Interim Work Report

24-12-2020

BIM-based permit procedure

Development of a software solution for the BIM-based authorization procedure

Reach u

European Union
European Structural and Investment Funds

Investing in your future

Republic of Estonia
Ministry of Economic Affairs and Communications

FUTURE INSIGHT

Index

1. Introduction

This document is the first report describing the results and activities executed within the first phase of the project to develop a software solution for the BIM based permitting procedure for the Ministry of Economic Affairs and Communications of Estonia (MKM).

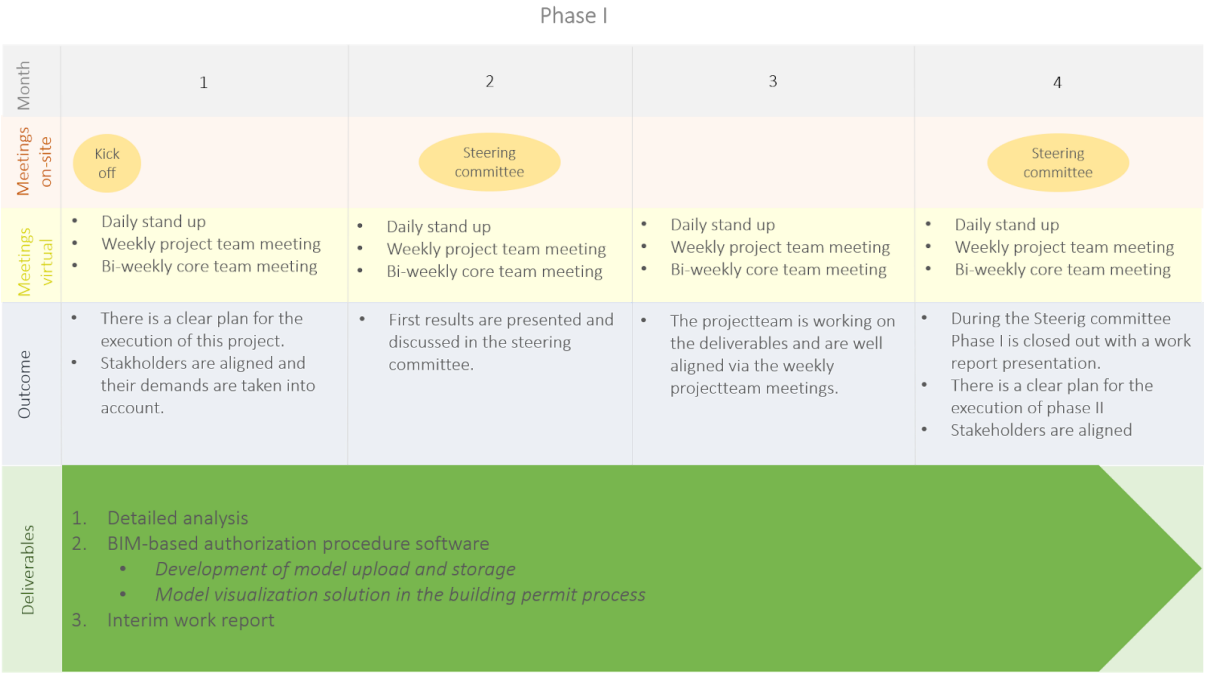
In 2019 we, as Future Insight already had the honour to work on the Proof of Concept for this project. Building the first outline for a BIM based process for building permits in Estonia. A very innovative and challenging assignment, but also an assignment that fitted in perfectly with our vision and experience and which gave us the opportunity to bring this vision into practice. An opportunity to show what already is possible with BIM models and which impact this can have related to efficiency and effectiveness within the daily work environment. And not without result, the Proof of Concept was also noticed internationally and even rewarded with a 'Special Mention' during the international Buildings Smart Awards in October 2020.

We are pleased to have also won the tender for the actual first implementation of the service, for which we have entered into a close partnership with Reach-U. They also have extensive experience in carrying out innovative projects and were, for example, already involved in the development of the 3D Digital Twin proof of concept, on behalf of the ministry. Together we think we have all the necessary (local) knowledge and experience available to bring this challenging project to a successful conclusion and be one of the first countries in the world to start using a truly working online BIM-based permit checking service.

You are now reading the interim work report belonging to the first phase of this project. The goal of this first phase was to analyse all the information that is in place. Regarding the proposed checks that should be executed, regarding requirements and classifications. Next to the analysis it was also the goal to get a first common understanding on the architectural setup of the software solution, the user interface and how certain functionalities could work. This all to get a very solid common ground and understanding what will be developed in the second phase of this project. What is feasible and what is not (yet). Hopefully we grab your attention and you feel confident in us setting the next steps in this fantastic project.

2. Activities

Building an innovative project like this is very challenging. Moreover, there are all kinds of requirements and expectations, on the one hand purely from the tender but on the other hand simply from the user's needs. In order to clarify this, an extensive analysis was performed in this first phase, in accordance with the approach in the diagram below. Ultimately, there was a weekly consultation with relevant employees of the ministry, in which questions about the analysis and results were discussed.



At the beginning of the project, work was still somewhat organized in two separate paths. An Estonian for software development and an English one for the analysis of the checks. This did not benefit the integrality and the overview of the project, so we decided to merge them.

During the analysis we also came to the conclusion that the possibilities and ideas are endless on the one hand. There are a lot of details making it nearly impossible to analyse them all. While on the other hand there are still a lot of dependencies and unknowns, which makes it impossible to get all the details completely clear already. It is not possible to implement all these ideas within this single project, so choices will have to be made along the way.

