430 PPM for the CO2 values, is far below the target value of 800 PPM set by TU/e and the minimum values mentioned in the literature. The temperature values in the individual rooms were indeed above the comfort values set by the TU/e. However, in all rooms, they meet the standard set in the literature. It also shows that the building can be considered comfortable. All in all, it can be assumed that the asset manager can make an assessment of the building performance based on the dashboard, thus completing the main objective of this thesis.

7 Discussion

In this chapter, the overall findings are evaluated. In particular, it will examine whether the results meet the expectations and whether there are alternative explanations for the findings. Subsequently, the implications are discussed in terms of whether the research aligns with the existing theoretical framework and what the practical implications of this research are. Lastly, the limitations of this study are discussed.

7.1 Interpretation and implications of the results:

To answer the main question, the literature review primarily looked at how previous studies looked at concepts such as asset management and what tasks they perform, what asset management is within a BIM environment, and especially how previous researchers have connected BMS sensor data to a static BIM model. The study by Quinn et al. (2020) shows that BMS and BAS systems are great for monitoring building conditions, and while this is true to a certain extent, it is less than ideal in this study. The interviews with the asset managers show that if the asset user does not understand the building because the equipment is not working properly or because the user is not able to read the correct temperature value, this can lead to a wrong perception of the building and, therefore to a lower level of satisfaction with the building. Section 4.1 shows that asset managers strive to make the purpose of the building clear to the end user. If a user fully understands the building, complaints can be prevented and thus user satisfaction improved. The Atlas case study shows that some sensors generate incorrect output values, so the findings of Quinn et al. (2020) with regard to how well the BMS works for monitoring the building can be called into question, as the wrong output value can lead to wrong policy choices being made. In addition, from the studies conducted by Quinn et al. (2020) and Valinejadshoubi, Moselhi, Bagchi & Salem (2020), the initial starting point for this study was to use Dynamo in order to connect the time-series data to the BIM model. This initial expectation turned out to be incorrect. This study proves that complex integrations of this kind do not have to be used at all. Instead, a link was established based on the BOT and PROPS ontologies in the TTL file. The sensors are manually inserted into this file and linked to the BIM model based on the spaces they are located in. These spaces have the same guids in both the glTF and TTL files and are connected to each other through these guids. By running queries on these spaces, it is possible to retrieve the sensors and their data from the query tables. However, the sensor data can be directly connected to the dashboard via MongoDB, FluxDB and Axios, as shown in section 2.8.4. This preferred approach of using APIs to form a connection would have been recommended for similar studies such as this one, rather than the current methods used in this study or those of Quinn et al. (2020) and Valinejadshoubi, Moselhi, Bagchi & Salem (2020). Nevertheless, as described in section 5.1.8, this integration is impossible without a provided API and has been excluded from this study for the time being.

Central to all literature studies encountered is the use of an ontology to link the sensor data with their sensor IDs to the BIM model. The ontologies vary per study. In this study, the BOT and PROPS ontology were used. Some researchers wrote their own ontology, such as Kim et al. (2018), allowing them to describe more properties of the sensors. Which in turn increases the semantic richness of the described sensors. However, writing your own ontology like the FMontology by Kim et al. (2018) is not necessary in the first place, in the first instance researchers should use existing ontologies. In retrospect, for this research and dashboard, it would have been far more interesting and effective to use the existing ontologies for describing the sensor data such as the SSN ontology or Kim et al. (2018) FMontology. By using these ontologies, more properties and definitions can be assigned to the sensor data, which would make it possible to extract more sensor-related information from the query tables.

The results from the interviews are particularly surprising and of great value. The interviews reveal that almost all asset managers make little or no use of KPIs for assessing the building performance. They base their estimates on subjective observations from their years of working experience. However, this is consistent with the findings of de Wilde (2019), in which he hypothesized that one of the assessment options for measuring building performance is primarily experience-based. Nevertheless, as pointed out in section 2.7.2, Omar et al. (2017), Li et al. (2020) and Jain et al. (2020) argue that an assessment according to measurable methods such as KPIs and based on sensor data is much more accurate and easier to reproduce by other asset managers. At the moment, the assessment may vary between the asset managers because each person has different built-up experiences that he or she has achieved over the years and would therefore make the same assessment differently than their colleagues. Remarkably, all interviews show that all asset managers would like to formulate their decisions and assessments based on facts derived from sensor data. This is a significant contradiction on its own since this would rule out their current ways of working as a suitable way of assessing the building performance.

The interviews also showed that the "building performance" assessment for asset managers is mainly based on the satisfaction of the end-user, and for the asset managers themselves, they view the occupancy rate as the most important indicator. However, in this study, no attention was paid to setting up a survey on measuring how satisfied users are with their experience in the building, as this was beyond the scope of this study. This makes the assessment of building performance incomplete, so the results are mainly based on one's own insights and can therefore be questioned. On the other hand, the results show that the asset manager can assess the building's performance with the help of the dashboard and thereby is able to formulate a critical decision. Thus, the main objective of formulating a decision based on the sensor data has been achieved.

7.2 Limitations of the Research

It should be taken into account that this research has focused exclusively on internal climate factors that influence the comfort experience of the asset users and the occupancy rate. If the focus would be on economic, ecological or energy performance-related factors, the results would possibly be different. The dashboard uses sensors exclusively and places particular emphasis on them. It does not say anything about the materials or equipment used in the building or to what extent cost reduction is achieved. For this reason, no general statement can be made about the building performance by organizations that do not focus on comfort perception and occupancy rates. Moreover, this research is mainly concerned with utility buildings, thus the building performance indicators chosen, and overall use case are only focused on these types of buildings. If it would have been a different kind of building, the use case or indicators chosen might have been different. Therefore, no statement can be made as to whether the use case and measured values would show the same results if the building type were different.

For practical reasons, this thesis cannot provide a complete overview of the opinions and interests held by all asset managers. In fact, five interviews were conducted. Thus, the sample size is on the small side to consider all opinions. Whereas this is precisely where an enormous added value lies because it was from these discussions in particular that the user requirements emerged, and more insights were gained into the potential of linked data for asset management. However, it does meet the minimum requirement for the number of interviews conducted for a qualitative study. Moreover, the sample is mainly focused on Dutch universities, so it is not possible to conclude with certainty whether the user requirements in the Netherlands also apply to asset managers in other countries.

Besides limitations in the field of conducting qualitative research, there are also limitations in the context of the supplied data samples. Firstly, with the sensor data supplied, it was assumed that an API connection would be established at the start of this research, as this would also increase future usability. Instead, historical sensor data was delivered from the Atlas living lab server in an Excel format that only applies to the 8th floor. The problem with these files was the way in which the sensor data was structured within the columns and spread out over the spreadsheets. There is nothing wrong with this file format for a typical Excel user, but for further data integration and automation, it is less than ideal. This is because the dashboard cannot read an Excel file to build the chart. Given the large amount of data to be converted, the final data used in the dashboard has been drastically reduced to three sensor types and nine sensors. As a result, the total scope of the research was reduced from two floors to just three rooms. The effect of the observed values on the total floor could therefore not be calculated.

There were also problems with the supplied glTF file. No spaces were provided in that file, which made it impossible to execute a SPARQL query on the spaces and check whether the sensors were present. As a result, the glTF file had to be generated again with all the spaces attached, but the conversion had problems with parsing the guids instead of the IFC names. This left the problem unsolved as the dashboard uses guids for all kinds of functions and queries. In response to this, the decision was made to insert the guids manually into the glTF files. However, this only applied the spaces 8.128, 8.127 and 8.126 for which the analyses were made. The dashboard continued to work, so it can be assumed with certainty that if the guids are parsed correctly, no problems will arise. However, it is essential that the parsing is conducted correctly. This requires several different software installations. With all these dependencies, the problem is that it makes replicating the outcomes in this thesis difficult for other researchers.

The main limitation of this research are the hardcoded codes found in the front end. As is the case with the various software packages required to create the glTF files, these hardcoded lines of code create dependencies. This makes it less than ideal for another researcher to use the dashboard directly, as he or she must first find and rewrite these codes to suit their own situation. First, there is the semantic enrichment of the sensor data and feeding it back into the TTL file. The TTL file is normally generated using the IFCtoLBD converter written by Oraskari (2020)[52] or Pauwels (2017)[53] found both on GitHub. However, this automation requires that the sensors are also included in the Revit model and especially in the IFC file. In this research, the sensors were not modulated, and it was decided to first describe the sensors in Excel and then paste them directly into the TTL file, although pieces of Excel code were written to speed up this process. Nevertheless, this should be an automated process, as hardcoded codes are prone to human error and not very scalable.

In addition, parts of the code around the creation of the graphs were also hardcoded. This is partly because the source files were not suitable. Therefore, it was decided to create a separate JSON format that acts as the data file for the chart.js graphs. The problem with this is that the source data file cannot be scaled up with more sensors data, which means it is too limited to the three spaces. It would have been better for future scalability to get the delivered data directly into a proper format or to receive it via an API and to use a written function to transform the converted JSON format into the desired format.

Finally, there are limitations in terms of query possibilities. The queries written within this software application are bound to parameters that must be fulfilled to execute a query. This is partly due to the fact that the application is bound to the initial state context file in which all states are fetched and pushed. As a result, some states are predefined which are not dynamic enough to receive any other query parameters. For the dashboard, there are two query parameters with a specific function. First, the ?guid parameter helps the dashboard highlight the building elements. Finally, there is the ?p parameter which aims to retrieve the properties from the TTL file and present them in a query table. No rooms can be highlighted without one of these two parameters, and no query tables with sensors information can be retrieved. In conclusion, the limitations can be narrowed down to two main aspects: flexibility and dataset constraints.

[52] GitHub - jyrkioraskari/IFCtoLBD: IFCtoLBD converts IFC (Industry Foundation Classes STEP formatted files into the Linked Building Data ontologies.
[53] https://github.com/pipauwel/IFCtoRDF/releases

8 Conclusion

8.1 Answering the main research question

This study focuses on the creation and evaluation of a web application that integrates BMS sensor data with BIM data, using semantic web technologies in order to create a digital twin. The main goal is to support asset managers in their decision making. In addition, this study aims to evaluate the benefits of this approach for asset managers. To achieve these goals, a case study was set up within the built environment, trying to find out if it is possible to link BIM and BMS data, followed by an analysis of how the web application can retrieve this data, what functionalities this system needs, and should support so that this data can not only be stored, but can also be displayed correctly. The intended purpose of this research was to answer the following research question:

- **" To what extent can BIM data be used to realise a decision support system that helps asset managers to optimise the building performance of utility buildings during the exploitation phase?"**

To answer this main research question, this research consisted of five parts divided into a literature review, establishing the methodology and system framework, developing a proof of concept and evaluating the Atlas case study. The main objective of the literature study was to dissect the different concepts within this research and find a link between asset management, building data, BIM concepts and semantic web technologies. From this literature study, it was possible to determine that asset management can be defined as a joint effort of an organisation to realise/recover value from an asset. Whereby value realisation occurs when a balance is sought between costs, risks, opportunities and increasing the building performance. The term value refers, with regard to this research, to a building that fulfils its functional value. This functional value of a building depends on the organisation that manages the building but can be subdivided into various categories such as economic value, social value, cultural value, operational value and symbolic value.

Central to this study is the realisation of the operational value. An asset manager can try to realise all kinds of different categories of value with his building. However, for utility buildings, the operational value in relation to the end-user is the main focus. In other words, it means ensuring that the building is operational so that the end user remains satisfied with the functioning of the building and also feels comfortable. The asset manager's objective is to strive for this value realisation through all kinds of activities he performs. In addition, the asset manager must guarantee this operational functionality by steering on the building performance during the operation and maintenance phase.

Among these activities are the use of asset information systems. According to both the literature and the interviews, the use of BIM by asset managers during the O&M phase to monitor and control the sensor data of the building has been ruled out for the time being. Because this technique is hardly used during the O&M phase. Instead, many organisations use individual asset information systems such as CMMS, BMS, BAS, or EAMS to carry out individual task. Although these systems can be excellent for monitoring individual building components such as the HVAC, lighting, and security systems, there are several limitations to these systems. Each system comes from a vendor with its own data structure, which hinders integration between the different systems often found in buildings. In addition, they often require manual human input. This leads to human error margins. Furthermore, they do not have 3D interactive models that can be used to address specific sensors and/or defects. The concept of BIM solves the above problems to some extent. The use of a BIM methodology not only leads to higher use of IT in construction but also increases the coordination of design activities, integration of different systems and the transfer of building information. BIM makes it possible to significantly reduce the time spent on finding semantic O&M data together with the geographical data for the material and building components. However, BIM is not equally suited to the monitoring and measurement of building performance aspects. The concept of building performance and BMS are closely related to each other, as BMS is primarily aimed at monitoring the building's internal climate and energy consumption. A combination of BIM and BMS sensor data is, therefore, necessary to solve the aforementioned problems. The sensor data in BMS mainly represent the output values related to the building performance. The term building performance in relation to asset management is the concept of targeting the improvement of the building's internal climate and energy use during different building phases. This includes controlling the CO2, humidity, temperature, airspeed, and energy consumption indicators. The literature study shows that a distinction can be made between target values in warmer and colder seasons. The values to be aimed for can fluctuate between 1180 and 1300 PPM for the CO2 values, between 0.1 and 0.3 m/s for the airspeed. The temperature should be between 22.8 and 24.2 °C, and the air humidity between 37.3% for cold seasons and 50.9% for warm seasons. Too high or too low below these values and the comfort of the occupants may be perceived as uncomfortable. As a result, it can have a negative impact on overall user satisfaction. Both the literature and interviews show that BIM information models have the potential to allow an asset manager to monitor the building throughout its lifecycle on sensor data. However, it also shows that these developments are as of yet barely present.

An interesting approach from the literature is that of Valinejadshoubi, Moselhi, Bagchi & Salem (2020), in which the physical sensors are coupled to virtual sensors in the BIM model by using their sensor IDs. They suggested that the timestamp and measurement units are important for creating a digital twin in addition to the sensor IDs. To this end, they used a MySQL database and Dynamo to establish the connection and thus create a digital twin. In contrast to these authors, for this research, the building data is linked to the sensor data using semantic web technologies. By using linked data and the BOT and PROPS ontologies, it is possible to provide the sensor data and the BIM model with semantic rich data. Using the BOT ontology, spaces can be defined, and the sensors can be placed in the spaces using the bot:containsElement attribute. In combination with the PROPS ontology that defines and inserts the properties of the sensors, a link can be formed between the BMS sensor data and the BIM model. This connection is achieved by using the bot:containsElement and bot:hasGuid attributes. In addition to the bot:containsElement, which is used to indicate which sensors can be found in each space, the bot:hasGuid can be used to establish a link between the same guid found in the BIM model and the guid found at bot:hasGuid in the semantic rich turtle file.

To support this methodology and to build the decision support system, a system architecture was developed during this research with a React.js front-end and MongoDB & GraphDB back-end. This system architecture is built from the use case and user requirements that followed from the interviews with the asset managers. A total of 5 interviews were conducted with senior asset managers from different Dutch universities. From these interviews, the scope for the building performance aspects was further narrowed. Instead of energy and all indoor environment indicators, the scope was set to the temperature and occupancy of the building. This is because the interviews showed that the occupancy rate is the most important indicator for the asset managers, and for the end-user, it is the comfort that they perceive in the building. Starting from these requirements and use case, it was decided to achieve the following as a result: The asset manager is able to display the triple store visualization of the BIM model in a web UI and is able to access data from the Atlas living lab within a graph. Hereby the asset manager must be able to upload the BIM model, and connect to the sensor data without being bothered with converting and processing individual files. The proposed framework and use case can be assessed for strengths and weaknesses. The weaknesses are covered in the section limitations. From the overall study, it can be concluded that the dashboard contributes to better asset management since the asset manager is able to view O&M sensor data faster, more efficiently and more interactively based on the BIM model in the viewer. The dashboard is not only able to display the data of the building, but also to display the individual sensor data for the individual rooms where these sensors are located. This allows the asset manager to steer more effectively on the basis of this information. With the occupancy rate per room, the asset manager is able to monitor not only which rooms are being used, but especially how efficiently and intensively the building is being used. This makes it possible to start the discussion with the asset owner and end-users about the number of square metres that is being used, so that more can be done about the functional use and thus the operational functioning of the building. In addition, the asset manager is also able to monitor the internal climate per room and to display this in graphs. The interviews show that the translation of the building to the end-user is not always optimal. However, user satisfaction is a key objective of the asset manager. The interviews also revealed an interest in showing the charts to end users. Based on the graphs for the temperature, the usage of the building can be shown to the end-user so that they can understand the building better and are therefore able to be satisfied with the building. It can therefore be concluded that BIM data and BMS sensor data are, with the help of semantic web technologies, to a certain extent capable of building a decision support system for the asset manager that allows the asset manager to steer on the building performance.

8.2 Contribution to current literature and societal relevance

The current study is an addition to the existing literature on linked building data and asset management. As previous studies have not provided conclusions on how asset managers can make critical decisions around their assets using a combination of semantic web technologies, asset sensor data and BIM. Most studies (Dibley et al. 2012; Hu et al. 2018; Chevallier et al. 2020; Quinn et al. 2020) focus only on integrating BMS data and the BIM model or on writing ontologies (Kim et al. 2018; Mohamed, Abdallah & Marzouk 2020). Based on this research, asset managers should focus on further integrating sensors by using linked building data principles and thus aim for creating digital twins, in order to make much more effective, efficient and factual choices around their policy considerations about their assets. What asset managers should avoid here is an approach using complex integration methods such as using Dynamo to integrate the sensor data with the BIM model. However, what this study mainly shows is that an integration directly from a living lab is preferred, especially when an API connection is available and can be used to integrate the sensor data directly into a MongoDB or other SQL backend. If this is not the case, then a integration using linked data and ontologies is preferred. It is important to use the sensor ID to represent both the physical and virtual sensor and to link all data, such as the sensor data to this sensor ID. The interviews show that linked data has a lot of potential for asset management because it offers asset managers the possibility to connect their asset information systems with other semantic databases. This enables them to establish connections that were previously not possible and to extract value from these new information sources. It is particularly interesting for utility building owners to see whether they can integrate their school scheduling system with the BIM model and their asset information system using a linked data approach. The societal added value of this study comes from the fact that the asset manager is better able to see, on the basis of this dashboard, what the exact occupancy rate is and what the indoor environmental quality is. This enables him to discuss the actual occupancy of the building with the end-users and asset owners. As a result, the functional layout of the building can be handled more efficiently, or it can be downsized because the number of square metres is too large in relation to the current usage. As a result, fewer new buildings will be constructed in the future, or the number of larger buildings will be reduced, thus reducing the ecological footprint.

8.3 Recommendations

The following section describes the recommendations that follow from the limitations in section 7.2 in this study.

User Satisfaction

The user satisfaction is not included in the assessment of the building performance, although this is a central objective of the asset manager. It would be of great value to further develop the decision support framework by setting up a survey for the end-user or physically setting up a tool in the rooms to monitor user satisfaction and link it to the dashboard. In this way, the sensor data, the BIM model and the actual satisfaction of the user can be linked to each other.

API Integration

In this research, it was impossible to set up an API connection with the BIM viewer, so the sensor data is not updated, and no live connection could be established. For a follow-up study, it is recommended to connect the sensor data to a back-end framework, preferably MongoDB and connect it live to the front-end using APIs. This would also remove the limitations around the hardcoded sensor data in the JSON file and resolve some of the dependencies.