Not all details are clear yet, but sufficient insight has been gained to start the second phase with confidence. In this phase we will, according to the method of agile development, get started with the actual development of the service while learning and deciding together. We think this way of working should be the basis for the next phase as well, we elaborate further on this in chapter 5.

3. Detailed analysis

The main goal of this first phase is to be well prepared for the actual development in phase 2. As requested in the assignment, a first detailed analysis is performed on the 64 proposed checks, also in relation to the BIM requirements and the CCI classification. In addition, we have started the analysis of the user flow. This was originally planned for phase 2 but we see an important impact of this user flow on the requirements for the technical infrastructure. In order to get started as smoothly as possible with the development sprints in phase 2, we wanted to gain a good first picture of the user flows already by the end of phase 1.

In this chapter we show you our results of the initial analyzes of the different topics. Given the innovative nature of the project, these are certainly not set in stone and will be further developed in phase 2.

3.1 Analysis of the 64 proposed checks

Included with the initial assignment there was a list of 64 checks (Annex 1A) that the ministry would like to see performed automatically within this project. Together with the ministry we performed a detailed analysis of these checks. We looked at the background, the technical feasibility and usability. Based on this, we have indicated whether or not we want to or can develop a check.

Instead of completing Annex 1A, as indicated in the assignment, we have opted in mutual consultation to draw up a separate 'living' document for the further elaboration of the checks. It can be found in the shared Google Drive environment via: https://docs.google.com/document/d/1x1b0jlU8VFR7g2r95NZ7kafT3C50bP-Qw0euXmm2D8U/edit?usp=sharing. A current version of the document is attached as Annex 3.
This gave us more possibilities to properly describe the checks, while of course still trying to fulfill the information needs from Annex 1A as well. During the second phase of this project, this document will be filled further check by check, as details become more and more known. The table below shows which elements in the end will be described per check

3.1: Name of the check

Classification (Low, Medium, High)
This gives a short indication about the feasibility and usefulness of the check ranging from low, medium to high.

For feasibility, we look at both the complexity of the technology to be developed as the additional requirements that should be imposed on the BIM models to be supplied and how common these are. By usefulness we mean how often a check could be used. Some checks are quite general and can therefore be used very often, while others are very specific and will be used less often.

When both values are high it is probably a good idea to develop the check, whereas if they are both low it might be wise to reconsider.

Detailed description:
This field should give a detailed description of the check to be executed. This often needs to be more specific than is now indicated in the table. A check to be carried out by the computer cannot make an assessment itself, but can only check predefined conditions. These must therefore be very clear.

Technical approach:
The technical approach of the check will be elaborated here. It will be indicated which functions from chapter 2 will be used and which values will be used to perform the check.

A short example (Check 1) looks like this:
When {2.2} == ONEOF(LIVING | STUDY | BEDROOM) (applicability)
Then WARNING IF {2.3} < 8m2 (requirement)

Reliability:
The reliability of the results will depend to a greater or lesser extent on, for example, the quality and completeness of the supplied BIM models and other data sources. These kinds of dependencies will be indicated here.

Necessary CCI classifications	**BIM requirements**	**IDS checks:**
When certain CCI classifications are necessary for a proper execution of the check, they will be described here.	If other BIM requirements are necessary for a proper execution of the check, they will be described here.	When requirements are set for the data that can already be checked by means of an IDS check, this will be described here.

Is the check an error or warning?
If this check fails is it blocking (Red) or just annoying (Orange)?
The reliability of the check also plays a role in this.

Will this check be executed?
Will the check be implemented or not? Yes will be green, No will be red

Next to these tables the document will also elaborate on all functions (to be developed) that are needed to perform the checks. A function can be reused in several checks and can therefore also be used to define new checks in the future.

To get a better overview of the feasibility and approach of all checks, an outline has been created in which the most important points are summarized. This can be found as well in the shared Google Drive environment via: https://docs.google.com/spreadsheets/d/1LH-bmNbp1TE-BB6nDo978h3zsZRqIsPJScMLtwIkwN8/edit?usp=sharing. The latest version of this table is included in Annex 1.

3.1.1 First analysis

The proposed checks are derived from various regulations. Some of these are more general, others quite specific. Below is a brief overview of these regulations with the link to the original legislation. The fact that these regulations are the basis, but can also change, indicates that the checks in the BIM service must be easy to adjust.

- M85: Housing requirements https://www.riigiteataja.ee/akt/109072020017
- M97: Requirements for construction projects (https://www.riigiteataja.ee/akt/118072015007)
- M28: Special needs of people with disabilities (https://www.riigiteataja.ee/akt/131052018055)
- EhS Design Conditions https://www.riigiteataja.ee/akt/130102020006
- M62: Requirements for construction project expertise (https://www.riigiteataja.ee/akt/109062015025)
- M17 Fire safety requirements for buildings and requirements for fire water supply (https://www.riigiteataja.ee/akt/130112018012)

A first thing to notice is that it must be determined whether certain rules/checks apply to a design. Additional information needs to be supplied for those specific checks, Either in the BIM model or via a separate entry in the EHS to trigger the right checks. When this information should be provided via the BIM model, it needs to be specified in the BIM requirements or the CCI classification table. Our findings and recommendations related to this, are further elaborated in Paragraph 3.2 and 3.3.

Another point of attention is that several checks require external location information from, for example, zoning plans or detailed plans. For this we are dependent on the availability of this type of data and facilities.