Conversion Tools

In this study, there were problems with the conversions of the IFC to glTF and TTL files. For a follow-up study, it is recommended to develop a conversion tool that is linked to the dashboard. This conversion tool must be integrated into the dashboard as a button. This allows a user to upload the IFC file, which a back-end converts to a glTF and LBD file using IFCtoCOLLADA, COLLADA2glTF and IFCtoLBD. The IFCtoLBD tool does not yet format the TTL file in an orderly fashion, which can make it confusing for a developer to work with.

Extend colour coding BIM/sensor viewer

The BIM model is able to highlight certain building elements when a SPARQL query is executed. In order to further promote the digital twin integration, it is recommended for a follow-up study to visualize the current occupancy rate, energy consumption, temperature or other sensor values in the BIM viewer with a colour coding scheme. This makes it possible to monitor the building performance in the BIM viewer more accurately and without SPARQL queries.

9 Reference list

Abu Bakar, N., Hassan, M. Y., Abdullah, H., Abdul Rahman, H., Abdullah, M. P., Hussin, F., & Bandi, M. (2015). Energy efficiency index as an indicator for measuring building energy performance: A review. *Renewable and Sustainable Energy Reviews*, *44*, 1–11. https://doi.org/10.1016/j.rser.2014.12.018

Akanmu, W. P., Nunayon, S. S., & Eboson, U. C. (2021). Indoor environmental quality (IEQ) assessment of Nigerian university libraries: A pilot study. *Energy and Built Environment*, *2*(3), 302–314. https://doi.org/10.1016/j.enbenv.2020.07.004

Al-Kasasbeh, M., Abudayyeh, O., & Liu, H. (2021). An integrated decision support system for building asset management based on BIM and Work Breakdown Structure. *Journal of Building Engineering*, *34*, 101959. https://doi.org/10.1016/j.jobe.2020.101959

Al Dakheel, J., Del Pero, C., Aste, N., & Leonforte, F. (2020). Smart buildings features and key performance indicators: A review. In *Sustainable Cities and Society* (Vol. 61, p. 102328). Elsevier Ltd. https://doi.org/10.1016/j.scs.2020.102328

Angela de Barros Lima. (2020). *A web-based application to integrate building management system sensor data and building information model data to support facility management tasks*.

ASHRAE. (2017). ANSI/ASHRAE Standard 55-2017 : Thermal Environmental Conditions for Human Occupancy. *ASHRAE Inc.*, *2017*, 66.

Azhar, S., Nadeem, A., Mok, johnny, & Leung, B. (2008). *Building Information Modeling (BIM): A New Paradigm for Visual Interactive Modeling and Simulation for Construction Projects*.

Aziz, N. D., Nawawi, A. H., & Ariff, N. R. M. (2016). ICT Evolution in Facilities Management (FM): Building Information Modelling (BIM) as the Latest Technology. *Procedia - Social and Behavioral Sciences*, *234*, 363–371. https://doi.org/10.1016/j.sbspro.2016.10.253

Barthelmes, V., Fabi, V., Corgnati, S., & Serra, V. (2019). Human Factor and Energy Efficiency in Buildings: Motivating End-Users Behavioural Change. In S. Bagnara, R. Tartaglia, S. Albolino, T. Alexander, & Y. Fujita (Eds.), *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)* (pp. 514–525). Springer International Publishing.

Bassier, M., Bonduel, M., Derdaele, J., & Vergauwen, M. (2020). Processing existing building geometry for reuse as Linked Data. *Automation in Construction*, *115*, 103180. https://doi.org/10.1016/j.autcon.2020.103180

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data - The story so far. *International Journal on Semantic Web and Information Systems*, *5*(3), 1–22. https://doi.org/10.4018/jswis.2009081901

Boje, C., Guerriero, A., Kubicki, S., & Rezgui, Y. (2020). Towards a semantic Construction Digital Twin: Directions for future research. In *Automation in Construction* (Vol. 114, p. 103179). Elsevier B.V. https://doi.org/10.1016/j.autcon.2020.103179

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., & Klein, R. (2018). The IFC to linked building data converter - Current status. *CEUR Workshop Proceedings*, *2159*, 34–43.

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building Information Modeling: Why? What? How?: Technology Foundations and Industry Practice. In *Building Information Modeling: Technology Foundations and Industry Practice* (pp. 1–24). https://doi.org/10.1007/978-3-319-92862-3_1

Brous, P., Herder, P., & Janssen, M. (2016). Governing Asset Management Data Infrastructures. *Procedia Computer Science*, *95*. https://doi.org/10.1016/j.procs.2016.09.339

Brous, P., Herder, P., & Janssen, M. (2015). Towards Modelling Data Infrastructures in the Asset Management Domain. *Procedia Computer Science*, *61*, 274–280. https://doi.org/10.1016/j.procs.2015.09.215

Chevallier, Z., Finance, B., & Boulakia, B. C. (2020). A reference architecture for smart building digital twin. *CEUR*

Workshop Proceedings, *2615*, 1–12.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., & O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, *27*(2), 206–219. https://doi.org/10.1016/j.aei.2012.10.003

Dave, B., Buda, A., Nurminen, A., & Främling, K. (2018). A framework for integrating BIM and IoT through open standards. *Automation in Construction*, *95*(August), 35–45. https://doi.org/10.1016/j.autcon.2018.07.022

Davis, P. R. (2015). Monitoring and control of thermal energy storage systems. In *Advances in Thermal Energy Storage Systems: Methods and Applications* (pp. 419–440). Elsevier Inc. https://doi.org/10.1533/9781782420965.4.419

de Wilde, P. (2019). Ten questions concerning building performance analysis. *Building and Environment*, *153*, 110–117. https://doi.org/10.1016/j.buildenv.2019.02.019

Demanega, I., Mujan, I., Singer, B. C., Anđelković, A. S., Babich, F., & Licina, D. (2021). Performance assessment of low-cost environmental monitors and single sensors under variable indoor air quality and thermal conditions. *Building and Environment*, *187*, 107415. https://doi.org/10.1016/j.buildenv.2020.107415

Dibley, M. J. (2011). An intelligent system for facility management. *Computers & Industrial Engineering*, *38*(3), 397–412. http://orca.cf.ac.uk/23277/

Dibley, M., Li, H., Rezgui, Y., & Miles, J. (2012). An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, *28*, 1–14. https://doi.org/10.1016/j.autcon.2012.05.018

Ding, L. Y., Zhong, B. T., Wu, S., & Luo, H. B. (2016). Construction risk knowledge management in BIM using ontology and semantic web technology. *Safety Science*, *87*, 202–213. https://doi.org/10.1016/j.ssci.2016.04.008

Eastman, C., Teicholz, P., Sacks, R. and Liston, K. (2011). Animated), BIM Handbook: A Guide to Building Information Modelling for Owners, Managers, Designers, Engineers, and Contractorshorsehairs. In *Notes and Queries* (Vols. s7-II, Issue 32). https://doi.org/10.1093/nq/s7-II.32.110-e

Elhakeem, A., & Hegazy, T. (2012). Building asset management with deficiency tracking and integrated life cycle optimisation. *Structure and Infrastructure Engineering*, *8*, 729–738. https://doi.org/10.1080/15732471003777071

Farghaly, K., Abanda, F. H., Vidalakis, C., & Wood, G. (2018). Taxonomy for BIM and Asset Management Semantic Interoperability. *Journal of Management in Engineering*, *34*(4), 04018012. https://doi.org/10.1061/(asce)me.1943-5479.0000610

Fatima, H., & Wasnik, K. (2016). Comparison of SQL, NoSQL and NewSQL databases for internet of things. *2016 IEEE Bombay Section Symposium (IBSS)*, 1–6. https://doi.org/10.1109/IBSS.2016.7940198

Geraldi, M. S., & Ghisi, E. (2020). Building-level and stock-level in contrast: A literature review of the energy performance of buildings during the operational stage. *Energy and Buildings*, *211*, 109810. https://doi.org/https://doi.org/10.1016/j.enbuild.2020.109810

Gouda Mohamed, A., Abdallah, M. R., & Marzouk, M. (2020). BIM and semantic web-based maintenance information for existing buildings. *Automation in Construction*, *116*. https://doi.org/10.1016/j.autcon.2020.103209

Guillen, A. J., Crespo, A., Gómez, J., González-Prida, V., Kobbacy, K., & Shariff, S. (2016). Building Information Modeling as Assest Management Tool. *IFAC-PapersOnLine*, *49*(28), 191–196. https://doi.org/10.1016/j.ifacol.2016.11.033

Heilman, C., & Francis, N. (2007). Web development solutions. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Holland, C. P., Shaw, D. R., & Kawalek, P. (2005). BP's multi-enterprise asset management system. *Information and Software Technology*, *47*(15). https://doi.org/10.1016/j.infsof.2005.09.006

Homaei, S., & Hamdy, M. (2020). A robustness-based decision making approach for multi-target high performance buildings under uncertain scenarios. *Applied Energy*, *267*, 114868. https://doi.org/https://doi.org/10.1016/j.apenergy.2020.114868

Howell, S., Rezgui, Y., & Beach, T. (2018). Water utility decision support through the semantic web of things. *Environmental Modelling and Software*, *102*, 94–114. https://doi.org/10.1016/j.envsoft.2018.01.006

Hu, S., Corry, E., Horrigan, M., Hoare, C., Dos Reis, M., & O'Donnell, J. (2018). Building performance evaluation using OpenMath and Linked Data. *Energy and Buildings*, *174*, 484–494. https://doi.org/10.1016/j.enbuild.2018.07.007

Hu, Z. Z., Tian, P. L., Li, S. W., & Zhang, J. P. (2018). BIM-based integrated delivery technologies for intelligent MEP management in the operation and maintenance phase. *Advances in Engineering Software*, *115*, 1–16. https://doi.org/10.1016/j.advengsoft.2017.08.007

Iddianozie, C., & Palmes, P. (2020). Towards smart sustainable cities: Addressing semantic heterogeneity in Building Management Systems using discriminative models. *Sustainable Cities and Society*, *62*(April), 102367. https://doi.org/10.1016/j.scs.2020.102367

Jain, N., Burman, E., Stamp, S., Mumovic, D., & Davies, M. (2020). Cross-sectoral assessment of the performance gap using calibrated building energy performance simulation. *Energy and Buildings*, *224*. https://doi.org/10.1016/j.enbuild.2020.110271

Jose, B., & Abraham, S. (2020). Performance analysis of NoSQL and relational databases with MongoDB and MySQL. *Materials Today: Proceedings*, *24*, 2036–2043. https://doi.org/https://doi.org/10.1016/j.matpr.2020.03.634

Joseph, J. (2018). Facility Design and Process Utilities. In *Biopharmaceutical Processing: Development, Design, and Implementation of Manufacturing Processes* (pp. 933–986). Elsevier. https://doi.org/10.1016/B978-0-08-100623-8.00045-1

Kallio, J., Vildjiounaite, E., Koivusaari, J., Räsänen, P., Similä, H., Kyllönen, V., Muuraiskangas, S., Ronkainen, J., Rehu, J., & Vehmas, K. (2020). Assessment of perceived indoor environmental quality, stress and productivity based on environmental sensor data and personality categorization. *Building and Environment*, *175*, 106787. https://doi.org/https://doi.org/10.1016/j.buildenv.2020.106787

Kepner, J., Gadepally, V., Hutchison, D., Jananthan, H., Mattson, T., Samsi, S., & Reuther, A. (2016). Associative array model of SQL, NoSQL, and NewSQL databases. *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–9. https://doi.org/10.1109/HPEC.2016.7761647

Kim, K., Kim, H., Kim, W., Kim, C., Kim, J., & Yu, J. (2018). Integration of ifc objects and facility management work information using Semantic Web. *Automation in Construction*, *87*, 173–187. https://doi.org/10.1016/j.autcon.2017.12.019

Kim, Y. M., Ahronheim, J., Suzuka, K., King, L. E., Bruell, D., Miller, R., & Johnson, L. (2007). Enterprise digital asset management system pilot: Lessons learned. *Information Technology and Libraries*, *26*(4). https://doi.org/10.6017/ital.v26i4.3266

Korsavi, S. S., Montazami, A., & Mumovic, D. (2020). The impact of indoor environment quality (IEQ) on school children's overall comfort in the UK; a regression approach. *Building and Environment*, *185*, 107309. https://doi.org/https://doi.org/10.1016/j.buildenv.2020.107309

Kučera, A., & Pitner, T. (2018). Semantic BMS: Allowing usage of building automation data in facility benchmarking. *Advanced Engineering Informatics*, *35*(February), 69–84. https://doi.org/10.1016/j.aei.2018.01.002

Kučera, A., & Pitner, T. (2016). *Semantic BMS: Ontology for Analysis of Building Automation Systems Data BT - Technological Innovation for Cyber-Physical Systems* (L. M. Camarinha-Matos, A. J. Falcão, N. Vafaei, & S. Najdi (eds.); pp. 46–53). Springer International Publishing.

Kuck, G. (2004). Tim Berners-Lee's Semantic Web. *South African Journal of Information Management*, *6*. https://doi.org/10.4102/sajim.v6i1.297

Lee, Y. C., Solihin, W., & Eastman, C. M. (2019). The Mechanism and Challenges of Validating a Building Information

Model regarding data exchange standards. *Automation in Construction*, *100*, 118–128. https://doi.org/10.1016/j.autcon.2018.12.025

Li, G., Yan, L., & Ma, Z. (2019). An approach for approximate subgraph matching in fuzzy RDF graph. *Fuzzy Sets and Systems*, *376*, 106–126. https://doi.org/10.1016/j.fss.2019.02.021

Li, H., Hong, T., Lee, S. H., & Sofos, M. (2020). System-level key performance indicators for building performance evaluation. *Energy and Buildings*, *209*. https://doi.org/10.1016/j.enbuild.2019.109703

Li, Y., García-Castro, R., Mihindukulasooriya, N., O'Donnell, J., & Vega-Sánchez, S. (2019). Enhancing energy management at district and building levels via an EM-KPI ontology. *Automation in Construction*, *99*, 152–167. https://doi.org/10.1016/j.autcon.2018.12.010

Lima, E. S., McMahon, P., & Costa, A. P. C. S. (2021). Establishing the relationship between asset management and business performance. *International Journal of Production Economics*, *232*. https://doi.org/10.1016/j.ijpe.2020.107937

Lin, Shien & Gao, Jing & Koronios, A. (n.d.). *A Data Quality Framework for Engineering Asset Management*. https://doi.org/10.1007/978-1-84628-814-2_51.

Lin, S., Gao, J., & Koronios, A. (2006). Validating a data quality framework in engineering asset management. *ACIS 2006 Proceedings - 17th Australasian Conference on Information Systems*.

Love, P. E. D., Zhou, J., Matthews, J., Sing, C. P., & Carey, B. (2015). A systems information model for managing electrical, control, and instrumentation assets. *Built Environment Project and Asset Management*, *5*(3), 278–289. https://doi.org/10.1108/BEPAM-03-2014-0019

Lu, Q., Xie, X., Parlikad, A. K., & Schooling, J. M. (2020). Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. *Automation in Construction*, *118*, 103277. https://doi.org/10.1016/j.autcon.2020.103277

Maes, K., Reynders, E., Rezayat, A., Roeck, G. De, & Lombaert, G. (2016). Offline synchronization of data acquisition systems using system identification. *Journal of Sound and Vibration*, *381*, 264–272. https://doi.org/https://doi.org/10.1016/j.jsv.2016.06.015

Makonin, S. (2016). App programming and its use in smart buildings. In *Start-Up Creation: The Smart Eco-Efficient Built Environment* (pp. 451–463). Elsevier Inc. https://doi.org/10.1016/B978-0-08-100546-0.00018-2

Malcolm, A., Werbrouck, J., Pauwels, P., Eloy, S., Leite Viana, D., & Morais, F. (2020). LBD server: Visualising Building Graphs in web-based environments using semantic graphs and glTF-models LK - https://tue.on.worldcat.org/oclc/9062833780. In *TA - TT* -. Springer SE -.

Management, G. F. on M. and A. (2014). The Asset Management Landscape Second Edition The Global Forum on Maintenance and Asset Management. *Global Forum on Maintenance and Asset Management*, *2*(March).

Maslesa, E., Jensen, P. A., & Birkved, M. (2018). Indicators for quantifying environmental building performance: A systematic literature review. *Journal of Building Engineering*, *19*, 552–560. https://doi.org/https://doi.org/10.1016/j.jobe.2018.06.006

Migilinskas, D., Popov, V., Juocevicius, V., & Ustinovichius, L. (2013). The benefits, obstacles and problems of practical bim implementation. *Procedia Engineering*, *57*, 767–774. https://doi.org/10.1016/j.proeng.2013.04.097

Munir, K., & Sheraz Anjum, M. (2018). The use of ontologies for effective knowledge modelling and information retrieval. In *Applied Computing and Informatics* (Vol. 14, Issue 2, pp. 116–126). Elsevier B.V. https://doi.org/10.1016/j.aci.2017.07.003

Munir, M., Kiviniemi, A., Finnegan, S., & Jones, S. W. (2019). BIM business value for asset owners through effective asset information management. *Facilities*, *38*(3–4), 181–200. https://doi.org/10.1108/F-03-2019-0036

Natephra, W., & Motamedi, A. (2019a). *Live data visualization of IoT sensors using Augmented Reality (AR) and BIM*. https://doi.org/10.22260/ISARC2019/0084

Natephra, W., & Motamedi, A. L. I. (2019b). Virtual Reality for Monitoring Indoor Conditions. *24th Annual Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2019)*, *365*, 1–10.

Neuhaus, H., & Compton, M. (2009). The Semantic Sensor Network Ontology: A Generic Language to DescribeSensor Assets. *AGILE Workshop on Challenges in Geospatial Data Harmonisation*.

Nimlyat, P. S., & Kandar, M. Z. (2015). Appraisal of indoor environmental quality (IEQ) in healthcare facilities: A literature review. *Sustainable Cities and Society*, *17*, 61–68. https://doi.org/https://doi.org/10.1016/j.scs.2015.04.002

O'Donnell, J., Corry, E., Hasan, S., Keane, M., & Curry, E. (2013). Building performance optimization using cross-domain scenario modeling, Linked data, And complex event processing. *Building and Environment*, *62*, 102–111. https://doi.org/10.1016/j.buildenv.2013.01.019

Omar, M. F., Ibrahim, F. A., & Omar, W. M. S. W. (2017). Key Performance Indicators for Maintenance Management Effectiveness of Public Hospital Building. *MATEC Web of Conferences*, *97*, 1–6. https://doi.org/10.1051/matecconf/20179701056

Pärn, E., Edwards, D., & Sing, M. C. P. (2017). The building information modelling trajectory in facilities management: A review. *Automation in Construction*, *75*, 45–55. https://doi.org/10.1016/j.autcon.2016.12.003

Peng, S., Su, G., Chen, J., & Du, P. (2017). *Design of an IoT-BIM-GIS Based Risk Management System for Hospital Basic Operation*. https://doi.org/10.1109/SOSE.2017.22

Phuoc, D., & Hauswirth, M. (2009). Linked Open Data in Sensor Data Mashups. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09)* (Vol. 522).

Preiser, W. F. E., & Nasar, J. L. (2008). Assessing Building Performance: Its Evolution From Post-Occupancy Evaluation. *International Journal of Architectural Research: ArchNet-IJAR*, *2*(1), 84–99. https://doi.org/10.26687/archnet-ijar.v2i1.179

Preiser, W., & Vischer, J. (2005). *Assessing building performance*.