In addition, we see that when the regulations become more specific, the additional information to be linked also becomes increasingly specific. For example the fire safety requirements, for which very specific information needs to be added in order for the

automatic checks to work. To this end, a careful assessment needs to be made as to whether this is feasible.

Finally, we see that some rules are quite general and easy to check, but others are very detailed and specific. The latter usually ensures that a lot of additional requirements are imposed on the BIM models to be supplied, whereby the added value and feasibility seems less. Moreover, the reliability of these checks is limited. This may have led to the advice not to develop these checks (yet).

3.1.2 Checks to be developed

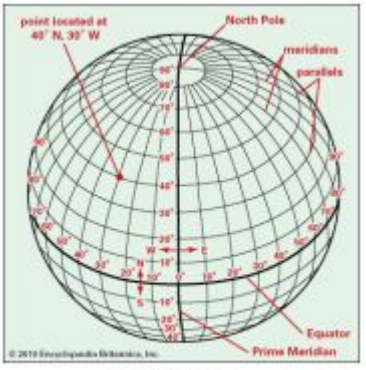
Based on the first extensive analysis, we now assume that at least a total of **32** checks can be properly implemented. For about **25** more we do see opportunities but at the same time there are also still a number of challenges in terms of technology or specifications. These are classified as 'maybe' meaning we still want to investigate their feasibility in phase 2. Note that 20 of these come from the fire safety checks, which is quite complex and which we really need to specify better first. Finally, we come to the conclusion that around **7** checks are currently not feasible for various reasons. Some are actually just not really checks but an introduction to the regulations, but for others we found the complexity of the check in combination with the extra requirements for the BIM models just too high relative to the usefulness. There are also a number of checks in the 'maybe' category where this consideration is involved. More details about the how and why off individual checks can be found in the extensive check table in Annex 1.

For the checks we assume that the various preconditions can be met, such as the BIM requirements and the availability of external information for the proper execution. During the second phase of the project, there may be some changes in these numbers, but this seems realistic to us based on current knowledge.

3.2 Common BIM requirements

A parallel project which is currently being executed is setting up the 'Common BIM requirements'. This is a very extensive document in which many requirements for BIM models are stated and which is still under development. For the analysis, we did not check whether the document is fully complete, but we looked more closely from the requirements of the checks to see whether these are present in the document. We have elaborated on the most important ones below.

- Geolocation
  A very important requirement is to make sure that the BIM designs are delivered including proper geolocation attached according to the guidelines defined by bSI. (https://www.buildingsmart.org/wp-content/uploads/2020/02/User-Guide-for-Geo-referencing-in-IFC-v2.0.pdf).
  Currently, this requirement is not extensively mentioned in the document, although it is a very important one. Multiple checks



Earth System

need this geolocation for proper operation and it is of course necessary for a good combination with the 3D Digital Twin. Usually this is currently not the case and different BIM tools have different ways storing and exporting this location. It is therefore important to clearly specify what is expected and to organize further explanations where possible. With the recent domain extensions to the IFC schema for infrastructure a lot of extra attention has been given to this topic and entities such as IfcMapConversion and IfcProjectedCRS have been devised to model associations to local datums, but implementation in software still needs to catch up.

- The whole building needs to be divided into 'IfcSpaces'
  Another very important requirement is that the entire model is divided into 'ifcSpaces'. This is necessary both for triggering the correct checks as well as for carrying them out. This principle has already been mentioned in chapter 5.2 of the BIM requirements, but the requirements for the exact name of the Spaces will go a lot further than currently stated. Also there will be additional values that must be added to the Spaces for performing the various checks. Think of the maximum number of people allowed for fire safety checks.

- The spaces need to be divided into 'IfcZones'
  In certain cases, the aforementioned Spaces must also be grouped in 'ifcZones'. This is especially important for triggering the correct checks (public use), but also for the implementation of a number of them. The ifcZones are not yet mentioned in the current version of the common BIM requirements and will therefore have to be added. The exact interpretation will be further discussed in phase 2 in combination with the Spaces.

- IfcSite needs to be included
  For several checks the combination of the design of the building with the direct surroundings is necessary. In addition, this is needed to assess the plan in its entirety in its environment. That is why it is necessary, in addition to the building, to also provide the design of the adjacent environment within the 'ifcSite'. This is already mentioned in chapter 5.2 of the BIM requirements. But here as well more detailed preconditions will arise in order to be able to perform various checks. Especially when it is necessary to check if the design fits well within the building plot and whether it fits well within the environment.

3.3 CCI Classification

As mentioned in the previous paragraph a fairly extensive structure of ifcZones and ifcSpaces needs to be set up. This should point out with which functions and, for example, the publicity of spaces, need to be labeled in order to trigger the right checks This has a lot of impact for the designers, but is crucial for a good understanding of the designs and the correct operation of the automatic checks. Although there are already a lot of spaces defined in the CCI classification table, not all necessary definitions are available and ifcZones are not available at all.

Since the CCI classification is an international standard it's not that easy to change or add classifications. We have agreed with the ministry during this first phase that we will take a pragmatic approach.. We will choose other available classifications as an alternative, without immediately adjusting the original classifications. Our proposal is to make a plan for this as soon as possible in phase 2, as this is a very important precondition for the execution of many checks.

3.4 Functional logic and GUI implementation

The purpose of the module (or micro-service) of BIM-based checks is to provide additional possibility to use BIM models as an input when applying for a building permit in the existing EHR environment. The module to be created must be integrated both functionally and visually into the existing environment, which means that the aim of this project is to change the existing operating logics as little as possible and rather to supplement the existing system with BIM-derived inputs and capabilities.