Quinn, C., Shabestari, A. Z., Misic, T., Gilani, S., Litoiu, M., & McArthur, J. J. (2020). Building automation system - BIM integration using a linked data structure. *Automation in Construction*, *118*, 103257. https://doi.org/10.1016/j.autcon.2020.103257

Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., & Gómez-Pérez, A. (2015). Guidelines for Linked Data generation and publication: An example in building energy consumption. *Automation in Construction*, *57*, 178–187. https://doi.org/10.1016/j.autcon.2015.04.002

Re Cecconi, F., Dejaco, M. C., Moretti, N., Mannino, A., & Blanco Cadena, J. D. (2020). Digital asset management. In *Research for Development*. Springer International Publishing. https://doi.org/10.1007/978-3-030-33570-0_22

Rio, J., Ferreira, B., & Martins, J. (2013). Expansion of IFC model with structural sensors. *Informes de La Construcción*, *65*, 219–228. https://doi.org/10.3989/ic.2013.v65.i530

Riso. (2012). Overview, Principles and terminology. *International Organization for Standardization*.

Roberts, B. C. (2020). *Developing Organization Resilience through Asset Management to Respond to COVID-19*. *May*, 1–5.

Salman, A. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering*, *11*(3), 241–252. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127

Shen, W., Hao, Q., Mak, H., Neelamkavil, J., Xie, H., Dickinson, J., Thomas, R., Pardasani, A., & Xue, H. (2010). Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review. *Advanced Engineering Informatics*, *24*(2), 196–207. https://doi.org/10.1016/j.aei.2009.09.001

Sinopoli, J. (2010). Facility Management Systems. In *Smart Building Systems for Architects, Owners and Builders* (Vol. 101, Issue SUPPL., pp. 129–137). Elsevier. https://doi.org/10.1016/B978-1-85617-653-8.00012-0

Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering principles and methods. *Data and Knowledge Engineering*.

Tan, T. &. (2002). Measuring operational building asset performance - concepts and implementation. *CIB Publication*, *283*, 384–387.

Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., & Gao, X. (2020). BIM assisted Building Automation System information exchange using BACnet and IFC. *Automation in Construction*, *110*, 103049. https://doi.org/10.1016/j.autcon.2019.103049

Tanrivermis, H. (2020). Possible impacts of COVID-19 outbreak on real estate sector and possible changes to adopt: A situation analysis and general assessment on Turkish perspective. *Journal of Urban Management*, *9*. https://doi.org/10.1016/j.jum.2020.08.005

Tchouanguem Djuedja, J. F., Abanda, F. H., Kamsu-Foguem, B., Pauwels, P., Magniont, C., & Karray, M. H. (2021). An integrated Linked Building Data system: AEC industry case. *Advances in Engineering Software*, *152*. https://doi.org/10.1016/j.advengsoft.2020.102930

Terkaj, W., Schneider, G. F., & Pauwels, P. (2017). Reusing domain ontologies in linked building data: The case of building automation and control. *CEUR Workshop Proceedings*, *2050*.

The Institute of Asset Management. (2015). *Asset Management – an anatomy (v3)*. *December*, 1–84. www.theIAM.org/AMA

Valinejadshoubi, M., Moselhi, O., Bagchi, A., & Salem, A. (2021). Development of an IoT and BIM-based automated alert system for thermal comfort monitoring in buildings. *Sustainable Cities and Society*, *66*, 102602. https://doi.org/https://doi.org/10.1016/j.scs.2020.102602

von Rosing, M., Scheer, A.-W., von Scheel, H., von Rosing, G., & Coloma, D. (2015). The Business Process Management Handbook. In *The Business Process Management Handbook*.

Wang, Y., Wang, X., Wang, J., Yung, P., & Jun, G. (2013). Engagement of Facilities Management in Design Stage through BIM: Framework and a Case Study. *Advances in Civil Engineering*, *2013*. https://doi.org/10.1155/2013/189105

Wehbe, R., & Shahrour, I. (2019). Use of BIM and Smart Monitoring for buildings' Indoor Comfort Control. *MATEC Web of Conferences*, *295*, 02010. https://doi.org/10.1051/matecconf/201929502010

Werbrouck, J., Pauwels, P., Bonduel, M., Beetz, J., & Bekers, W. (2020). Scan-to-graph: Semantic enrichment of existing building geometry. *Automation in Construction*, *119*(June), 103286. https://doi.org/10.1016/j.autcon.2020.103286

Wienker, M., Henderson, K., & Volkerts, J. (2016). The Computerized Maintenance Management System an Essential Tool for World Class Maintenance. *Procedia Engineering*, *138*, 413–420. https://doi.org/10.1016/j.proeng.2016.02.100

Wu, I. C., & Liu, C. C. (2020). A visual and persuasive energy conservation system based on BIM and IoT technology. *Sensors (Switzerland)*, *20*(1). https://doi.org/10.3390/s20010139

Zhang, X., Arayici, Y., Wu, S., Abbott, C., & Aouad, G. (2009). Integrating BIM and GIS for large-scale facilities asset management: a critical review. *The Twelfth International Conference on Civil, Structural and Environmental Engineering Computing*. http://usir.salford.ac.uk/11418/2/Integrating_GIS_and_BIM_for_large_scale_asset_management.docx.pdf

Appendix A Asset management activities

Focus Area as defined by (IAM, 2015)	Asset management Process as defined by (GFMasset manager, 2014; Lima, McMahon & Costa, 2020)	Context as defined by (GFMasset manager, 2014)	Products as defined by (GFMasset manager, 2014; IAM, 2015; Lima, McMahon & Costa, 2020)
Strategy & Planning	1. asset management Policy 2. Demand Analysis 3. asset management Strategy & Objectives 4. Strategic Planning 5. asset management Planning	1. Set of principles that guide the development of an organizations asset management strategy and objectives 2. Analyzing future demand for the product and its requirements. 3. Strategic long-term approach to managing the physical assets 4. The process and organisation undertakes towards strategic asset management Planning 5. Activities taken to develop detailed asset management plans	• Demand forecast • Historical demand analysis • Demand scenarios • Demand management strategy • Service level specifications • asset management Policy • Criteria made towards lifecycle cost and risk analysis for optimizing asset interventions • asset management Strategy • asset management Objectives based upon scenario analysis • Strategic Asset management plan integrated with other organizational plans (finance, health & safety, HR) • Work volumes and costs analysis
Asset management Decision making	6. Capital investment Decision making 7. O&M Decision making 8. Resourcing Strategy 9. Lifecycle value realisation 10. Shutdowns & Outages Strategy	6. Approach towards evaluating alternative long-term beneficial investments 7. Managerial activities involved in determining necessary O&M activities towards the asset management Objectives. 8. An analysis that determines the best way to procure the required resources for achieving the asset management objectives 9. The method used that ensure the best total value from asset acquisition, creation, utilization, maintenance 10. Requirements that enable reduced downtime and outages while considering the cost to carry out an activity	• Consideration of all lifecycle costs, risks, information in a life cycle analysis (Life cycle costing algorithms) • Prioritising process for Capital Investments • Customer quality requirements • Asset capabilities (Product, service flexibility, quality) requirements • Determination of maintenance requirements activities • Maintenance standards and specifications • Material spares and equipment management strategy • Procurement plans for purchasing hard and software • Resource strategy and project plans • Determining methodologies for determining the value • Criteria set up for decision making • Lifecycle value analysis processes and application criteria • Shutdown & outage strategy, including operations, maintenance, engineering, project development, among others. • The trade-off analysis between efficient shutdown and outages • Long–term planned outages schedule • Shutdown and outage procedure requirements
Organisation & People	11. Procurement & Supply chain management 12. asset management Leadership 13. Organizational Structure 14. Organizational Culture 15. Competence Management	45 All asset management operations are matched with objectives in the method employed by organizations 46. Leadership of an organization required to promote a whole life asset management approach to deliver organizational asset management objectives 47. Capabilities of an organization to match its structure with its organizational and asset management objectives 48. Culture of an organization 49. maintaining an adequate supply of competent and motivated people to achieve asset management objectives	• Procurement policy, out/insourcing policy, contracts, service level specification, improvement plans • Leadership Management strategies/ Gap analysis/ continuity management plan/ accountability descriptions • Culture management strategy/ culture surveys / behavioural patterns analysis • Competence framework/assessment, training needs/analysis/specifications

Risk & Review	16. Risk Assessment & Management 17. Contingency planning & Resilience Analysis 18. Sustainable development 19. Management of Change 20. Assets Performance & Health monitoring 21. asset management System Monitoring 22. Management Review, Audit & Assurance 23. Asset Costing & Valuation 24. Stakeholder Engagement	50. Policies and processes for identifying and quantifying and mitigating risk and exploiting opportunities 51. Contingency plans for continuing operating assets in the event of an unforeseen happening. 52. Sustainable, balanced approach to economic activities, environmental responsibilities, and social progress. 53. identification, assessment, implementation and communicating changes to people, processes, and assets. 54. Assessment of the performance and health of the assets using indicators 55. The measurement used by an organization to assess the performance and health of its asset management systems 56. Review and auditing of the effectiveness of its asset management processes and asset management system 57. Setting up Asset costing structure/framework for capturing as-built maintenance and renewal unit costs and methods used for valuation of its assets 58. Scenario and methods that describe stakeholder engagement	• Generating/developing/executing Risk management policies • Aligning strategic, tactical, and operational risks towards risks registers, • Set up risk mitigation strategies. • Set up a contingency plan. • Environmental/social, and financial impact of asset management plans • Development Change management plans/policies/processes and execute them. • Define critical measures across all asset lifecycle stages, all linked towards objectives. • Establish monitoring programs for the evaluation of performance measure, analysis and the use of this information for management decision making and action plan. • Establish criteria in order to monitor deviations in the required asset performance. • Establish plans to monitor, measure and evaluate the asset across all stages of the life cycle. • Predict future asset performance and health. • Assessment of purpose of the asset management systems • Development of audit policies/measurement and execution • Asset valuation register, expenditure reports • Stakeholder policies, execution plans.
Life cycle delivery	25. Technical standards & Legislation 26. Asset Creating & Acquisition 27. Systems Engineering 28. Configuration management 29. Maintenance Delivery 30. Reliability Engineering 31. Asset Operations 32. Resource Management 33. Shutdown & Outage Management 34. Fault & Incident response 35. Asset Decommissioning & Disposal	59. process ensuring asset management are conducive towards technical standards and legislation 60. Acquisition, installation and commissioning of Assets 61. Interdisciplinary description of processes and requirements related to various analysis, design and evaluation of assets 62. Recording and monitoring assets functional, physical and support status. 63. Management of preventive and corrective maintenance activities 64. Process of ensuring reliability within components without failure of loss 65. Achieving business objectives through operating the assets 66. Implementing resourcing strategy in which funds, people, plant, tools, and materials are managed. 68. Plan development for responding on fault and incident	• Register of applicable Technical standards and legislation • Multiple Acquisition-related products such as; Strategy, Request, Agreement, agreement change request and communication reports • Programme management reports • Project management procedures • Project technical plan • Work breakdown structure • Verification report • Traceability mapping • Validation Report • Construction Progress Report • Acceptance Criteria Documents • Delivery Acceptance Report • Various system related documents such as engineering management plan, engineering performance measures, requirements, system analysis plan & reports, and description • Traceability mapping • Verification and validation strategies • Various configuration products such as; config management strategies, records, plans, status reports, system releases, baselines and agreements. • Maintenance staffing requirements, tools, strategy & tactics, and ICT infrastructure • Asset managers modelling output • RCM analysis output • Weibull plots and analysis • Completed root cause analysis
Asset information	36. Asset information strategy 37. Asset information Standards 38. Asset data Systems	70. Describes how asset information supports the delivery of asset management strategy and objectives. 71. specification for collecting and storing asset information 73. data and information held within the asset data systems	• Asset information policy, strategy, business cases, system business requirements • Asset information standards and guidelines, asset data dictionary, asset data quality definitions and guidelines • IT strategy, information systems architecture/strategy & business case /

	39. Data & information management		implementation & migration plan / governance • Data collection plans/ management procedure/ governance procedures/ assurance and auditing

Appendix B Building performance

This analysis was conducted by cross referencing various literature studies with each other. Each indicator found in these studies are cross referenced with one of the 83 indicators found by de Wilde (2019).

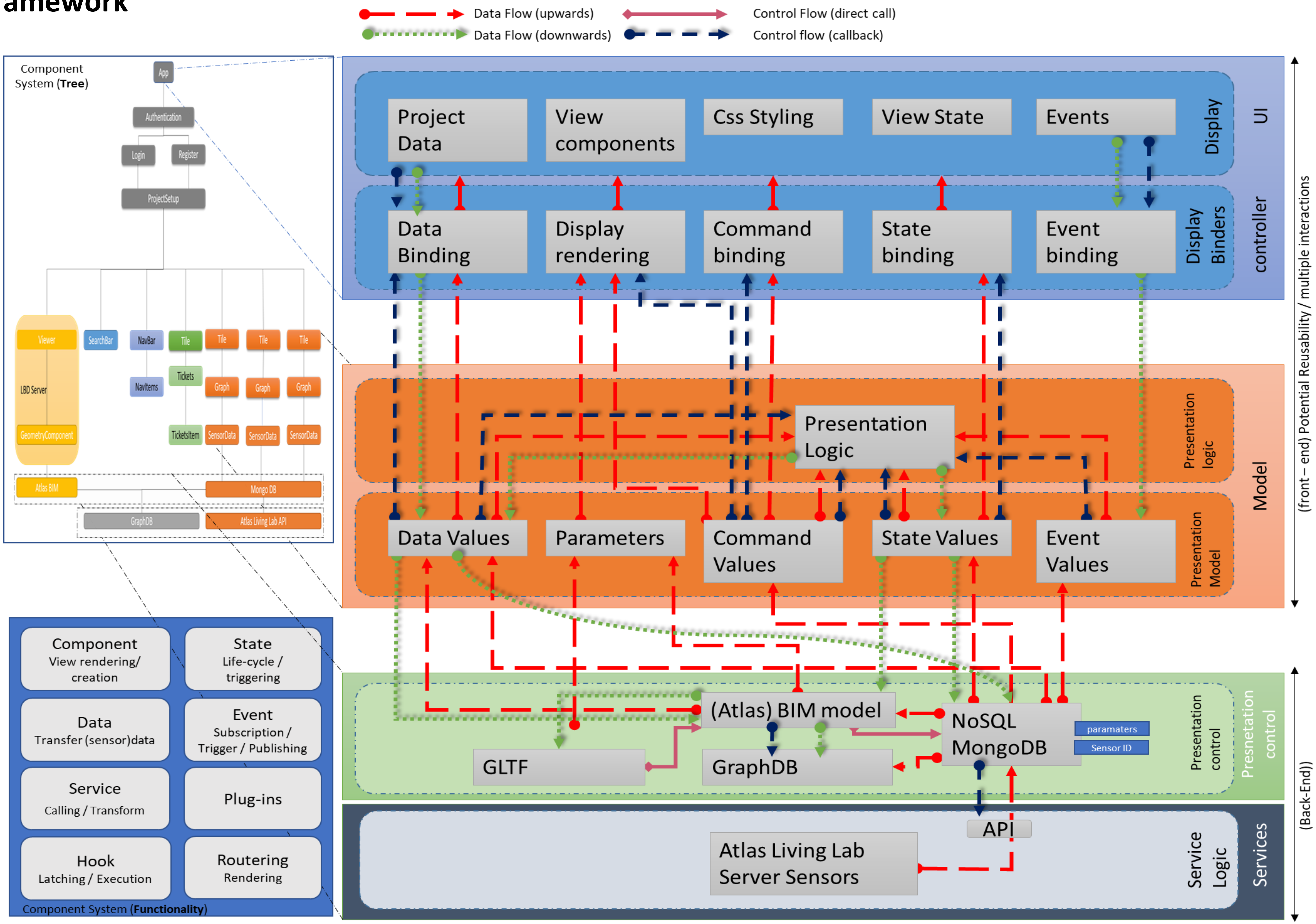
index	Aspects (as defined by bldg-perf.org)	Sub Aspects	Tu/e (2020)	Preiser & Nasar (2008)	Hong, Lee, Sofos (2020)	[Habibi (2017)]
1	Occupant satisfaction (provide)	x	x	x	x	x
2	Energy (make efficient use of)	x	√	x	√	√
3	Thermal comfort (maintain)	a. air temperature (control)	√	√	x	√
		b. radiant temperature (control)			x	
		c. air velocity (control)			x	
		d. relative humidity (control)			x	
		e. airspeed (control)			x	
		f. overheating (prevent)			x	
		g. undercooling (prevent)			x	
		h. wind chill (prevent)			x	
4	Acoustical comfort (maintain)	a. speech intelligibility (provide)		x	x	√
		b. reverberation time (control)	√	x	x	√
5	Visual comfort (maintain)	a. glare (prevent)	√	√	x	x
		b. flickering (prevent)	√	√	x	x
6	Olfactory comfort	a. odour (control)	√	x	x	x
7	Indoor air quality (maintain)	a. smoke, fumes, stale air (dispose of)	√	√	x	x
		b. fresh air (provide)	√	√	x	√
8	Structural integrity (maintain)	x	x	√	x	x
9	View to the outside	a. outside world (provide a connection with)	x	√	x	√
		b. Circadian rhythm (support)	x	x	x	x
10	Identity (provide)	x	x	x	x	x
11	Privacy (provide)	x	x	x	x	x
12	Inclusivity (support)	x	x	√	x	x
13	Relative humidity (control)	x	x	x	x	x
14	Vibration (protect from/limit)	x	x	x	x	x
15	Noise (protect from/limit)	x	x	x	x	√
16	Glare (protect from/limit)	x	x	x	x	x
17	Precipitation (keep out)	x	x	x	x	x

18	Ground/surface water (keep out)	x	**x**	**x**	**x**	**x**
19	Unwanted visitors/vermin (keep out)	x	**x**	**x**	**x**	**x**
20	Outdoor pollutants (keep out)	x	**x**	**x**	**x**	**x**
21	Electricity/gas/water (provide uninterrupted supply of)	x	**x**	**x**	**x**	**x**
22	Drainage/sewerage (provide safe and adequate)	x	**x**	**x**	**x**	**x**
23	Wayfinding (support)	x	**x**	**x**	**x**	**x**
24	Wind flow around building (control)	x	**x**	**x**	**x**	**x**
25	Condensation (prevent)	x	**x**	**x**	**x**	**x**
26	Contamination (prevent)	x	**x**	**x**	**x**	**x**
27	Complaints (minimize number of)	x	**x**	**x**	**x**	**x**
28	Fire ignition (prevent)	x	**x**	**x**	**x**	**x**
29	Fire spread (prevent)	x	**x**	**x**	**x**	**x**
30	Congestion, crowding (prevent)	x	**x**	**x**	**x**	**x**
31	Community (provide sense of)	x	**x**	**x**	**x**	**x**
32	Historical significance (have/respect)	x	**x**	**x**	**x**	**x**
33	Local and national heritage (contribute to)	x	**x**	**x**	**x**	**x**
34	Income/revenue (generate)	x	**x**	**x**	**x**	**x**
35	Key processes/work (enable)	x	**x**	**x**	**x**	**x**
36	Productivity (enable)	x	**x**	**x**	**x**	**x**
37	Ease of movement/circulation (provide)	x	**x**	√	**x**	**x**
38	Structural loading (carry)	a. dead load/own weight (resist)	**x**	**x**	**x**	**x**
		b. live load/occupants, furniture (resist)	**x**	**x**	**x**	**x**
		c. live load/wind, precipitation (resist)	**x**	**x**	**x**	**x**
		d. cycling loads/fatigue (resist)	**x**	**x**	**x**	**x**
39	Heating/cooling (provide)	x	√	√	√	√
40	Ventilation/fresh air (provide)	x	√	√	√	√
41	Daylight/sunlight (provide)	x	√	√	√	√

42	Hot and cold water (supply)	x	**x**	**x**	√	
43	Artificial lighting (provide)	x	**x**	**x**	**x**	√
44	ICT connectivity (provide)	x	**x**	**x**	**x**	√
45	Safety (ensure)	a. falling risk (mitigate)	**x**	**x**	**x**	**x**
		b. cutting risk (mitigate)	**x**	**x**	**x**	**x**
		c. risk from machines (mitigate)	**x**	**x**	**x**	**x**
		d. electrocution risk (mitigate)	**x**	**x**	**x**	**x**
46	(continuity of) services	x	**x**	**x**	**x**	**x**
47	Water (make efficient use of)	x	**x**	**x**	**x**	**x**
48	Material (make efficient use of)	x	**x**	**x**	**x**	**x**
49	Renewable energy (generate)	x	**x**	√	√	**x**
50	Rainwater (harvest)	x	**x**	**x**	**x**	**x**
51	Water (minimize use of)	x	**x**	**x**	**x**	**x**
52	Local ecosystem (protect)	x	**x**	**x**	**x**	**x**
53	Rare and endangered species (protect)	x	**x**	**x**	**x**	**x**
54	Wear and tear (resist)	x	**x**	**x**	**x**	**x**
55	Decay and rot (resist)	x	**x**	**x**	**x**	**x**
56	Corrosion (resist)	x	**x**	**x**	**x**	**x**
57	Construction costs (control)	x	**x**	**x**	**x**	**x**
58	Construction time (control)	x	**x**	**x**	**x**	**x**
59	Operational costs (control)	x	**x**	**x**	**x**	**x**
60	Cleanability (provide)	x	**x**	**x**	**x**	**x**
61	Maintenance and repair (efficiently provide)	x	**x**	**x**	**x**	**x**
62	Greenhouse gas emissions (limit)	x	**x**	**x**	**x**	**x**
63	Access control (provide)	x	**x**	**x**	**x**	**x**
64	HVAC control (provide)	x	**x**	√	√	**x**
65	Lighting control (provide)	x	**x**	√	√	√
66	Darkness (provide)	x	**x**	**x**	**x**	**x**
67	Solar radiation (control)	x	**x**	**x**	**x**	**x**
68	Urban context (respond to)	x	**x**	**x**	**x**	**x**
69	Site conditions (respond to)	x	**x**	**x**	**x**	**x**
70	Outside hours access (allow)	x	**x**	**x**	**x**	**x**
71	Modifications to building (allow)	x	**x**	**x**	**x**	**x**
72	Service life (manage)	x	**x**	**x**	**x**	**x**
73	Fire/smoke alarm (raise)	x	**x**	**x**	**x**	**x**
74	Intrusion alarm (raise)	x	**x**	**x**	**x**	**x**
75	Evacuation (allow)	a. evacuation route (provide)	**x**	**x**	**x**	**x**

		b. evacuation time (allow)	x	x	x	x
		c. survival time in refuge (guarantee)	x	x	x	x
76	Burglary (resist)	x	x	x	x	x
77	Vandalism (resist)	x	x	x	x	x
78	Extreme events (resist)	a. fire/smoke and heat (minimize impact of)	x	x	x	x
		b. explosion (minimize impact of)	x	x	x	x
		c. radioactivity spread (minimize impact of)	x	x	x	x
		d. poisonous substance spread (minimize impact of)	x	x	x	x
		e. heat wave (cope with)	x	x	x	x
		f. cold spell (cope with)	x	x	x	x
		g. natural disasters such as earthquakes (resist)	x	x	x	x
		h. human-made disasters such as terrorist attacks (resist)	x	x	x	x
79	Disease and infection (stop spreading of)	x	x	x	x	x
80	Buildability (provide)	x	x	x	x	x
81	Flexibility (posses)	x	x	x	x	x
82	Disposability (provide)	x	x	x	x	x
83	Aesthetics (consider)	a. architectural statement (make)	x	√	x	x
		b. creativity (demonstrate)	x	x	x	x
		c. interpretation (require)	x	x	x	x
		d. communication (engage in)	x	√	x	x
		e. embodiment (represent)	x	√	x	x
		f. image (portray)	x	x	x	x
		g. eloquence in composition (demonstrate)	x	x	x	x
		h. enchantment (instil)	x	x	x	x
		i. movement (suggest)	x	x	x	x
		j. structural elegance (express)	x	x	x	x

Appendix C Application Framework



Appendix D Interview analysis

		Organization	University 1	University 2	University 3	University 4	University 5		
TimeStamp			23-2-2021 15:00 - 16:00	23-2-2021 16:30-17:15	25-2-2021 14:30 -15:10	26-2-2021 14:00 - 14:45	04-2-2021 14:00 – 14:50		
Role			Asset manager Advisor	Senior Asset manager	Senior Asset manager	Campus Manager	Senior Asset Manager		
Topic	Q Index	Question	H.W	T.M	R.W	K.F	J.A	Correlations	User Requirements
Asset management	A1	**There are various AM management playing fields that are important for delivering the value of the asset ranging from connecting the organization goals towards the asset all the way down to delivering an optimal life cycle delivery. Out of all these AM management areas, which do you consider the most important and why.**	That is a difficult question because that depends if you look at it from our perspective, then I would look at it from our position (which aims at being sustainable) as the most important part, and that is quite a lot at the strategic and organisational level while your tool is mainly at the elemental level.	The most important thing, in my opinion, is the indoor climate, because it has to be really good; the user has to walk through the building and experience that everything is in order. I would also like to show this upon entering the building. For example, is there any fine dust in the systems? I want that because there are unjustified complaints in the building out of the 4500 occupants, 1% have delivered complaints. Those complaining are there because we have not explained some things properly about how things work. For example, we have a bandwidth between cooling and heating. You can heat below 20 degrees but not above 20, and you cannot cool below 24 degrees either. The idea behind this is that you consider opening a window to regulate your temperature. As far as I'm concerned, that's the biggest mistake we've made is in explaining how to use such a building. Secondly, as an end-user, I would like to know whether there are places we can reserve and work all day. From there, I want to make choices.	That is a difficult question. If you ask me as an asset manager, yes, the most important thing is which elements in your BIM model can be modified in such a way that they move unnoticed with the developments in education. Education from 1950 is not the same education as now. the most important thing then is the way in which you can or cannot modify and which parameters you measure in your models that have a predictive value about where you should or should not modify, and then I think those are found in the activity pattern of the people so in this case really the end-user from there onwards you should try working upwards	I am less on the technical side; we want to think more in terms of life cycle costing and less in terms of investment costs alone. In my experience, the initial investment is considered more than the OPEX costs. However, our university has two separate worlds: on the one hand, you have the technical performance with the technical specifications of the building, and on the other, the linear process of the user is the main focus of attention. We try to link the two, but it is not yet well organised.	Ultimately, the customer comes first, and the customer, in this case, is the faculty, students and professors who are conducting research. But you have to do it in a way that you are a sustainable university for the future.	Activity patterns, being able to make modifications, sustainability and organisational level, life cycle costing, technical performance, the process for the user, indoor climate and user satisfaction	
	A2	**What is value delivery according to you?**	If I look at this from our organisation's point of view, I would again raise sustainability realisation as one of the values, whereas for other parties, it would be what they consider to be the most important for their organisation. Value is strongly linked to your business goals	I don't know about value delivery, but it is about delivering a piece of quality, and that is mainly about indoor climate and the well-being of occupants, and the comfort of the equipment that the user has to use. Value is also about making assets understandable for the end-user.	In that case, you have to deal with the specific situation of educational buildings, and certainly, if you are talking about an Atlas building or another university building, the marketability of that real estate is very limited because in the traditional real estate world, it is about location and then other aspects are added that highlight specific situations; in your research, the location is less important in my opinion, and it is much more about the marketability of the adaptability, but the campus is not going to change just like that; the location stays, but what happens on the		For value creation, the focus lies on making efficient and effective use of the buildings whereby the changing needs of the primary process are practically addressed. In this, the primary	linked to business goals, quality of the indoor climate and wellbeing of occupants. Making asset understandable. The functionality of the building	

					campus does. If I look at our portfolio, which is that of University X, I can say, in gradation, that I will continue to invest in the portfolio of buildings because, in terms of marketability, they can be modified in such a way that in 5 years' time, in 10 years' time, in 25 years' time, they will still be a university building, so to speak, and that is the basic premise: it must remain a university building, but what layout it has or how it changes can shift with it because the end-user is constantly changing. So, value depends mostly on the functionality of the building		process is research and education valorisation. Where valorisation stands for the opening up of the internal knowledge of the university. We deliberately call them in that order, first research since that is where the money comes in, then opening up to society, where we share our knowledge. Finally, the focus is also on future-proof quality and objective assessment framework in your portfolio approach, that way, we can proactively respond to developments in the primary process		
	A3	**Do you consider asset information management to be a topic that is undervalued?**	Yes, we see that too little is actually done with the data that is available; sometimes, people do not even know how many square metres their building portfolio contains or too much is done with drawings.	No, I think we are falling short there because most people don't understand the building, and that gives a negative image of the building. Because the experience was not well finished and that had a negative impact on the user. Not even 10% use the app to control the lighting while it has many possibilities, such as asset tracking.	Yes, and of course that is because you want to know more, and what information is that, and then I come back to usage patterns - how intensively are buildings used throughout the day and for what activities?	I don't think we know exactly what we want to do with it yet. On the one hand, you have what you want to do, and on the other hand, you have what you can do with it. We see more and more suppliers setting up systems to show what they can do, but real use is still in its infancy. At Radboud, we mainly want to focus on the use and occupancy of our	We have not been at it that long. It is all still in development. On the other hand, we want to know what is happening to our buildings. Currently, we only monitor energy. We do this at the building level and not by room, while there is much to be gained there.	inefficient data usage, usage patterns of the buildings (occupancy),	

TU/e

Current Asset management IT Tools					workstations and less on the energy.			
	B1	**Do you believe that BIM is a way to solve certain problems within the asset management field?**	I can certainly see the future in this, but hardly any of the models become an exploitation model. They are often used for design or at least as-built. What you do is even a few steps further, but most are not that far.	The step towards the people must be logical. What can I do with this? If you use a BIM model to show the occupancy at the coffee machine, you might be able to say I'll wait. So you have to listen carefully to the people who come into the building so that they feel safe in such a building, and then you create value for the end-user.	BIM, as far as I know, does not, and as far as we use it, it's more with the engineering and maintenance. That's also where your sensor technology is. Whereas when it comes to the predictability of the future value of a property, the usage information is, in my opinion, much more relevant, also in terms of predictability for the maintenance cost or the operating cost in your schedule, where you pointed to the OPEX costs if you know how often and intensively a space is used, you know where you have to intensify cleaning costs or where the abrasion is higher, so again the intensity and type of usage of space is for me actually the starting point of all information needs	That depends. I think BIM will be the future in the long run. In the short term, not yet. Moreover, it's a broad spectrum - are you into the tooling, the sensors or the information exchange itself? At the moment, a lot of thinking is done based on the model, but only up to the construction phase; BIM is hardly used for operational management. If we have it on track within operational management, it becomes very logical to also link entire sensors to building information, and that seems to me to be the future.	I think it's a way, and it's more than just a model ultimately. It's about how many things you can tie to it. How far am I prepared to monitor that utilisation, and what you also indicated was how you were going to monitor that utilisation, then you come to a conclusion. Your conclusion may be that my utilisation is 40%, but what are you going to do with it and if so, is the organisation prepared to do something with it.	Yes will be the future, but currently not made for the O&M phase. Potentially suitable to link to usage and occupancy patterns. Potentially be used to predict the future value of property	**link to usage and occupancy**
	B2	**Do you see limitations in current CMMS or BMS, and other software systems for Asset management**	Yes, I don't know them well myself, but I do see limitations in accumulating and keeping your data up to date. There are some that do bits and pieces, but I don't see it becoming fully integrated yet.	-	-	-	-	not fully integrated systems and trouble keeping data up to date	**integrated system with up to date data**

Sensor Data	C1	**To what extent is sensor data important for you as an asset manager to fulfil your tasks**	In education, it is useful to know your occupation so that you can deliver a better client experience. And that means you need better control of the climate in order to deliver a better client experience. For your asset management, you need much more data about your systems and how intensively everything is being used. So, all that data together is important for your asset management. The richer your database, the more natural your outcome and the better you can make your decisions.	I find the occupancy particularly useful, that when I enter the building, I can immediately see which workspace is occupied. The internal climate and energy performance are also important; as an asset manager, I find that particularly interesting to see what happens to a building, but you have to visualise it in a sustainable way so that people also understand it. For example, we use 6 watts per square metre for lighting. That means nothing to the average person, but if you reflect this in comparison to last year or in comparison to other spaces, that is valuable for the end-user.	We defined our living lab as following; our building is well-building certified on the silver level, so not only BREAAM and that's where my point comes from BREAAM is really technical installation there it's about whether you have used the right heat pump and how that is controlled, but well-building certification is about how satisfied users are in the building and how they experience the building and what you have deployed in terms of, say, more specific measures and technology, and that's where we have deployed sensors. Building certification is about how satisfied users are in the building and how they experience the building and what you have deployed in terms of, say, more user-specific measures and technology, and that is also where we have deployed the sensors, which are not only temperature and climate but also usage patterns and satisfaction, and you can also see how people move through the building. So, we find sensor data useful to gain insights into usage patterns	It's a bit of a search, isn't it, of what do I want and what is there? At the moment, we don't do that much with it. A lot is still based on feeling and experience. That's just the practice. The only exception is energy. From what I just said, there are really clear and much better sensors in how I think that we as a real estate organisation should increasingly be in line with what our primary process requires, so I see a lot of sensing in the field of utilisation of facilities, finding things like that. You also have to use your building much more efficiently, which was already the case before corona, only now we see an acceleration of the process. Because we know that after Corona, everyone will stay at home for one or two working days, it is not the case that entire sections of a building will be closed down, only that you will become less and less dependent on your building policy. So we are more efficient with our occupancy time and use.	There are some that we use, but they are rare. We do think they are important because you can monitor occupancy better and control your building lighting, and if that comes from the sensor, that's of great value.	occupation, client experience, better climate control, system data for insight in intensity usage, user satisfaction, usage patterns	**occupation, client experience, better climate control, system data for insight in intensity usage, user satisfaction, usage patterns**

	C2	**imagine you get an abundance of sensor data on what errors are currently experienced with your system that you believe should be solved in the short term**	It starts with the question of what do you want to achieve by retrieving that data? So, first of all, what are you going to analyse so that you know what you want to retrieve. It seems to me that the university of Eindhoven strives for satisfied students and professors, so they benefit from knowing how occupied the building is. So, what do you want to do with your sensors? You must also reach people. A more expensive lighting installation does mean that you have to replace the lights less often and therefore achieve a higher level of satisfaction.	That depends on your sensor data and its accuracy. Within Atlas, the lighting is not everywhere in the right place. At some workstations, the lighting can be seen behind the workstation, which gives the wrong information. This is particularly true for the 11th floor. We can also see the utilisation, for example, but sometimes there are two sensors that measure one person twice.	Before we realised the living lab, we ran a number of pilots. We then installed systems to measure utilisation and occupancy. What we came up against was that there are so many different systems, some of which measure it via a sensor in the lighting, and others do it via a whole forest of technology. So we subsequently also carried out a pilot project with, in fact, a stand-alone sensor that was pinned loose and then we saw that the complexity of the systems was somewhat mutually exclusive and that what you observed earlier was that the platforms are capable of so much but that they don't talk to each other very well yet and so the reliability of the data is very limited.		I would, first of all, have a data analyst look at it, and from there, see which ones are relevant. So which buildings score the most poorly or which ones score well. Which buildings consume the most energy per square metre, for example, or which rooms have the most relative no shows, but which faculty may have the most no shows? Which buildings are the most heavily used?	Ask yourself what do you want to achieve with this data? how is the quality of the sensor data, does it interact well with each other, what do you want to analyse. Sensor readings must also reach the end-users.	**Reach end-users and judge the sensor quality**
Building Performance	D1	**What does building performance mean to you**	You emphasised that in your presentation very well, value comes from the balance you find for your portfolio. You also want to find the balance in the number of indicators you want to measure, and that is again client specific. For some, costs are more important than for others, or sustainability, or perceived experience. It is true, however, that we always strive to find the maximum achievable balance.	For building performance, we now measure the CO2 and the temperature in the building, which are benchmark values that you use for sensor data. The idea of the windows, for example, is that just before sunrise, the windows are fully opened to bring the outside temperature inside and to cool it down so that you reach the Co2 values of 500/600 at which you want to be 800 is already too harsh. This means that you can turn on your ventilation much later and therefore save a lot of kilowatts and thus energy. This is possible because the windows are set in such a way that they regulate a lot of the climate and cost little energy. The whole purpose of the climate installation was to control as much climate as possible with as little installation as possible. In terms of air velocities, we use the nominal values calculated for when the windows are open. However, this is not being operated. We accept that the building cannot be 100% temperature-controlled because of the many internal differences. We now consume 3 million kilowatts per year in the building, which	For me, building performance means the efficient use of the building. My position at the university is that for every metre we don't need, we don't have to use it, and we don't have to make it sustainable, so we really want to make a move towards reducing the number of square metres. Cost is a very important element in this, and I can use systems to manage operating costs efficiently. That is only possible if you use square metres efficiently. We, therefore, follow the guidelines in the following decision sequence: how many people do I have in the building, how efficient is that utilisation and how satisfied are those users. The appreciation of the end-users is just as important as when you manage the costs. That is why we measure comfort as an internal climate, for example.	Yes, building performance provides for a number of things. Energy is one important, but also operational safety. No continuity is an issue. That is because of the number of failures in climate conditions, and that is an important theme in our world, and so I think that most of the complaint or failure or reports we get are about climate conditions, so if that is really an important example, I think the building is doing in that area because it should do or not yes. With climate conditions, I mean great comfort for the users, is there enough ventilation, is the temperature right, is	If the faculty can house its primary processes in it, then a building performs for us, and it is up to us to ensure that this can be done in the long term so that on the one hand you are flexible, and on the other hand, you can do this in the long term.	Find balance in the number of measurement indicator, also its client specific. Efficient use of the building. Energy measurement but also operational safety and continuity.	