In our original proposal, we have provided a first sketch of the technical solution as we envision it. In order to get a better elaboration of this sketch, we started analyzing the user flows in this first analysis phase.
Phase 1 has been an important period for collecting information and getting a better understanding about the general business logic of EHR and how the new BIM-based module should logically fit into the existing building permit application workflow. Nevertheless, as especially processing of an application is a rather complex process, integration of some BIM-based input to the already existing flow needs to be specified in the beginning of phase 2.

As the development is agile, we have so far wireframed only the general UI views, details need to be discussed and agreed during Phase 2. As the BIM module must integrate seamlessly and logically to the current EHR permit application process, it is important to make sure that the cornerstones will be placed right. Discussions for finalising the analysis of applicant and processor workflows, designing UI views and describing all necessary connection points is planned for the beginning of phase 2. From the UX/UI point of view, as soon as the main workflows are agreed upon, the next step will be to add more functional details to the wireframes and from there move further to fully designed mock-ups.

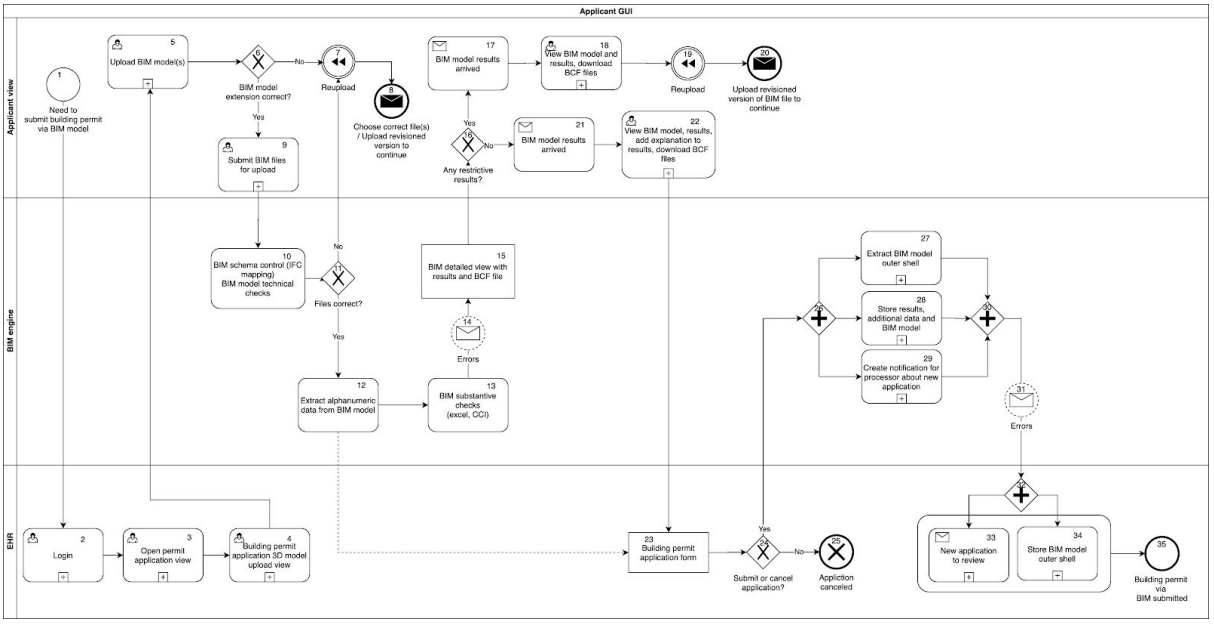
The following sections describe, in conceptual level, workflows, functional interactions and responsibilities between the BIM module and general EHR. On sketch level also different views of the GUI are visualized where relevant.

3.4.1 Permit applicant workflow

Currently, while applying for a building permit, users need to provide all the required information by filling forms and uploading several documents. As BIM models already contain a large part of the necessary information, they can significantly speed up the exchange of information required for the permit process. Keeping that in mind, the following core functionalities are important for the permit applicant in connection with BIM-module:

- upload files
- run automatic checks
- view inspection results
    - group results (by disciplines and by legislation)
    - filter results (by success level)
- view additional information about check results
    - expected result
    - actual result
- add comments to check results
- extraction of metadata from BIM model to be used as input for general application form
- link between inspection results and the corresponding elements in 3D-viewer
- possibility to use inspection results in external BIM software (BCF export)

The following diagram illustrates all the BIM-related steps involved in applying for a building permit.



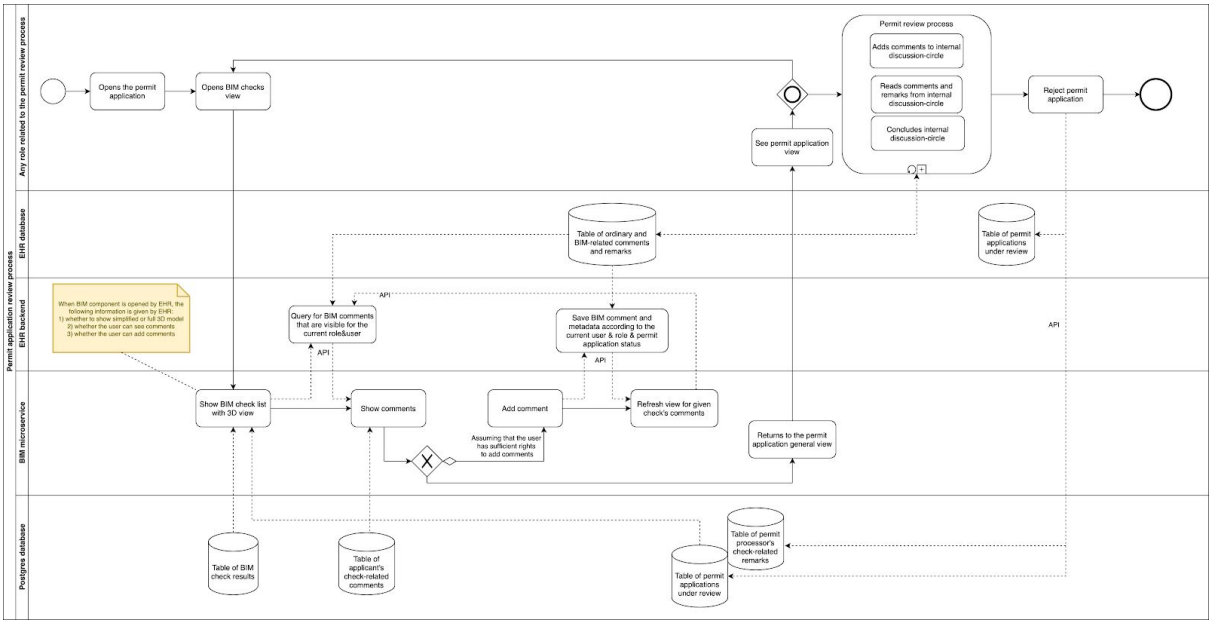
Permit applicant general flow diagram

3.4.2 Permit processor workflow

From the application processing point of view, an important goal is to simplify and speed up communication between the processing parties, using the results of BIM-based inspections. Depending on the area of responsibility of a particular party, it has the obligation to verify compliance with the requirements arising from the legislation of its field of competence.

For permit application processors the main functionalities are as follows

- view inspection results
    - group results (by legislation and by discipline)
    - filter results (by success level)
- view additional information about check results
    - expected result
    - actual result
- refer to a specific set of control results for another processing party
- add comments/remarks to check results
- link between inspection results and the corresponding elements in 3D-viewer
- possibility to use inspection results in external BIM software



Permit processor general flow diagram

3.4.3 Functional blocks of the BIM micro-service

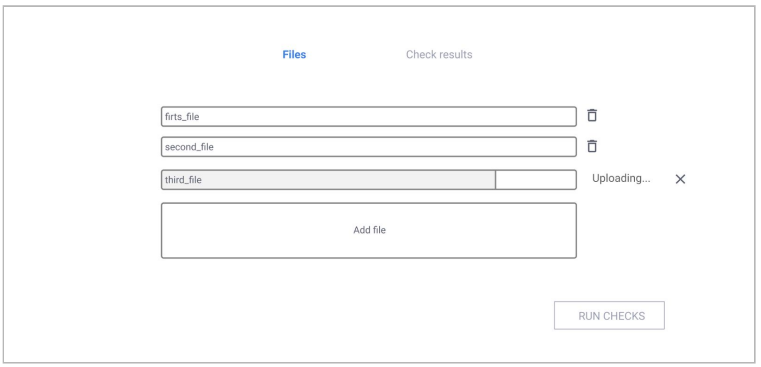
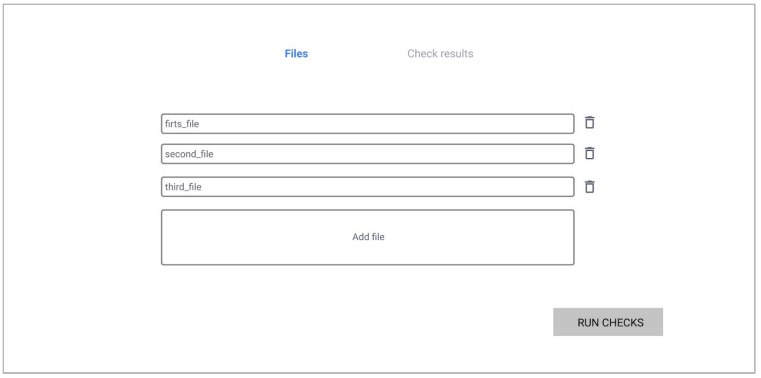
The section below describes the core functionalities that are included or related to the different views of the BIM module. Inputs, outputs and/or components required to implement the functionalities are also listed, and UI wireframes are added for better understanding of the workflow.

It is necessary for the applicant and the processor to compile views with slightly different possibilities, but the basic functionalities related to the BIM model and check results are still the same.

Login	
Role	Applicant, Processor, Manager
Responsible party	EHR
Service	KeyCloak (EHR)
Comment	Before reaching the BIM module, user has already logged in to the EHR environment, so BIM module itself does not have a "log in" button for user

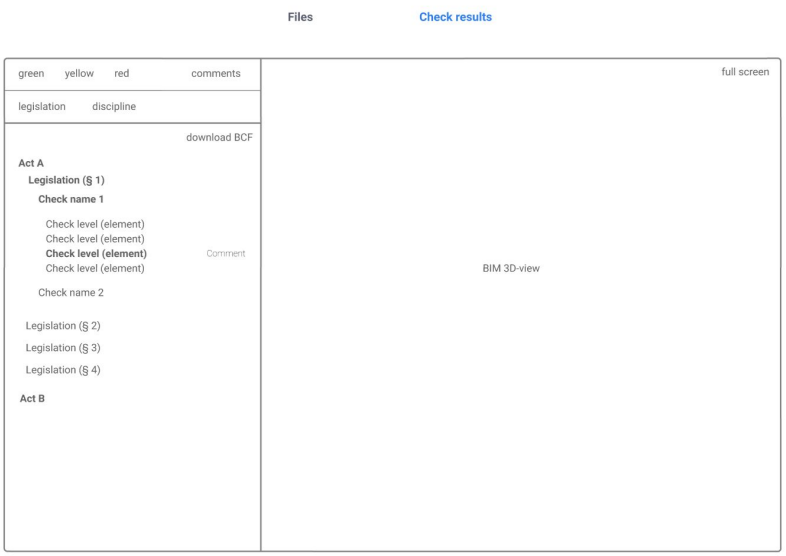
Check role	
Role	System
Responsible party	BIM micro-service
Service	KeyCloak (EHR)
Comment	System controls if the user is authorized to use the current setup of the BIM module. EHR KeyCloak solution will be used