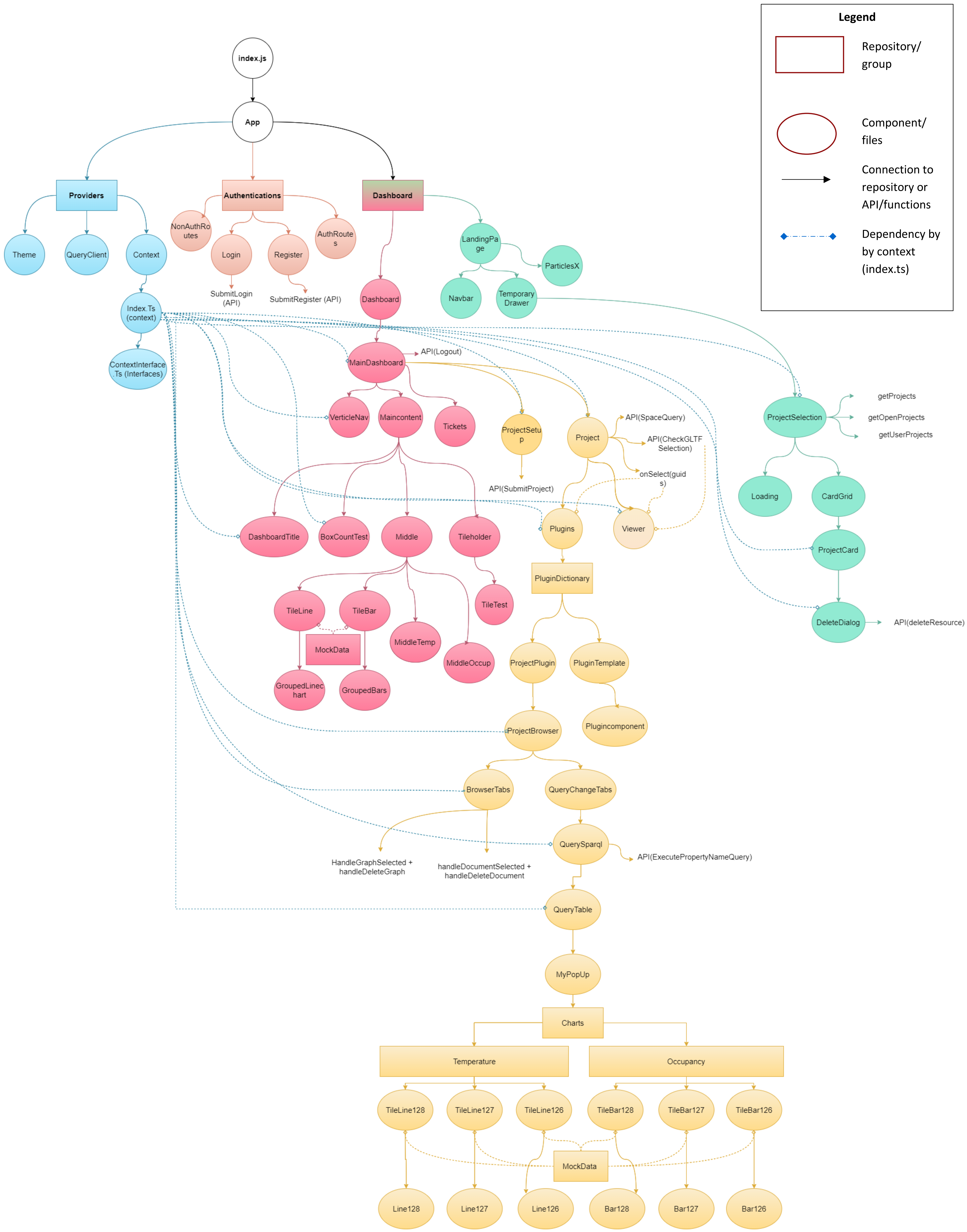
				means that the building complies with the 2050 climate agreement.		there a lot of Co2 that sort of thing and			
	D2	**Are there any margins that you would consider when you measure building performance or use any calculations?**			No indexes most come from experience	there is no single hard standard because it varies from building to building. One building is not like another, and one building can simply provide better climate conditions than the other.	Yes, we have some pre-defined KPIs which we use to benchmark	500/600 ppm CO2 and 800 is too high, nominal air velocity values, the building can't be 100% controlled, between 20-23 degrees can't heat or cool, <20 it is possible to heat and >24 to cool,	**500/600 ppm CO2 and 800 is too high, nominal air velocity values, the building can't be 100% controlled, between 20-23 degrees can't heat or cool, <20 it is possible to heat and >24 to cool,**
	D3	**Which measurement Indexes do you use**	We use different NEN standards for this.	No indexes. We do a lot based on experience. What we do know is what the building did before the renovation, and based on that, we have made decisions on what we think is acceptable and what is not.				no indexes most come from experience	
Linked Data	E1	**looking at the concept of linked data do you expect that in the near future link data may offer solutions for you as an asset manager to connect the organization goals with the asset itself**	That is difficult because organisational objectives are less concrete or measurable than spatial aspects that have clear parameters. On the other hand, I do see possibilities if, for example, on the basis of linked data, you are able to find ways of locating each other more easily in the building, that sort of possibility should be sought, and if we can test it to see to what extent these building supports this. Then most certainly this would add value to Asset management.	Yes, I see that as an opportunity to link more organisational goals to the user satisfaction with the asset. Where, for example, in the classroom you can use a system to indicate whether you are satisfied or not. I see many advantages in this, just as there are rooms that are in use or not in use that need to be cleaned.	I think that is very important and I would go a step further I would also like these systems to have a predictive value as to where people can find their place if that ensures that I can close off parts of the building so that if you come to the university you can say I have to be in room so much today and a certain floor is empty according to the schedule then it creates an increase in the use of space I would like to have a predictive value so that when people come to the building they do not always go to their own place but to a place where there are colleagues. I would like to have a predictive value so that when people come to the building, they don't always go to their own place but to a place where there are colleagues because they have seen from your schedule that you are coming for a meeting or for a lecture. I also want to know that if it's going to be hot tomorrow and I start cooling at night, that this kind of technology is going to be more automatic and thus more responsive to usage patterns.	No, I think it is good to use sensors. We can then see how long someone is sitting in such a room and whether that is sufficient and if there is almost no ventilation, yes, people will get a headache, then we can easily link them and we can also see remotely how warm it is in the building or in the room, we have quite extensive systems for that and one building can do that better than another because it is very much dependent on what the sensor exactly measures is that the temperature in the room itself or is that an average or for the space that is of	Yes, in that case you can design much more efficiently. In the end, it's all about efficiency, as you then get to know that you may not need 2,000 square metres, but that the building should be able to do fine with 1,000 square metres. If you can do all that with that 1000 square metres, then of course that's great. Maybe you can monitor that in this period nobody is using that building why is the central heating on why are the lights on? should the	Yes, locating each other more easily provides benefits. Yes, align user satisfaction with assets towards organisational goal. Yes, Predictive value when connecting with the schedule of students,	**aim at occupancy**

						course very different and that is what the user experiences as value was it what the system measures that can still be differences and I think that it is mainly on that side that valuable and a lot of data for us in case are of added value	security be there? all those kinds of things, you can report complete logistics grid, personnel, and organisation		
	E2	**If hypothetically, for example, we would connect various organizational systems through linked data with each other what value would this offer you as an asset manager to conduct your tasks.**		Yes, by carrying out a benchmark or something like that. It would be so interesting if we could see through the internet connected to objects from other universities how your building is performing compared to another asset management managed buildings. This is less interesting for the user though. Moreover, if you buy energy from a central party and charge it by faculty, this kind of concept would have a great impact on your operations. Because you would want to know what is happening in the market around you. As an asset manager, you want to know above all whether your rooms are being used or perhaps misused, and whether there are too many people in a room, which is particularly topical now with COVID-19. With linked data you could, for example, make connections between these via the internet or locally, but this does enable you to gain new insights.	I keep repeating it, but this makes the user flows more transparent and allows us, for example, to manage more asset traffic flows, thus increasing your user cooperation. That's because you can develop possible clusters based on that data. By clusters, I mean where all your colleagues from real estate or mechanical engineering etc. are. So actually, by using the data information from your asset, you are going to regulate your traffic flows in the campus, which I find very interesting to know.	Yes, that depends a bit on what type of data. I think it's useful to look at how your building performs on a number of dates compared to other buildings, in terms of energy consumption or maintenance costs. So, for the assessment, my building can be benchmarked against the norm or against others. The question that remains is what do you do with that data and to what extent will it yield something for my policy?	Yes, that would be great, that would be really great if you could eventually know things like that at the push of a button. That's great, and I think you can dream about that endlessly; you can know what your building is going to cost from the moment you start developing it, the first drawings. You can monitor your energy consumption and that means you know what that means in relation to the users who are in it. I think the possibilities are endless. You can make sure that your catering works well right through to your waste.	Yes, by carrying out benchmarking with other assets of other universities, yes for user flow and occupancy that would allow us to increase user cooperation and form clusters,	**access towards other data sources**

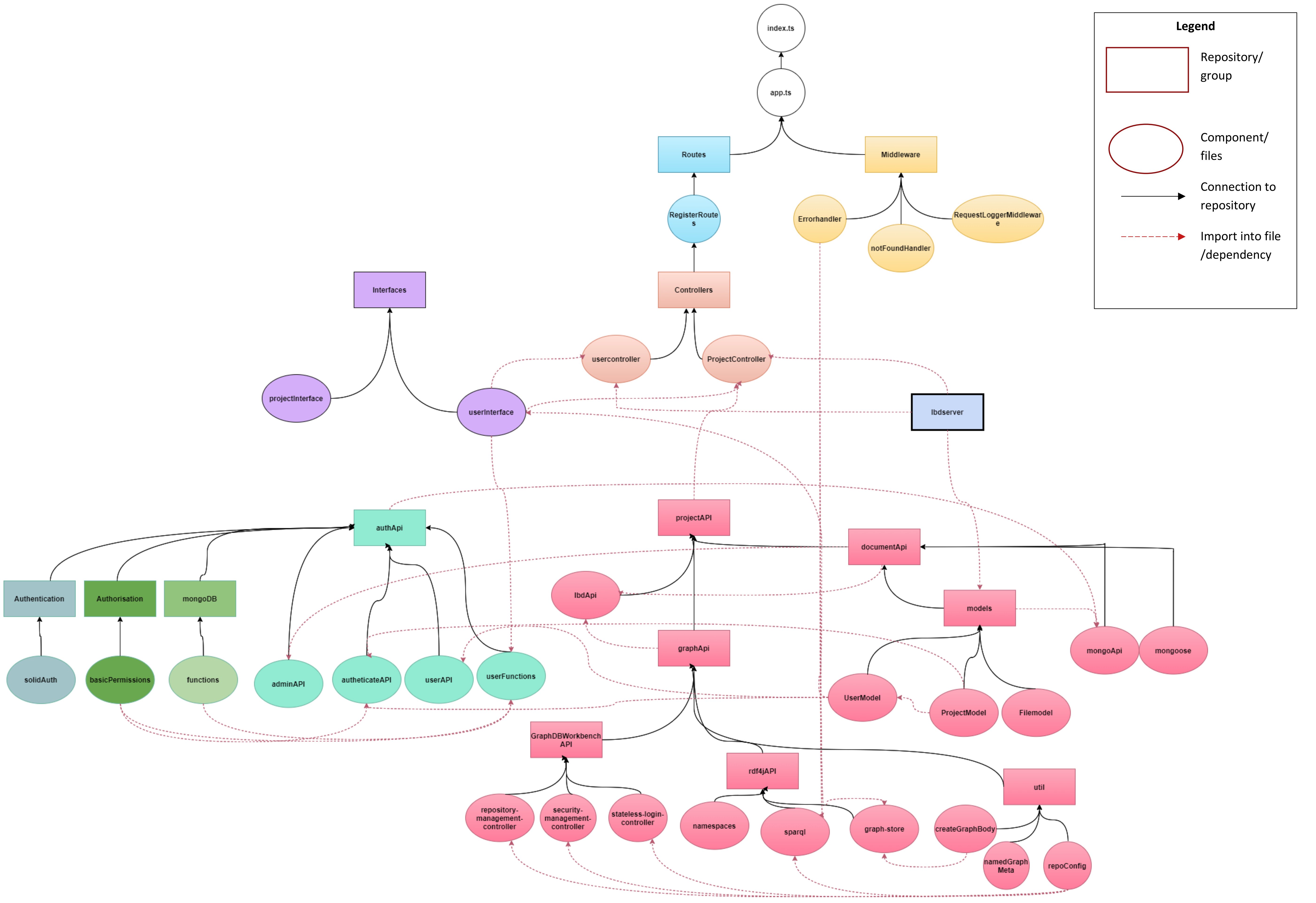
	E3	**My system aims at creating a linked building data system which connects to a digital twin and outputs in a user interface the critical sensor data on various building performance aspects such as in the Environmental Quality occupancy rate and energy use if for example would be able to connect this data to the web and pull or in the most ideal situation connect with various data sources from the web (maybe a school roster or other open source data sources) what value would this create for you as an asset manager**	Now that is really interesting. I see a lot of added value for a school as an actor in the fact that you can link a digital school timetable data source to your actual occupancy in the building. Because, in principle, you would be able to predict well in advance how busy your building will be on the basis of that timetable and, in this way, you would be able to use your systems more efficiently and intelligently. The moment you can have your building react to internal use or to the weather, then you can actually do so much that is currently not possible, and that will reduce your costs. Moreover, it is interesting to see to what extent you can automate this and make it less and less dependent on parties or people.	First you should ask yourself what you want to know and how reliable is the sensor data? Is the data validated? That is what I would like to know. Then you can apply the data and make decisions based on it for a specific purpose. However, you must take into account the sensitivity of the data in accordance with AVG.	Is a tricky question, the decision then becomes about the choices of what systems you introduce rather than the occupancy rate and the energy consumption you had and the internal climate of the building because those are the things you measure from your experience. I think it only works if the end user experiences a benefit from it. It is an added value when the end user also appreciates it, because he can then find quieter workplaces or experience better comfort. It is the task of the asset manager to offer the asset in the best possible way to those end users.	To be honest, we are not involved in that way at all. But I can imagine that it says a lot about the use of your building, and how it is occupied. These are very interesting things, which may enable us to have predictive data. What you can then do is adjust your climate system accordingly, for example, based on that use and occupancy or based on a school timetable that is online. The latter in particular can bring you all kinds of benefits.		linking to digital school timetable would add a lot of value, you must take into account the end user and the data quality, And there is a lot of gain to be made from the occupancy	**Occupancy linkage towards digital school timetable**
Decision Support	F1	**Let's say for example you would use a digital twin, based upon the sensor input you will get which decisions would you make let's say for example that you'll get critical sensor data about the indoor Environmental Quality, what baseline values do you use to measure this aspect and if this aspect has been overwritten how do you manage this problem.**	You are asking the right question here. The asset manager uses that data as input to make a fact-based judgement on what he should do during the building's lifecycle, should he renovate or dispose of it on the basis of satisfying users. Financially the building may not be profitable then you start looking at disposal or sustainability. There is so much information in your systems to build a thorough foundation and make your decisions with. What you also want to know is how intensively something is used, so that you can better plan and motivate your maintenance. By default, the more data you collect, the richer your decisions, the richer your insights and the better your predictive ability.			You are asking the right question here, when you have all the information, what do you do with it and what does it bring. I think it teaches us above all how to improve things. In terms of energy consumption, it would help, because we have a big target in the coming years. Everything that can contribute helps, only practice teaches us that the use of the building gives more insight, that's where your efficiency lies. From my own role, I find it interesting how our spaces are used.	the decisions that you could make is that you buy a building or maybe you sell it, or you have to renovate a building that you have, that you talk to the faculty and say, hey, we see that you say that you need space, but this data shows that you don't. Suppose you come to collect waste and when you know that the waste is full because the containers are full instead of you just filling in a certain work schedule and that	what do you want to measure why do you want that how is the data quality what can we do to improve what is occurring at this event	

	F2	**I'm looking for the decisions that you make based upon the information that you would get from certain systems could you take me with the line of thought that you consider when you make a decision based upon these sensor data**	If you manage to set this up, you make it less objective and much more fact oriented. But for now, as an asset manager, I use mostly experience and gut feeling. On the other hand, we do not often use KPIs. The use of KPI for energy performance, occupancy rate strongly depends on what you find important and why you want to steer on this and that all depends on your target group.		Yes, look, that's where I started, there is always a financial component, because everything must ultimately remain affordable, because the user is not always the one who feels immediately at the expenses, so they have different expectations. But if you offer the end-user more comfort, and comfort is not just air and light, but also making it possible for him to meet others, then you are responding to the right need. as an asset manager, you must be able to facilitate this meeting, but the added value would be that this then becomes an intention based on data, because a lot of decisions are now made based on experience. Because there is not enough data now, the data that is available is not of such a quality that you can make a really well-founded decision on it. There is also a human component behind it, we always think we know better, so our confidence in the system must be higher and higher to accept its decisions as decisions.	If you know that rooms are not used or are used infrequently, you could look into how to rearrange them. For example, we want to know how the study places are all occupied and whether there are any complaints from students about their workplaces. So that we can look into this and assess why these workplaces are not being used. Are they difficult to find or is the environment and lighting not pleasant? Such a line of thought helps you to deal better with your space.	every time the container is only half full it is collected again. So what I would do is, indeed, my first step would be to scan where the most energy is consumed and where is the biggest mismatch between rostering and actual staffing so you really do go based on low hanging fruit and then you have a discussion with the faculties	more fact orientated, however most decisions come from experience. Due to insufficient data or weak data quality. lack of trust in technology.	
	F3	**if you make these decisions do you use any KPIs calculations which are relevant for measuring aspects such as indoor Environmental Quality occupancy rate energy use if so, are you willing to share the calculation that you will make**		No, we have never used KPIs in any respect. For example, I am now working on room utilisation and that we can at least indicate the no-show in Atlas, and the total reservations. That when a person does not show up, space becomes available again via book myspace. We are trying to measure these using sensors. We are using new methods to gain this kind of insight into the use of space, which could be valuable for future policy on campus objects. But we don't use KPIs, we mainly do so on the basis of experience.	No, not really. In this case, practical experience is your teacher.	no KPIs		No KPIs	

Appendix E Component Framework (Front-End)



Appendix F Component Framework (Back-End)