Model upload (add/remove files)	
Role	Applicant
Responsible party	BIM micro-service
Service	BIM.Works SDK - bimworks/sdk/modeluploader/modeluploader
Wireframes	File upload process

	 Model is ready for checks 

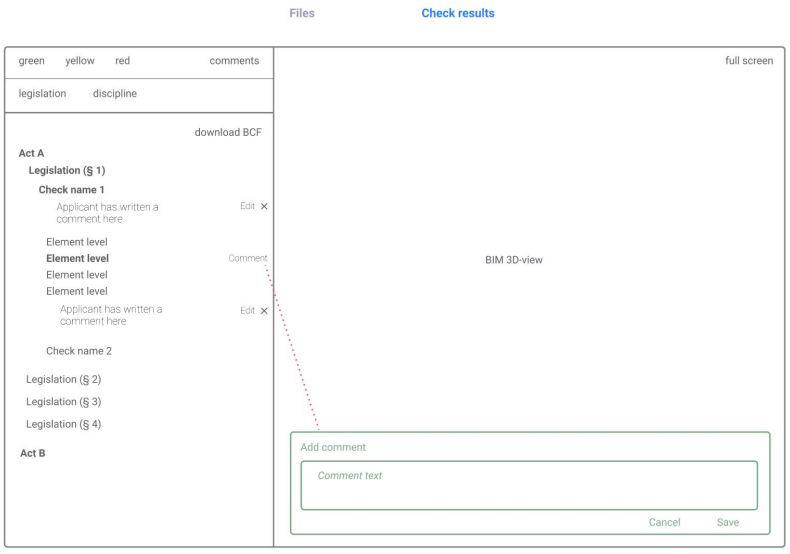
Run automated checks	
Role	Applicant
Responsible party	BIM micro-service
Service	BIM.Works
Comment	-

View check results	
Role	Applicant, Processor
Responsible party	BIM micro-service
Service	API from BIM.Works
Comment	Check results are listed and grouped by legislation or check type Filtering is available by success level

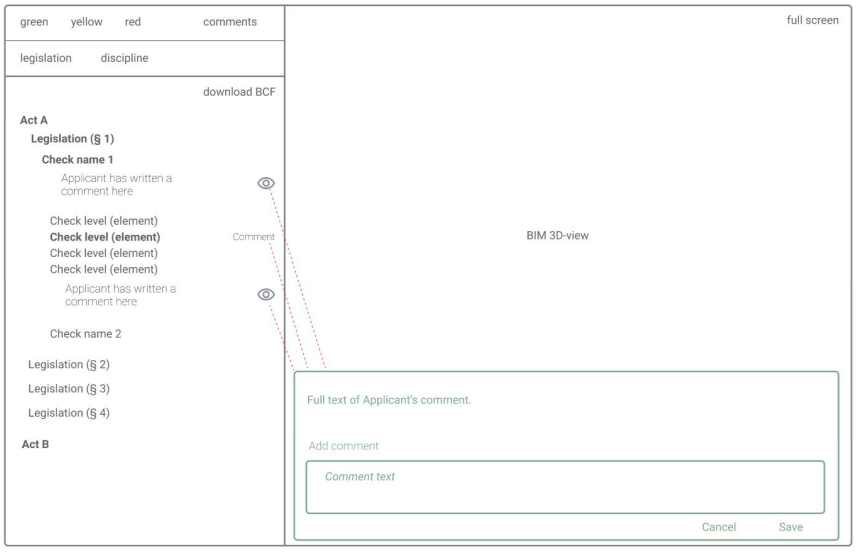
Wireframe	

Show 3D background data from Digital Twin	
Role	Applicant, Processor
Responsible party	BIM micro-service, Digital Twin
Input	API from Digital Twin
Comment	Background data is displayed in BIM Surfer together with the BIM model. Details of how to integrate the Digital Twin data is to be specified in phase 2

Interaction between check results and the corresponding 3D-elements	
Role	Applicant, Processor
Responsible party	BIM micro-service
Service	BIM.works SDKs component RecipeExecutor
Comment	If user clicks on a check result, the corresponding element is shown and highlighted in 3D view

Add comments to check results	
Role	Applicant
Responsible party	To be discussed and specified in the beginning of phase 2
Service	To be discussed and specified in the beginning of phase 2
Comment	User can add comments to any of the check results before submitting the permit application. Our current position is that applicant comments could be handled by the BIM module. In the wireframe below, the commenting block is shown in green. Final solution needs to be specified in the beginning of phase 2.
Wireframe	

Add comments and remarks regarding BIM-based check results	
Role	Processor
Responsible party	To be discussed and specified in the beginning of phase 2
Service	To be discussed and specified in the beginning of phase 2
Comment	User can add comments/remarks related to BIM check results and elements. Our current proposal is that processors' comments and remarks should be handled the same way it is organised now, but extra check-related link could be added to each comment/remark from the BIM micro-service, to maintain the necessary connection between a comment/remark and the related element. In the wireframe below, the commenting block is shown in green.