Appendix G Sensor inventory

TTL space	Sensor ID	Room number and sensor type	Sensor ID	Room number and sensor type	Sensor ID	Room number and sensor type											
892	11NR008TE-001TRL	RUIMTETEMPERATUUR 8_128	11NR008LT-001PIRTM	AANWEZIGHEID 8_128	11NR008TE-001CPA	CPA VIA WANDMODULE RUIMTE 8_128											
1023	11NR008TE-003TRL	RUIMTETEMPERATUUR 8_127	11NR008LT-003PIRTM	AANWEZIGHEID 8_127	11NR008TE-003CPA	CPA VIA WANDMODULE RUIMTE 8_127											
1144	11NR008TE-004TRL	RUIMTETEMPERATUUR 8_126	11NR008LT-004PIRTM	AANWEZIGHEID 8_126	11NR008TE-004CPA	CPA VIA WANDMODULE RUIMTE 8_126											
1526	11NR008TE-005TRL	RUIMTETEMPERATUUR 8_Z02 (8,124+8,123+8,122	11NR008LT-005PIRTM	AANWEZIGHEID 8_Z02	11NR008TE-005CPA	CPA VIA WANDMODULE RUIMTE 8_Z02											
1656	11NR008TE-010TRL	RUIMTETEMPERATUUR 8_121	11NR008LT-010PIRTM	AANWEZIGHEID 8_121	11NR008TE-010CPA	CPA VIA WANDMODULE RUIMTE 8_121											
1776	11NR008TE-011TRL	RUIMTETEMPERATUUR 8_119	11NR008LT-011PIRTM	AANWEZIGHEID 8_119	11NR008TE-011CPA	CPA VIA WANDMODULE RUIMTE 8_119											
1912	11NR008TE-012TRL	RUIMTETEMPERATUUR 8_118	11NR008LT-012PIRTM	AANWEZIGHEID 8_118	11NR008TE-012CPA	CPA VIA WANDMODULE RUIMTE 8_118											
2056	11NR008TE-013TRL	RUIMTETEMPERATUUR 8_201	11NR008LT-013PIRTM	AANWEZIGHEID 8_201	11NR008TE-013CPA	CPA VIA WANDMODULE RUIMTE 8_201	11NR008QT-013CO2	*CO2 METING 8_201	11NR008FT-013FLW	FLOW TOEVOER VAV 8_201	11NR008RS-013VT2RS	REGELSIGNAAL TOEVOER VAV 8_201	11NR008RS-013VT3RS	REGELSIGNAAL TOEVOER VAV 8_201	11NR008RS-013VT4RS	REGELSIGNAAL TOEVOER VAV 8_201	11NR008RS-013VTVRS
19278	11NR008TE-016TRL	RUIMTETEMPERATUUR 8_Z03 (8,205+8,206)	11NR008LT-016PIRTM	AANWEZIGHEID 8_Z03	11NR008TE-016CPA	CPA VIA WANDMODULE RUIMTE 8_Z03											
3283	11NR008TE-018TRL	RUIMTETEMPERATUUR 8_208	11NR008LT-018PIRTM	AANWEZIGHEID 8_208	11NR008TE-018CPA	CPA VIA WANDMODULE RUIMTE 8_208											
3412	11NR008TE-020TRL	RUIMTETEMPERATUUR 8_209	11NR008LT-020PIRTM	AANWEZIGHEID 8_209	11NR008TE-020CPA	CPA VIA WANDMODULE RUIMTE 8_209											
19278	11NR008TE-021TRL	RUIMTETEMPERATUUR 8_Z05	11NR008LT-021PIRTM	AANWEZIGHEID 8_Z05	11NR008TE-021CPA	CPA VIA WANDMODULE RUIMTE 8_Z05											
3699	11NR008TE-025TRL	RUIMTETEMPERATUUR 8_304	11NR008LT-025PIRTM	AANWEZIGHEID 8_304	11NR008TE-025CPA	CPA VIA WANDMODULE RUIMTE 8_304											
3824	11NR008TE-026TRL	RUIMTETEMPERATUUR 8_305	11NR008LT-026PIRTM	AANWEZIGHEID 8_305	11NR008TE-026CPA	CPA VIA WANDMODULE RUIMTE 8_305											
4736	11NR008TE-027TRL	RUIMTETEMPERATUUR 8_Z06a	11NR008LT-027PIRTM	AANWEZIGHEID 8_Z06a	11NR008TE-027CPA	CPA VIA WANDMODULE RUIMTE 8_Z06a											
4116	11NR008TE-029TRL	RUIMTETEMPERATUUR 8_335	11NR008LT-029PIRTM	AANWEZIGHEID 8_335	11NR008TE-029CPA	CPA VIA WANDMODULE RUIMTE 8_335											
4236	11NR008TE-031TRL	RUIMTETEMPERATUUR 8_333	11NR008LT-031PIRTM	AANWEZIGHEID 8_333	11NR008TE-031CPA	CPA VIA WANDMODULE RUIMTE 8_333											
4365	11NR008TE-032TRL	RUIMTETEMPERATUUR 8_331	11NR008LT-032PIRTM	AANWEZIGHEID 8_331	11NR008TE-032CPA	CPA VIA WANDMODULE RUIMTE 8_331											
4485	11NR008TE-034TRL	RUIMTETEMPERATUUR 8_330	11NR008LT-034PIRTM	AANWEZIGHEID 8_330	11NR008TE-034CPA	CPA VIA WANDMODULE RUIMTE 8_330											
19278	11NR008TE-035TRL	RUIMTETEMPERATUUR 8_Z06b	11NR008LT-035PIRTM	AANWEZIGHEID 8_Z06b	11NR008TE-035CPA	CPA VIA WANDMODULE RUIMTE 8_Z06b											
6216	11NR008TE-038TRL	RUIMTETEMPERATUUR 8_326	11NR008LT-038PIRTM	AANWEZIGHEID 8_326	11NR008TE-038CPA	CPA VIA WANDMODULE RUIMTE 8_326	11NR008QT-038CO2	*CO2 METING 8_326	11NR008FT-038FLW	FLOW TOEVOER VAV 8_326	11NR008RS-038VTVRS	REGELSIGNAAL TOEVOER VAV 8_326					
6087	11NR008TE-039TRL	RUIMTETEMPERATUUR 8_324	11NR008LT-039PIRTM	AANWEZIGHEID 8_324	11NR008TE-039CPA	CPA VIA WANDMODULE RUIMTE 8_324	11NR008QT-039CO2	*CO2 METING 8_324	11NR008FT-039FLW	FLOW TOEVOER VAV 8_324	11NR008RS-039VT2RS	REGELSIGNAAL TOEVOER VAV 8_324	11NR008RS-039VTVRS	REGELSIGNAAL TOEVOER VAV 8_324			
5967	11NR008TE-040TRL	RUIMTETEMPERATUUR 8_323	11NR008LT-040PIRTM	AANWEZIGHEID 8_323	11NR008TE-040CPA	CPA VIA WANDMODULE RUIMTE 8_323	11NR008QT-040CO2	*CO2 METING 8_323	11NR008FT-040FLW	FLOW TOEVOER VAV 8_323	11NR008RS-040VTVRS	REGELSIGNAAL TOEVOER VAV 8_323					
6353	11NR008TE-041TRL	RUIMTETEMPERATUUR 8_401	11NR008LT-041PIRTM	AANWEZIGHEID 8_401	11NR008TE-041CPA	CPA VIA WANDMODULE RUIMTE 8_401											
6473	11NR008TE-042TRL	RUIMTETEMPERATUUR 8_402	11NR008LT-042PIRTM	AANWEZIGHEID 8_402	11NR008TE-042CPA	CPA VIA WANDMODULE RUIMTE 8_402											
6602	11NR008TE-043TRL	RUIMTETEMPERATUUR 8_404	11NR008LT-043PIRTM	AANWEZIGHEID 8_404	11NR008TE-043CPA	CPA VIA WANDMODULE RUIMTE 8_404											
6722	11NR008TE-044TRL	RUIMTETEMPERATUUR 8_405	11NR008LT-044PIRTM	AANWEZIGHEID 8_405	11NR008TE-044CPA	CPA VIA WANDMODULE RUIMTE 8_405											
19917	11NR008TE-045TRL	RUIMTETEMPERATUUR 8_406	11NR008LT-045PIRTM	AANWEZIGHEID 8_406	11NR008TE-045CPA	CPA VIA WANDMODULE RUIMTE 8_406											
19917	11NR008TE-046TRL	RUIMTETEMPERATUUR 8_407	11NR008LT-046PIRTM	AANWEZIGHEID 8_407	11NR008TE-046CPA	CPA VIA WANDMODULE RUIMTE 8_407											
6966	11NR008TE-049TRL	RUIMTETEMPERATUUR 8_409	11NR008LT-049PIRTM	AANWEZIGHEID 8_409	11NR008TE-049CPA	CPA VIA WANDMODULE RUIMTE 8_409											
7095	11NR008TE-050TRL	RUIMTETEMPERATUUR 8_410	11NR008LT-050PIRTM	AANWEZIGHEID 8_410	11NR008TE-050CPA	CPA VIA WANDMODULE RUIMTE 8_410											
7215	11NR008TE-051TRL	RUIMTETEMPERATUUR 8_411	11NR008LT-051PIRTM	AANWEZIGHEID 8_411	11NR008TE-051CPA	CPA VIA WANDMODULE RUIMTE 8_411											
7344	11NR008TE-052TRL	RUIMTETEMPERATUUR 8_412	11NR008LT-052PIRTM	AANWEZIGHEID 8_412	11NR008TE-052CPA	CPA VIA WANDMODULE RUIMTE 8_412											
19917	11NR008TE-053TRL	RUIMTETEMPERATUUR 8_413	11NR008LT-053PIRTM	AANWEZIGHEID 8_413	11NR008TE-053CPA	CPA VIA WANDMODULE RUIMTE 8_413											
8461	11NR008TE-054TRL	RUIMTETEMPERATUUR 8_414	11NR008LT-054PIRTM	AANWEZIGHEID 8_414	11NR008TE-054CPA	CPA VIA WANDMODULE RUIMTE 8_414											
8592	11NR008TE-055TRL	RUIMTETEMPERATUUR 8_417	11NR008LT-055PIRTM	AANWEZIGHEID 8_417	11NR008TE-055CPA	CPA VIA WANDMODULE RUIMTE 8_417											
19917	11NR008TE-056TRL	RUIMTETEMPERATUUR 8_418	11NR008LT-056PIRTM	AANWEZIGHEID 8_418	11NR008TE-056CPA	CPA VIA WANDMODULE RUIMTE 8_418											
7723	11NR008TE-057TRL	RUIMTETEMPERATUUR 8_419	11NR008LT-057PIRTM	AANWEZIGHEID 8_419	11NR008TE-057CPA	CPA VIA WANDMODULE RUIMTE 8_419											
7594	11NR008TE-058TRL	RUIMTETEMPERATUUR 8_420	11NR008LT-058PIRTM	AANWEZIGHEID 8_420	11NR008TE-058CPA	CPA VIA WANDMODULE RUIMTE 8_420											
7474	11NR008TE-059TRL	RUIMTETEMPERATUUR 8_421	11NR008LT-059PIRTM	AANWEZIGHEID 8_421	11NR008TE-059CPA	CPA VIA WANDMODULE RUIMTE 8_421											
1266	11NR008TE-301TRL	RUIMTETEMPERATUUR 8_140	11NR008LT-301PIRTM	AANWEZIGHEID 8_140	11NR008TE-301CPA	CPA VIA WANDMODULE RUIMTE 8_140	11NR008QT-301CO2	*CO2 METING 8_140	11NR008FT-301FLW	FLOW TOEVOER VAV 8_140	11NR008RS-301VTVRS	REGELSIGNAAL TOEVOER VAV 8_140					
6846	11NR008TE-302TRL	RUIMTETEMPERATUUR 8_445	11NR008LT-302PIRTM	AANWEZIGHEID 8_445	11NR008TE-302CPA	CPA VIA WANDMODULE RUIMTE 8_445	11NR008QT-302CO2	*CO2 METING 8_445	11NR008FT-302FLW	FLOW TOEVOER VAV 8_445	11NR008RS-302VTVRS	REGELSIGNAAL TOEVOER VAV 8_445					

TU/e

Appendix H Sample description of Temperature and Occupancy sensors

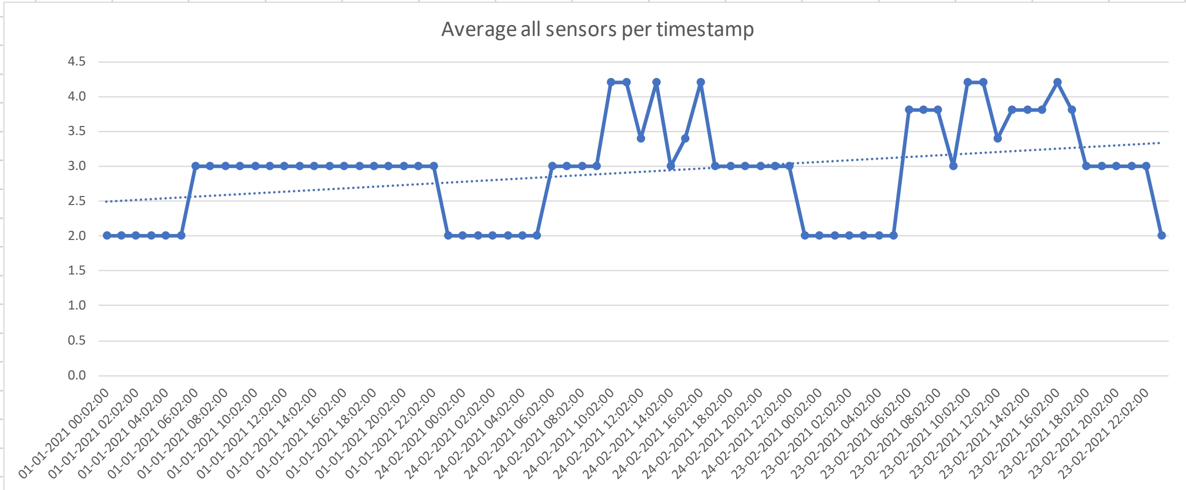
892	*inst:*	*TemperatureSensor_11NR008TE_001TRL*				*inst:TemperatureSensor_11NR008TE_001TRL*
	a bot:	Element ;				a bot:Element ;
	rdfs:	label	"Signify Smart Energy Savings Sytems : Signify Temperature Sensor:	001TRL	"^^xsd:string;	rdfs:label "Signify Smart Energy Savings Sytems : Signify Temperature Sensor: 001TRL"^^xsd:string;
	rdfs:	comment	""		^^xsd:string;	rdfs:comment""^^xsd:string;
	bot:	hasGuid"	A51C1272-8922-A53F-5389-FCD80A8B8744		"^^xsd:string;	bot:hasGuid"A51C1272-8922-A53F-5389-FCD80A8B8744"^^xsd:string;
	props:	tag"	001TRL		"^^xsd:string;	props:tag"001TRL"^^xsd:string;
	props:	reference	"Signify Temperature Sensor"		^^xsd:string;	props:reference"Signify Temperature Sensor"^^xsd:string;
	props:	reference	"Signify Temperature Sensor"		^^xsd:string;	props:reference"Signify Temperature Sensor"^^xsd:string;
	props:	level	"Level: Ceiling 8th floor"		^^xsd:string;	props:level "Level: Ceiling 8th floor"^^xsd:string;
	props:	sensorID"	11NR008TE-001TRL		"^^xsd:string;	props:sensorID"11NR008TE-001TRL"^^xsd:string;
	props:	area	""		^^xsd:string;	props:area""^^xsd:string;
	props:	volume	""		^^xsd:string;	props:volume""^^xsd:string;
	props:	mark	""		^^xsd:string;	props:mark""^^xsd:string;
	props:	category	"Temperature Sensors"		^^xsd:string;	props:category"Temperature Sensors"^^xsd:string;
	props:	family	"Signify Smart Energy Savings Systems"		^^xsd:string;	props:family"Signify Smart Energy Savings Systems"^^xsd:string;
	props:	familyandType	"Signify Smart Energy Savings Systems: Signify Temperature Sensor"		^^xsd:string;	props:familyandType"Signify Smart Energy Savings Systems: Signify Temperature Sensor"^^xsd:string;
	props:	hostId	"Compound Ceiling: Plain"		^^xsd:string;	props:hostId"Compound Ceiling: Plain"^^xsd:string;
	props:	type	"Signify Temperature Sensor"		^^xsd:string;	props:type"Signify Temperature Sensor"^^xsd:string;
	props:	typeId	"Signify Temperature Sensor"		^^xsd:string;	props:typeId"Signify Temperature Sensor"^^xsd:string;
	props:	phaseCreated"	New Construction		"^^xsd:string;	props:phaseCreated"New Construction"^^xsd:string;
	props:	typeName	"Signify Temperature Sensor"		^^xsd:string;	props:typeName"Signify Temperature Sensor"^^xsd:string;
	props:	category	"Temperature Sensors"		^^xsd:string;	props:category"Temperature Sensors"^^xsd:string;
	props:	familyName	"Signify Smart Energy Savings Systems"		^^xsd:string;	props:familyName"Signify Smart Energy Savings Systems"^^xsd:string;
	props:	reference	"Temperature Sensors"		^^xsd:string;	props:reference"Temperature Sensors"^^xsd:string;
	props:	reference	"Temperature Sensors"		^^xsd:string.	props:reference"Temperature Sensors"^^xsd:string.

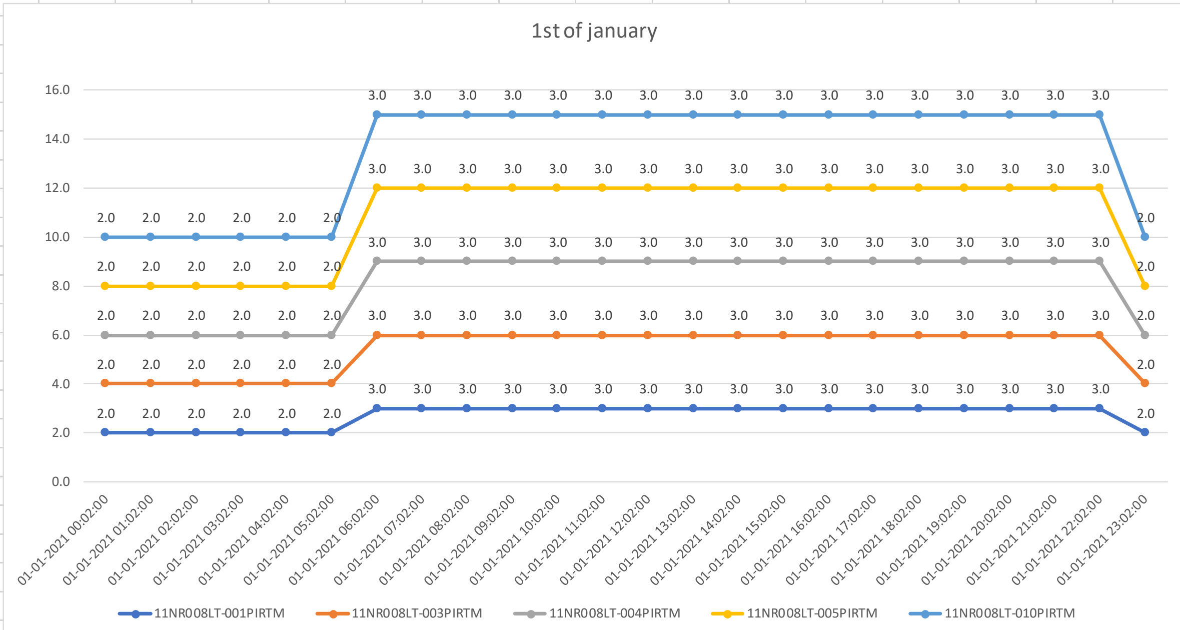
892	*inst:*	*OccupacySensor_11NR008LT_001PIRTM*				*inst:OccupacySensor_11NR008LT_001PIRTM*
	a bot:	Element ;				a bot:Element ;
	rdfs:	label	"Signify Smart Energy Savings Sytems : Signify Occupancy Sensor:	001PIRTM	"^^xsd:string;	rdfs:label "Signify Smart Energy Savings Sytems : Signify Occupancy Sensor: 001PIRTM"^^xsd:string;
	rdfs:	comment	""		^^xsd:string;	rdfs:comment""^^xsd:string;
	bot:	hasGuid"	68042A54-7BB9-2217-9380-AC74A2709594		"^^xsd:string;	bot:hasGuid"68042A54-7BB9-2217-9380-AC74A2709594"^^xsd:string;
	props:	tag"	001PIRTM		"^^xsd:string;	props:tag"001PIRTM"^^xsd:string;
	props:	reference	"Signify Occupancy Sensor"		^^xsd:string;	props:reference"Signify Occupancy Sensor"^^xsd:string;
	props:	reference	"Signify Occupancy Sensor"		^^xsd:string;	props:reference"Signify Occupancy Sensor"^^xsd:string;
	props:	level	"Level: Ceiling 8th floor"		^^xsd:string;	props:level "Level: Ceiling 8th floor"^^xsd:string;
	props:	sensorID"	11NR008LT-001PIRTM		"^^xsd:string;	props:sensorID"11NR008LT-001PIRTM"^^xsd:string;
	props:	area	""		^^xsd:string;	props:area""^^xsd:string;
	props:	volume	""		^^xsd:string;	props:volume""^^xsd:string;
	props:	mark	""		^^xsd:string;	props:mark""^^xsd:string;
	props:	category	"Occupancy Sensors"		^^xsd:string;	props:category"Occupancy Sensors"^^xsd:string;
	props:	family	"Signify Smart Energy Savings Systems"		^^xsd:string;	props:family"Signify Smart Energy Savings Systems"^^xsd:string;
	props:	familyandType	"Signify Smart Energy Savings Systems: Signify Occupancy Sensor"		^^xsd:string;	props:familyandType"Signify Smart Energy Savings Systems: Signify Occupancy Sensor"^^xsd:string;
	props:	hostId	"Compound Ceiling: Plain"		^^xsd:string;	props:hostId"Compound Ceiling: Plain"^^xsd:string;
	props:	type	"Signify Occupancy Sensor"		^^xsd:string;	props:type"Signify Occupancy Sensor"^^xsd:string;
	props:	typeId	"Signify Occupancy Sensor"		^^xsd:string;	props:typeId"Signify Occupancy Sensor"^^xsd:string;
	props:	phaseCreated"	New Construction		"^^xsd:string;	props:phaseCreated"New Construction"^^xsd:string;
	props:	typeName	"Signify Occupancy Sensor"		^^xsd:string;	props:typeName"Signify Occupancy Sensor"^^xsd:string;
	props:	category	"Occupancy Sensors"		^^xsd:string;	props:category"Occupancy Sensors"^^xsd:string;
	props:	familyName	"Signify Smart Energy Savings Systems"		^^xsd:string;	props:familyName"Signify Smart Energy Savings Systems"^^xsd:string;
	props:	reference	"Occupancy Sensors"		^^xsd:string;	props:reference"Occupancy Sensors"^^xsd:string;
	props:	reference	"Occupancy Sensors"		^^xsd:string.	props:reference"Occupancy Sensors"^^xsd:string.

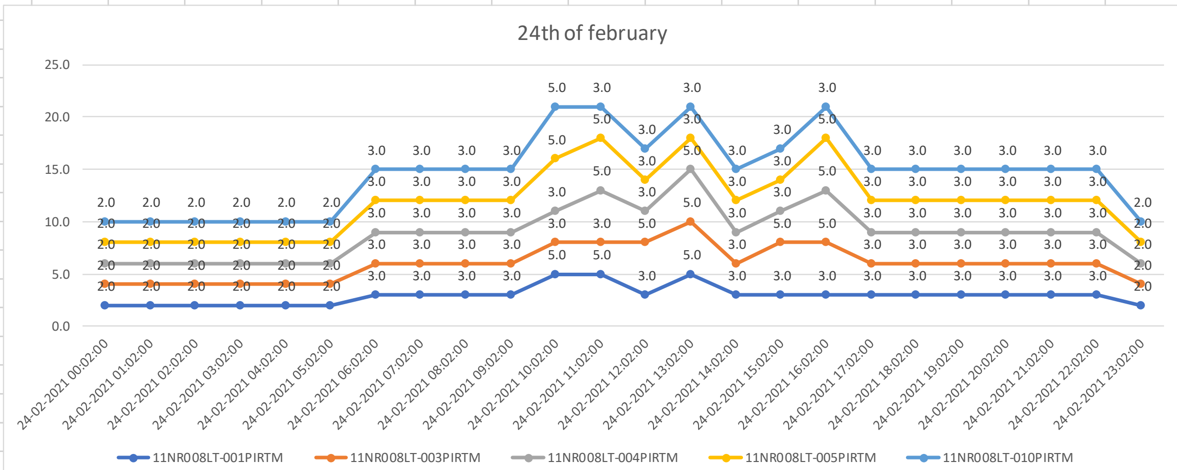
Appendix I Full Sensor analysis

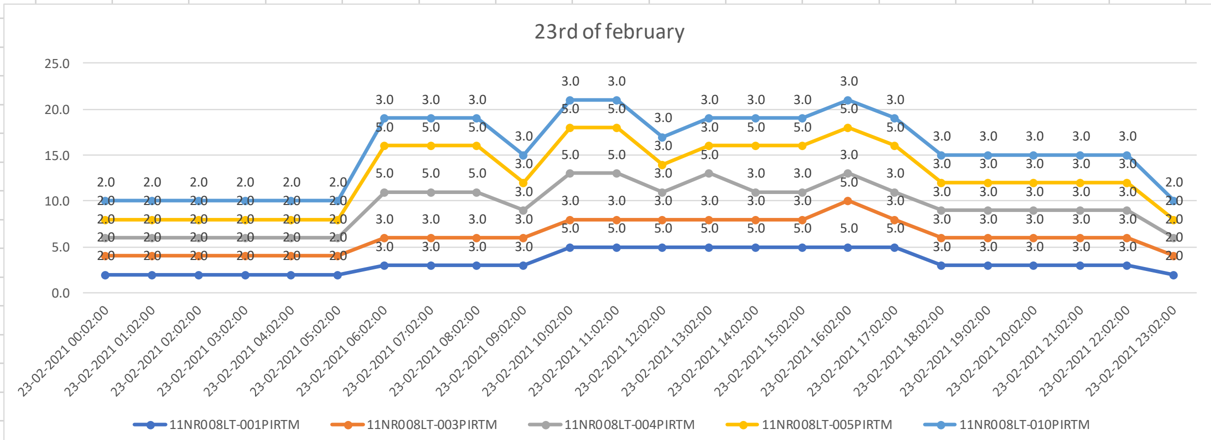
Occupancy Sensors

Timestamp	11NR008LT-001PIRTM	11NR008LT-003PIRTM	11NR008LT-004PIRTM	11NR008LT-005PIRTM	11NR008LT-010PIRTM	Average all sensors per timestamp
01-01-2021 00:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 01:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 02:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 03:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 04:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 05:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
01-01-2021 06:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 07:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 08:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 09:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 10:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 11:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 12:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 13:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 14:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 15:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 16:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 17:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 18:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 19:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 20:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 21:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 22:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
01-01-2021 23:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 00:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 01:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 02:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 03:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 04:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 05:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
24-02-2021 06:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 07:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 08:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 09:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 10:02:00	5.0	3.0	3.0	5.0	5.0	**4.2**
24-02-2021 11:02:00	5.0	3.0	5.0	5.0	3.0	**4.2**
24-02-2021 12:02:00	3.0	5.0	3.0	3.0	3.0	**3.4**
24-02-2021 13:02:00	5.0	5.0	5.0	3.0	3.0	**4.2**
24-02-2021 14:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 15:02:00	3.0	5.0	3.0	3.0	3.0	**3.4**
24-02-2021 16:02:00	3.0	5.0	5.0	5.0	3.0	**4.2**
24-02-2021 17:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 18:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 19:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 20:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 21:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 22:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
24-02-2021 23:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 00:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 01:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 02:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 03:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 04:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 05:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
23-02-2021 06:02:00	3.0	3.0	5.0	5.0	3.0	**3.8**
23-02-2021 07:02:00	3.0	3.0	5.0	5.0	3.0	**3.8**
23-02-2021 08:02:00	3.0	3.0	5.0	5.0	3.0	**3.8**
23-02-2021 09:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 10:02:00	5.0	3.0	5.0	5.0	3.0	**4.2**
23-02-2021 11:02:00	5.0	3.0	5.0	5.0	3.0	**4.2**
23-02-2021 12:02:00	5.0	3.0	3.0	3.0	3.0	**3.4**
23-02-2021 13:02:00	5.0	3.0	5.0	3.0	3.0	**3.8**
23-02-2021 14:02:00	5.0	3.0	3.0	5.0	3.0	**3.8**
23-02-2021 15:02:00	5.0	3.0	3.0	5.0	3.0	**3.8**
23-02-2021 16:02:00	5.0	5.0	3.0	5.0	3.0	**4.2**
23-02-2021 17:02:00	5.0	3.0	3.0	5.0	3.0	**3.8**
23-02-2021 18:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 19:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 20:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 21:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 22:02:00	3.0	3.0	3.0	3.0	3.0	**3.0**
23-02-2021 23:02:00	2.0	2.0	2.0	2.0	2.0	**2.0**
Average per sensor	**3.0**	**2.8**	**3.0**	**3.0**	**2.7**	



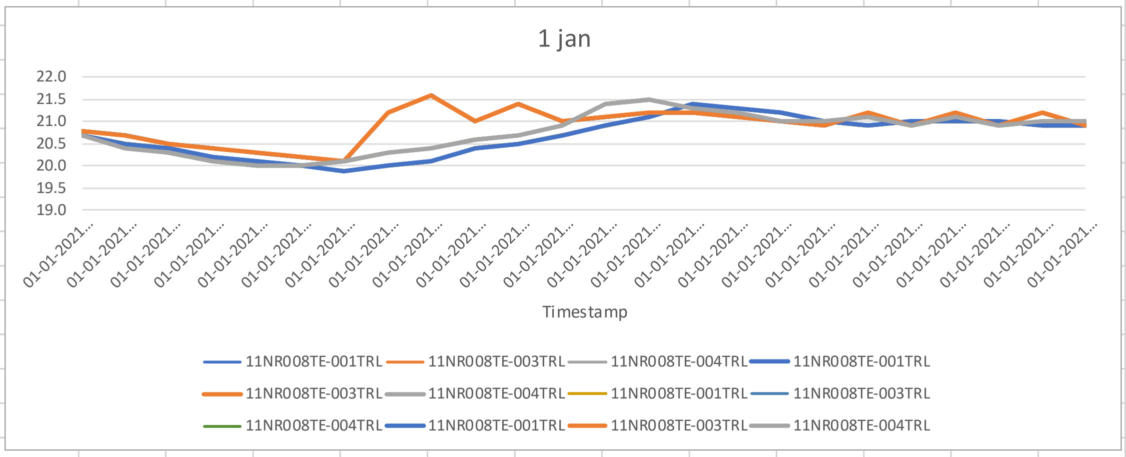


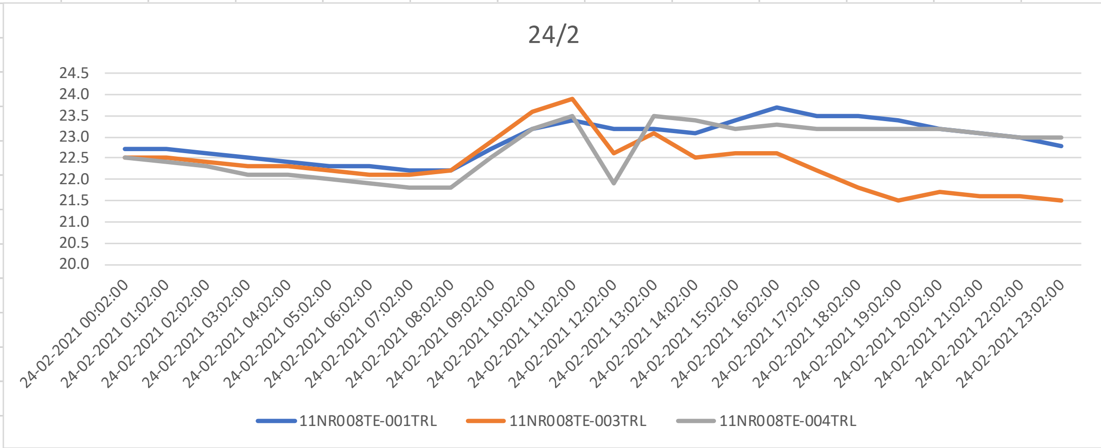


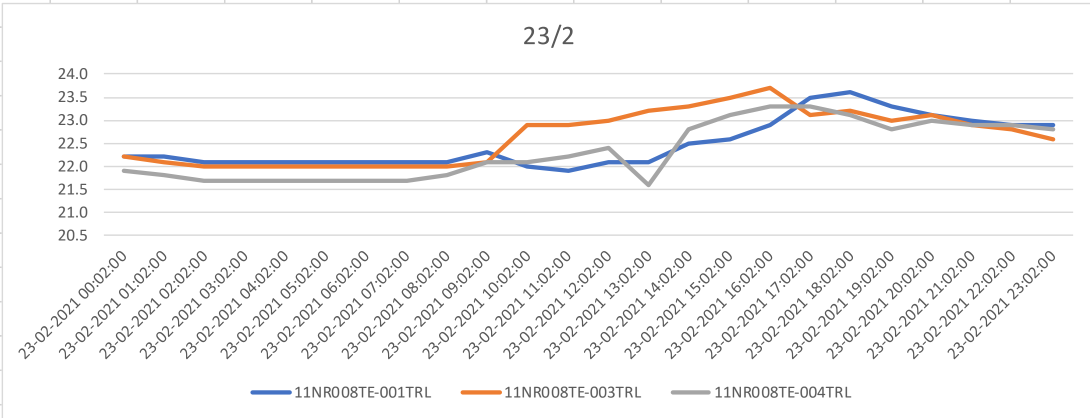


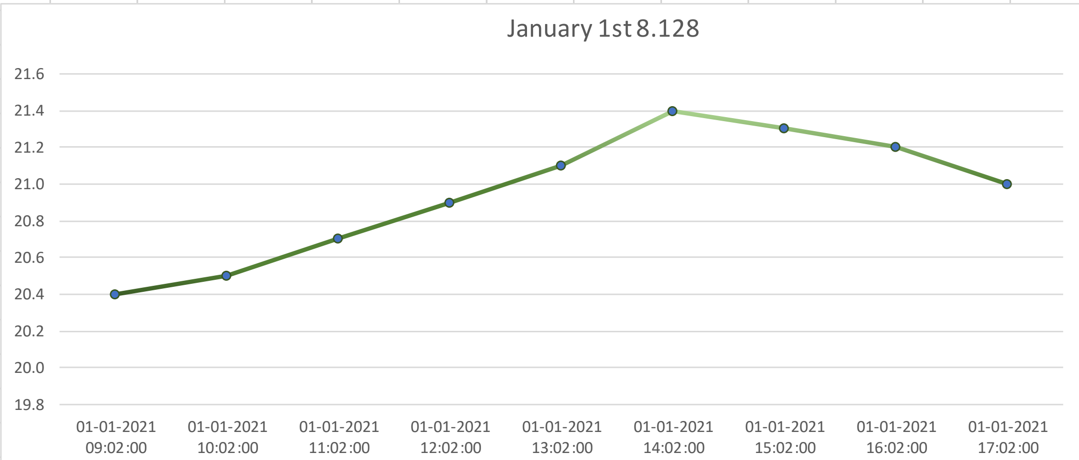
Temperature Sensors

Timestamp	11NR008 TE-001TRL	11NR008 TE-003TRL	11NR008 TE-004TRL	Average all sensors per timestamp
01-01-2021 00:02:00	20.7	20.8	20.7	20.7
01-01-2021 01:02:00	20.5	20.7	20.4	20.5
01-01-2021 02:02:00	20.4	20.5	20.3	20.4
01-01-2021 03:02:00	20.2	20.4	20.1	20.2
01-01-2021 04:02:00	20.1	20.3	20.0	20.1
01-01-2021 05:02:00	20.0	20.2	20.0	20.1
01-01-2021 06:02:00	19.9	20.1	20.1	20.0
01-01-2021 07:02:00	20.0	21.2	20.3	20.5
01-01-2021 08:02:00	20.1	21.6	20.4	20.7
01-01-2021 09:02:00	20.4	21.0	20.6	20.7
01-01-2021 10:02:00	20.5	21.4	20.7	20.9
01-01-2021 11:02:00	20.7	21.0	20.9	20.9
01-01-2021 12:02:00	20.9	21.1	21.4	21.1
01-01-2021 13:02:00	21.1	21.2	21.5	21.3
01-01-2021 14:02:00	21.4	21.2	21.3	21.3
01-01-2021 15:02:00	21.3	21.1	21.2	21.2
01-01-2021 16:02:00	21.2	21.0	21.0	21.1
01-01-2021 17:02:00	21.0	20.9	21.0	21.0
01-01-2021 18:02:00	20.9	21.2	21.1	21.1
01-01-2021 19:02:00	21.0	20.9	20.9	20.9
01-01-2021 20:02:00	21.0	21.2	21.1	21.1
01-01-2021 21:02:00	21.0	20.9	20.9	20.9
01-01-2021 22:02:00	20.9	21.2	21.0	21.0
01-01-2021 23:02:00	20.9	20.9	21.0	20.9
24-02-2021 00:02:00	22.7	22.5	22.5	22.6
24-02-2021 01:02:00	22.7	22.5	22.4	22.5
24-02-2021 02:02:00	22.6	22.4	22.3	22.4
24-02-2021 03:02:00	22.5	22.3	22.1	22.3
24-02-2021 04:02:00	22.4	22.3	22.1	22.3
24-02-2021 05:02:00	22.3	22.2	22.0	22.2
24-02-2021 06:02:00	22.3	22.1	21.9	22.1
24-02-2021 07:02:00	22.2	22.1	21.8	22.0
24-02-2021 08:02:00	22.2	22.2	21.8	22.1
24-02-2021 09:02:00	22.7	22.9	22.5	22.7
24-02-2021 10:02:00	23.2	23.6	23.2	23.3
24-02-2021 11:02:00	23.4	23.9	23.5	23.6
24-02-2021 12:02:00	23.2	22.6	21.9	22.6
24-02-2021 13:02:00	23.2	23.1	23.5	23.3
24-02-2021 14:02:00	23.1	22.5	23.4	23.0
24-02-2021 15:02:00	23.4	22.6	23.2	23.1
24-02-2021 16:02:00	23.7	22.6	23.3	23.2
24-02-2021 17:02:00	23.5	22.2	23.2	23.0
24-02-2021 18:02:00	23.5	21.8	23.2	22.8
24-02-2021 19:02:00	23.4	21.5	23.2	22.7
24-02-2021 20:02:00	23.2	21.7	23.2	22.7
24-02-2021 21:02:00	23.1	21.6	23.1	22.6
24-02-2021 22:02:00	23.0	21.6	23.0	22.5
24-02-2021 23:02:00	22.8	21.5	23.0	22.4
23-02-2021 00:02:00	22.2	22.2	21.9	22.1
23-02-2021 01:02:00	22.2	22.1	21.8	22.0
23-02-2021 02:02:00	22.1	22.0	21.7	21.9
23-02-2021 03:02:00	22.1	22.0	21.7	21.9
23-02-2021 04:02:00	22.1	22.0	21.7	21.9
23-02-2021 05:02:00	22.1	22.0	21.7	21.9
23-02-2021 06:02:00	22.1	22.0	21.7	21.9
23-02-2021 07:02:00	22.1	22.0	21.7	21.9
23-02-2021 08:02:00	22.1	22.0	21.8	22.0
23-02-2021 09:02:00	22.3	22.1	22.1	22.2
23-02-2021 10:02:00	22.0	22.9	22.1	22.3
23-02-2021 11:02:00	21.9	22.9	22.2	22.3
23-02-2021 12:02:00	22.1	23.0	22.4	22.5
23-02-2021 13:02:00	22.1	23.2	21.6	22.3
23-02-2021 14:02:00	22.5	23.3	22.8	22.9
23-02-2021 15:02:00	22.6	23.5	23.1	23.1
23-02-2021 16:02:00	22.9	23.7	23.3	23.3
23-02-2021 17:02:00	23.5	23.1	23.3	23.3
23-02-2021 18:02:00	23.6	23.2	23.1	23.3
23-02-2021 19:02:00	23.3	23.0	22.8	23.0
23-02-2021 20:02:00	23.1	23.1	23.0	23.1
23-02-2021 21:02:00	23.0	22.9	22.9	22.9
23-02-2021 22:02:00	22.9	22.8	22.9	22.9
23-02-2021 23:02:00	22.9	22.6	22.8	22.8
Average per sensor	22.0	22.0	21.9	