	Final solution needs to be specified in the beginning of phase 2.
Wireframe	

View comments	
Role	Applicant, Processor
Responsible party	To be discussed and specified in the beginning of phase 2
Service	To be discussed and specified in the beginning of phase 2
Comment	User can see the comments added to check results

View processors' BIM-related comments/remarks as a list	
Role	Processor
Responsible party	To be discussed and specified in the beginning of phase 2
Service	To be discussed and specified in the beginning of phase 2
Comment	User must see a complete list of BIM-related comments/remarks close to the BIM viewer and have a possibility for quick visual connection between a comment and its related 3D-element. Our current proposal is that the list of comments/remarks should be generated the same way it is done currently. The extra link to 3D-element that was added when comment was created, allows to quickly open the item in 3D-view. In the wireframe below, the list of comments is shown in green.

	Final solution needs to be specified in the beginning of phase 2.
Wireframe	

Download BCF	
Role	Applicant, Processor
Responsible party	BIM micro-service
Service	API from BIM.Works
Comment	User can download BCF that contains all the check results generated and remarks/comments added during the current application processing session

Extracted data from BIM model	
Role	System
Responsible party	BIM micro-service
Service	API from BIM.Works
Comment	Alphanumeric data will be automatically extracted from uploaded BIM model and made available for further use anywhere in the EHR general permit application process (e.g. forms etc)

Generate simplified 3D-model for 3D Digital Twin	
Role	System
Responsible party	BIM micro-service
Service	API from BIM.Works
Comment	From each BIM model that is uploaded to the system, a simplified model is generated that can be used in 3D Digital Twin

Submit building permit application	
Role	Applicant
Responsible party	EHR
Comment	Process of submitting the application remains as it currently is as is not affected by the BIM micro-service

3.5 Software architecture

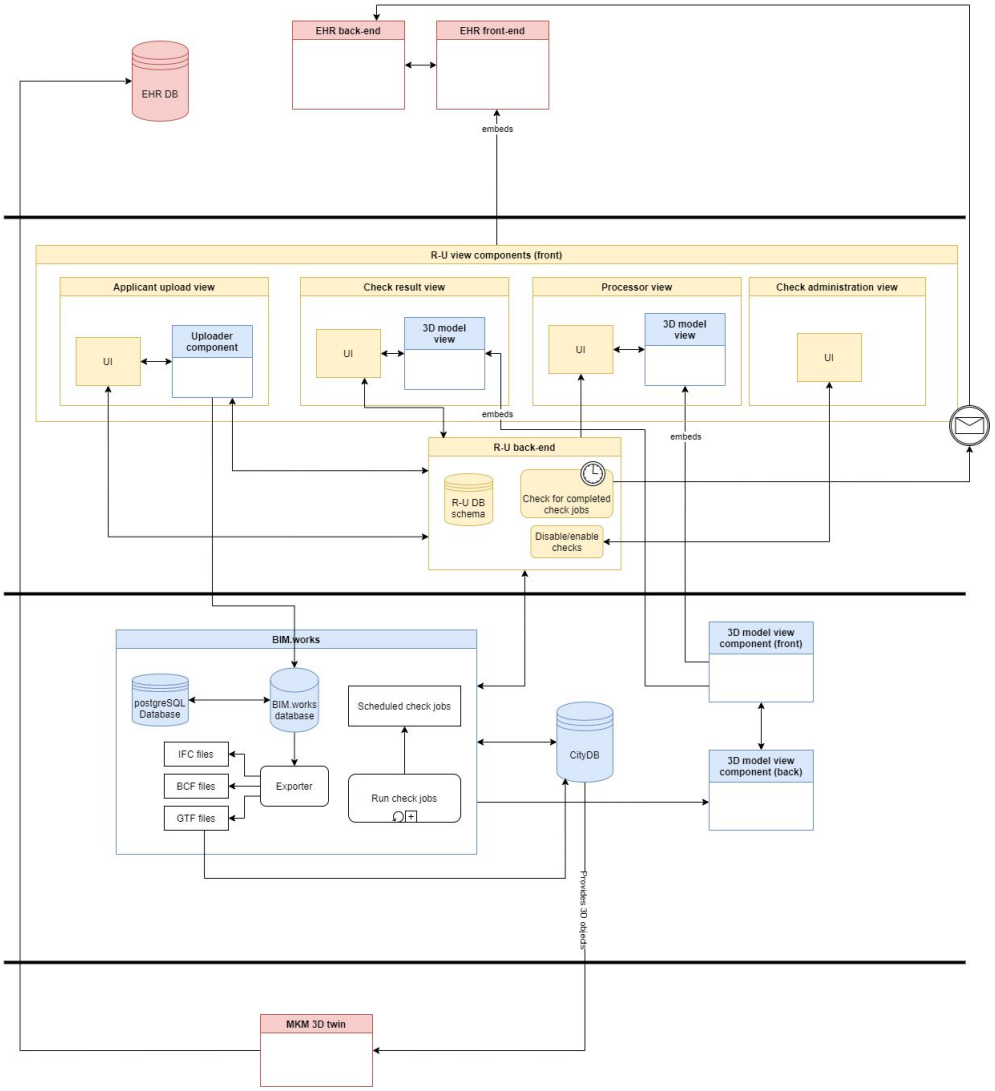
In our original proposal, we have provided a first sketch of the technical solution as we envision it. In order to get a better elaboration of this sketch, we started analyzing the user flows in this first analysis phase. It has been very important to collect information and get a better understanding about the general structure of EHR and how the new BIM-based module should logically fit into the existing building permit application workflow.