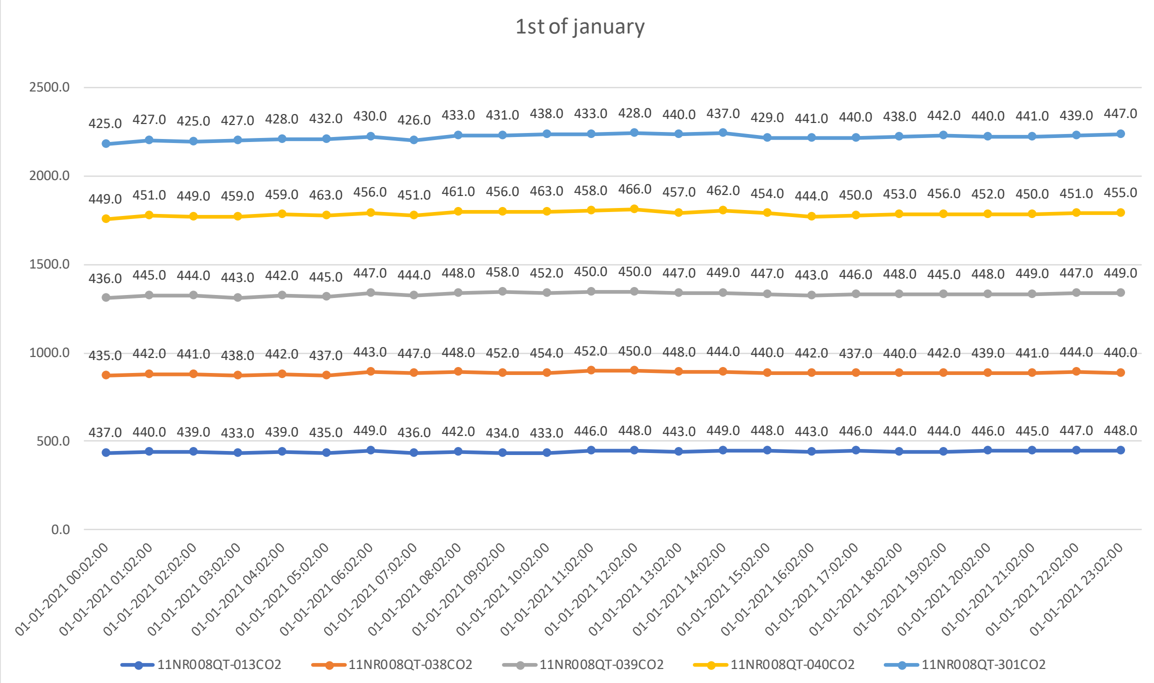


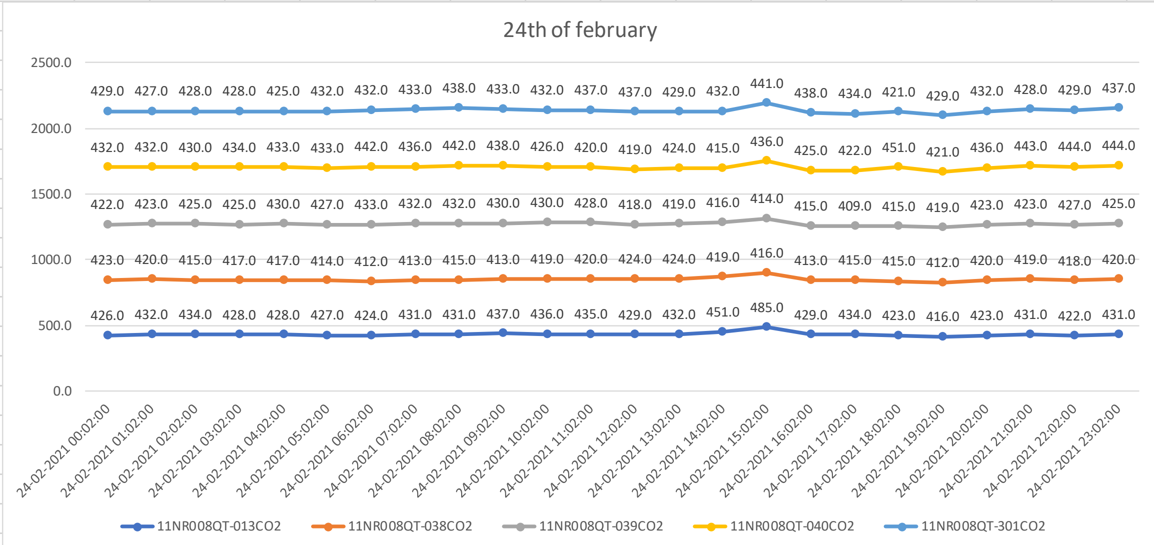


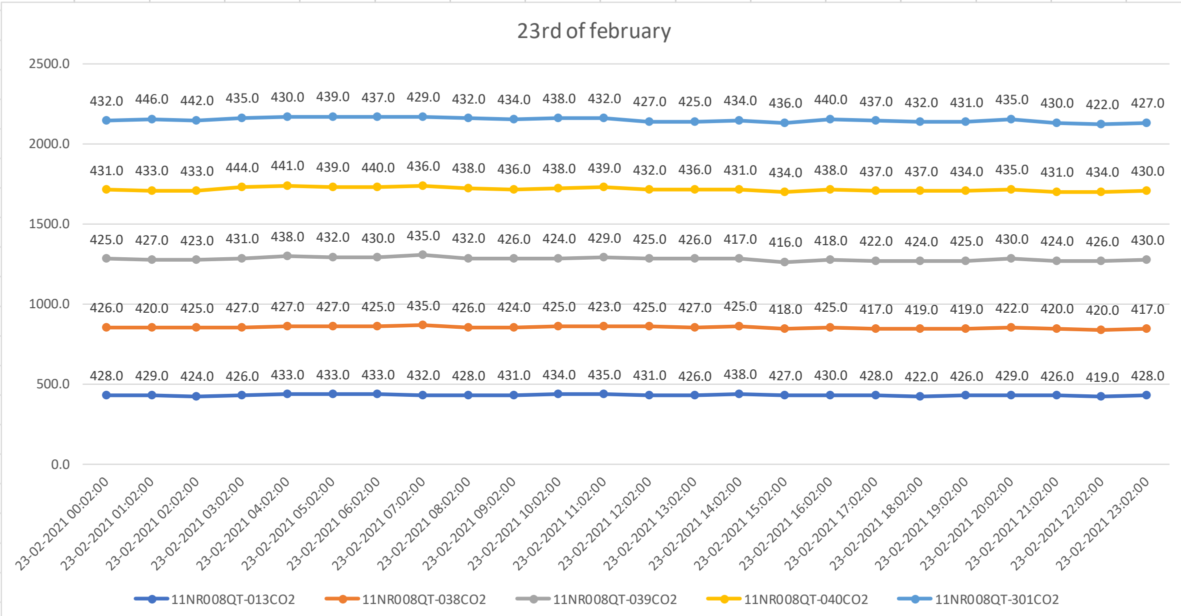


CO2

Timestamp	11NR008 QT-013CO2	11NR008 QT-038CO2	11NR008 QT-039CO2	11NR008 QT-040CO2	11NR008 QT-301CO2	Average all sensors per timestamp
01-01-2021 00:02:00	437.0	435.0	436.0	449.0	425.0	**436.4**
01-01-2021 01:02:00	440.0	442.0	445.0	451.0	427.0	**441.0**
01-01-2021 02:02:00	439.0	441.0	444.0	449.0	425.0	**439.6**
01-01-2021 03:02:00	433.0	438.0	443.0	459.0	427.0	**440.0**
01-01-2021 04:02:00	439.0	442.0	442.0	459.0	428.0	**442.0**
01-01-2021 05:02:00	435.0	437.0	445.0	463.0	432.0	**442.4**
01-01-2021 06:02:00	449.0	443.0	447.0	456.0	430.0	**445.0**
01-01-2021 07:02:00	436.0	447.0	444.0	451.0	426.0	**440.8**
01-01-2021 08:02:00	442.0	448.0	448.0	461.0	433.0	**446.4**
01-01-2021 09:02:00	434.0	452.0	458.0	456.0	431.0	**446.2**
01-01-2021 10:02:00	433.0	454.0	452.0	463.0	438.0	**448.0**
01-01-2021 11:02:00	446.0	452.0	450.0	458.0	433.0	**447.8**
01-01-2021 12:02:00	448.0	450.0	450.0	466.0	428.0	**448.4**
01-01-2021 13:02:00	443.0	448.0	447.0	457.0	440.0	**447.0**
01-01-2021 14:02:00	449.0	444.0	449.0	462.0	437.0	**448.2**
01-01-2021 15:02:00	448.0	440.0	447.0	454.0	429.0	**443.6**
01-01-2021 16:02:00	443.0	442.0	443.0	444.0	441.0	**442.6**
01-01-2021 17:02:00	446.0	437.0	446.0	450.0	440.0	**443.8**
01-01-2021 18:02:00	444.0	440.0	448.0	453.0	438.0	**444.6**
01-01-2021 19:02:00	444.0	442.0	445.0	456.0	442.0	**445.8**
01-01-2021 20:02:00	446.0	439.0	448.0	452.0	440.0	**445.0**
01-01-2021 21:02:00	445.0	441.0	449.0	450.0	441.0	**445.2**
01-01-2021 22:02:00	447.0	444.0	447.0	451.0	439.0	**445.6**
01-01-2021 23:02:00	448.0	440.0	449.0	455.0	447.0	**447.8**
24-02-2021 00:02:00	426.0	423.0	422.0	432.0	429.0	**426.4**
24-02-2021 01:02:00	432.0	420.0	423.0	432.0	427.0	**426.8**
24-02-2021 02:02:00	434.0	415.0	425.0	430.0	428.0	**426.4**
24-02-2021 03:02:00	428.0	417.0	425.0	434.0	428.0	**426.4**
24-02-2021 04:02:00	428.0	417.0	430.0	433.0	425.0	**426.6**
24-02-2021 05:02:00	427.0	414.0	427.0	433.0	432.0	**426.6**
24-02-2021 06:02:00	424.0	412.0	433.0	442.0	432.0	**428.6**
24-02-2021 07:02:00	431.0	413.0	432.0	436.0	433.0	**429.0**
24-02-2021 08:02:00	431.0	415.0	432.0	442.0	438.0	**431.6**
24-02-2021 09:02:00	437.0	413.0	430.0	438.0	433.0	**430.2**
24-02-2021 10:02:00	436.0	419.0	430.0	426.0	432.0	**428.6**
24-02-2021 11:02:00	435.0	420.0	428.0	420.0	437.0	**428.0**
24-02-2021 12:02:00	429.0	424.0	418.0	419.0	437.0	**425.4**
24-02-2021 13:02:00	432.0	424.0	419.0	424.0	429.0	**425.6**
24-02-2021 14:02:00	451.0	419.0	416.0	415.0	432.0	**426.6**
24-02-2021 15:02:00	485.0	416.0	414.0	436.0	441.0	**438.4**
24-02-2021 16:02:00	429.0	413.0	415.0	425.0	438.0	**424.0**
24-02-2021 17:02:00	434.0	415.0	409.0	422.0	434.0	**422.8**
24-02-2021 18:02:00	423.0	415.0	415.0	451.0	421.0	**425.0**
24-02-2021 19:02:00	416.0	412.0	419.0	421.0	429.0	**419.4**
24-02-2021 20:02:00	423.0	420.0	423.0	436.0	432.0	**426.8**
24-02-2021 21:02:00	431.0	419.0	423.0	443.0	428.0	**428.8**
24-02-2021 22:02:00	422.0	418.0	427.0	444.0	429.0	**428.0**
24-02-2021 23:02:00	431.0	420.0	425.0	444.0	437.0	**431.4**
23-02-2021 00:02:00	428.0	426.0	425.0	431.0	432.0	**428.4**
23-02-2021 01:02:00	429.0	420.0	427.0	433.0	446.0	**431.0**
23-02-2021 02:02:00	424.0	425.0	423.0	433.0	442.0	**429.4**
23-02-2021 03:02:00	426.0	427.0	431.0	444.0	435.0	**432.6**
23-02-2021 04:02:00	433.0	427.0	438.0	441.0	430.0	**433.8**
23-02-2021 05:02:00	433.0	427.0	432.0	439.0	439.0	**434.0**
23-02-2021 06:02:00	433.0	425.0	430.0	440.0	437.0	**433.0**
23-02-2021 07:02:00	432.0	435.0	435.0	436.0	429.0	**433.4**
23-02-2021 08:02:00	428.0	426.0	432.0	438.0	432.0	**431.2**
23-02-2021 09:02:00	431.0	424.0	426.0	436.0	434.0	**430.2**
23-02-2021 10:02:00	434.0	425.0	424.0	438.0	438.0	**431.8**
23-02-2021 11:02:00	435.0	423.0	429.0	439.0	432.0	**431.6**
23-02-2021 12:02:00	431.0	425.0	425.0	432.0	427.0	**428.0**
23-02-2021 13:02:00	426.0	427.0	426.0	436.0	425.0	**428.0**
23-02-2021 14:02:00	438.0	425.0	417.0	431.0	434.0	**429.0**
23-02-2021 15:02:00	427.0	418.0	416.0	434.0	436.0	**426.2**
23-02-2021 16:02:00	430.0	425.0	418.0	438.0	440.0	**430.2**
23-02-2021 17:02:00	428.0	417.0	422.0	437.0	437.0	**428.2**
23-02-2021 18:02:00	422.0	419.0	424.0	437.0	432.0	**426.8**
23-02-2021 19:02:00	426.0	419.0	425.0	434.0	431.0	**427.0**
23-02-2021 20:02:00	429.0	422.0	430.0	435.0	435.0	**430.2**
23-02-2021 21:02:00	426.0	420.0	424.0	431.0	430.0	**426.2**
23-02-2021 22:02:00	419.0	420.0	426.0	434.0	422.0	**424.2**
23-02-2021 23:02:00	428.0	417.0	430.0	430.0	427.0	**426.4**
Average per sensor	**434.5**	**428.0**	**432.2**	**441.1**	**433.1**	







FLOW

Timestamp	11NR008 FT-013FLW	11NR008 FT-038FLW	11NR008 FT-039FLW	11NR008 FT-040FLW	11NR008 FT-301FLW	Average all sensors per timestamp
01-01-2021 00:02:00	0.0	0.0	0.0	28.0	0.0	**5.6**
01-01-2021 01:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 02:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 03:02:00	0.0	0.0	0.0	28.0	0.0	**5.6**
01-01-2021 04:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 05:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 06:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 07:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 08:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 09:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 10:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 11:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 12:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 13:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 14:02:00	0.0	0.0	0.0	28.0	0.0	**5.6**
01-01-2021 15:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 16:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 17:02:00	0.0	0.0	0.0	28.0	0.0	**5.6**
01-01-2021 18:02:00	0.0	0.0	0.0	28.0	0.0	**5.6**
01-01-2021 19:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 20:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 21:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 22:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
01-01-2021 23:02:00	0.0	0.0	0.0	30.0	0.0	**6.0**
24-02-2021 00:02:00	64.0	115.0	84.0	115.0	111.0	**97.8**
24-02-2021 01:02:00	64.0	115.0	83.0	115.0	110.0	**97.4**
24-02-2021 02:02:00	64.0	114.0	83.0	114.0	111.0	**97.2**
24-02-2021 03:02:00	64.0	116.0	84.0	117.0	110.0	**98.2**
24-02-2021 04:02:00	64.0	115.0	83.0	114.0	110.0	**97.2**
24-02-2021 05:02:00	64.0	114.0	84.0	114.0	110.0	**97.2**
24-02-2021 06:02:00	64.0	114.0	82.0	114.0	111.0	**97.0**
24-02-2021 07:02:00	66.0	119.0	83.0	119.0	113.0	**100.0**
24-02-2021 08:02:00	65.0	116.0	82.0	117.0	135.0	**103.0**
24-02-2021 09:02:00	66.0	115.0	81.0	116.0	135.0	**102.6**
24-02-2021 10:02:00	67.0	115.0	82.0	267.0	132.0	**132.6**
24-02-2021 11:02:00	64.0	114.0	80.0	113.0	136.0	**101.4**
24-02-2021 12:02:00	66.0	115.0	82.0	115.0	134.0	**102.4**
24-02-2021 13:02:00	66.0	116.0	82.0	115.0	136.0	**103.0**
24-02-2021 14:02:00	66.0	117.0	84.0	115.0	138.0	**104.0**
24-02-2021 15:02:00	67.0	269.0	80.0	115.0	136.0	**133.4**
24-02-2021 16:02:00	67.0	118.0	82.0	269.0	135.0	**134.2**
24-02-2021 17:02:00	65.0	118.0	80.0	270.0	136.0	**133.8**
24-02-2021 18:02:00	68.0	118.0	80.0	269.0	136.0	**134.2**
24-02-2021 19:02:00	67.0	113.0	80.0	113.0	134.0	**101.4**
24-02-2021 20:02:00	65.0	116.0	82.0	113.0	111.0	**97.4**
24-02-2021 21:02:00	64.0	115.0	82.0	116.0	110.0	**97.4**
24-02-2021 22:02:00	64.0	115.0	82.0	117.0	111.0	**97.8**
24-02-2021 23:02:00	65.0	115.0	82.0	116.0	113.0	**98.2**
23-02-2021 00:02:00	64.0	118.0	84.0	115.0	110.0	**98.2**
23-02-2021 01:02:00	64.0	117.0	83.0	115.0	110.0	**97.8**
23-02-2021 02:02:00	64.0	118.0	84.0	115.0	110.0	**98.2**
23-02-2021 03:02:00	64.0	116.0	82.0	115.0	111.0	**97.6**
23-02-2021 04:02:00	64.0	118.0	83.0	115.0	110.0	**98.0**
23-02-2021 05:02:00	64.0	117.0	83.0	115.0	111.0	**98.0**
23-02-2021 06:02:00	64.0	116.0	82.0	264.0	111.0	**127.4**
23-02-2021 07:02:00	64.0	116.0	82.0	274.0	113.0	**129.8**
23-02-2021 08:02:00	67.0	117.0	84.0	268.0	134.0	**134.0**
23-02-2021 09:02:00	65.0	115.0	82.0	117.0	136.0	**103.0**
23-02-2021 10:02:00	67.0	114.0	82.0	116.0	135.0	**102.8**
23-02-2021 11:02:00	64.0	114.0	82.0	116.0	134.0	**102.0**
23-02-2021 12:02:00	66.0	114.0	82.0	116.0	139.0	**103.4**
23-02-2021 13:02:00	68.0	113.0	81.0	115.0	136.0	**102.6**
23-02-2021 14:02:00	67.0	117.0	82.0	115.0	139.0	**104.0**
23-02-2021 15:02:00	64.0	118.0	82.0	116.0	136.0	**103.2**
23-02-2021 16:02:00	64.0	117.0	82.0	269.0	135.0	**133.4**
23-02-2021 17:02:00	64.0	116.0	83.0	115.0	135.0	**102.6**
23-02-2021 18:02:00	64.0	117.0	84.0	115.0	138.0	**103.6**
23-02-2021 19:02:00	64.0	115.0	82.0	114.0	136.0	**102.2**
23-02-2021 20:02:00	64.0	114.0	83.0	118.0	110.0	**97.8**
23-02-2021 21:02:00	63.0	114.0	83.0	118.0	110.0	**97.6**
23-02-2021 22:02:00	63.0	114.0	82.0	117.0	110.0	**97.2**
23-02-2021 23:02:00	64.0	114.0	82.0	117.0	111.0	**97.6**
Average per sensor	**43.3**	**79.3**	**54.9**	**103.9**	**82.1**