Phase 1 has been very important for collecting information and getting a better understanding about the general structure of EHR and how the new BIM-based module should logically fit into the existing building permit application workflow. Based on several meetings with the customer and exploration of background materials provided by MKM, .

3.5.1 General structure of the system

In general, the software for BIM-based permit application is planned to be built as a separate micro-service or module that can be integrated into the existing EHR environment. In order to function, the BIM module, on the one hand, takes in existing external services (eg authorization, 3D background data etc.) and, on the other hand, outputs data that can be used to improve the existing general flow of EHR building permit application.

The following diagram describes interaction of system parts, and colors indicate which party is responsible for the particular section (blue - Future Insight, orange - Reach-U, red - MKM).



General system diagram

3.5.2 External connections

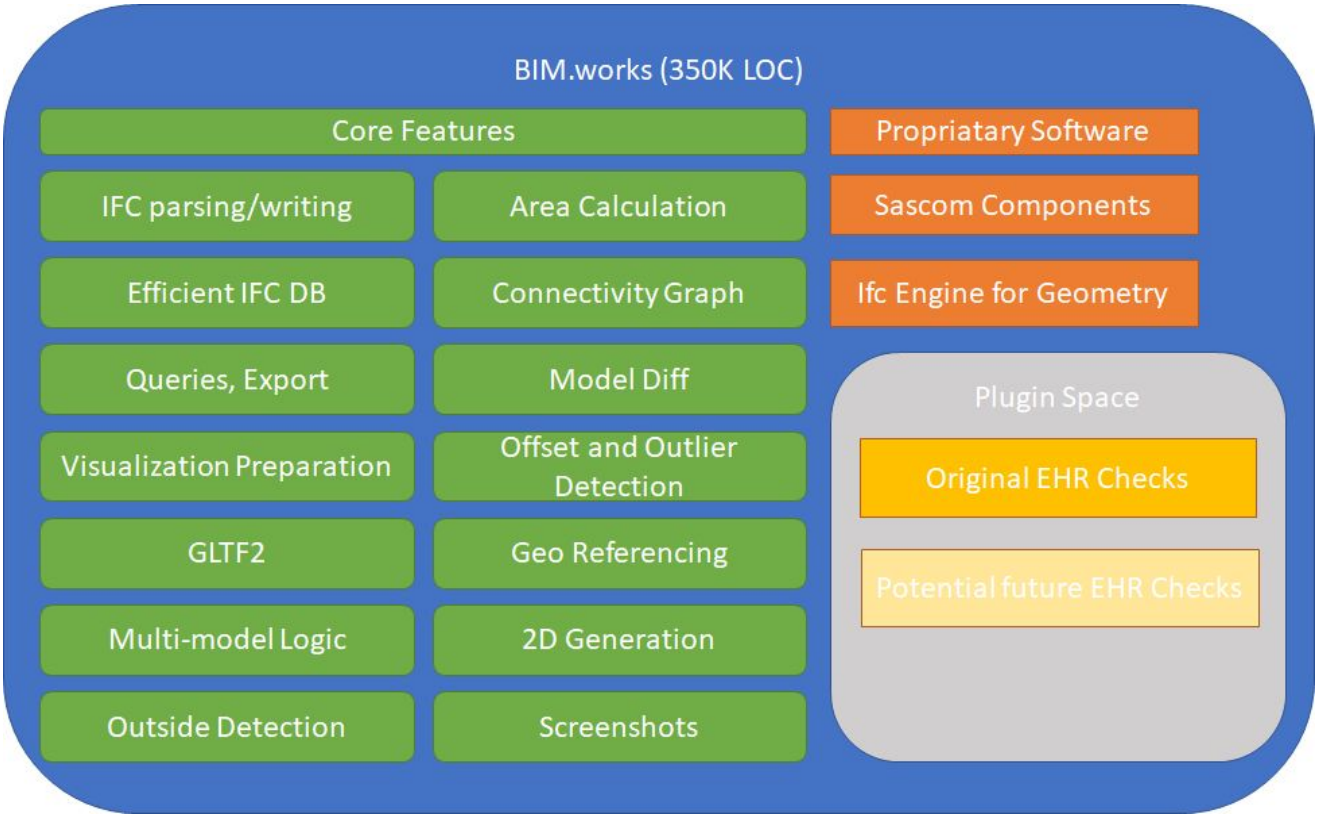
- BIM module as a micro-service component will be embedded into general EHR.
- KeyCloak will be used for authentication and authorization.
    - https://git.mkm.ee/ehr/ehr-k8s-pipeline/keycloak/-/tree/master_ehr
    - https://git.mkm.ee/ehr/ehr-k8s-pipeline/ehr-user-api
- Background data from Digital Twin via API to BIM module (details will be specified during phase 2). Input from https://devkluster.ehr.ee/ui/3dtwin/v1/
    - Open question: if there is an existing building in the same location, will 3D twin provide a specific API call (which receives the building ID as a parameter) which provides the background data without the existing building?
- BIM module provides simplified 3D models for Digital Twin in GTF format via API (technical details will be specified during phase 2).
- BIM module provides extracted metadata from BIM models via API to EHR (API technical details will be specified in phase 2).
- How closely and how exactly the existing management of comments and notes during the procedural process should be related to the BIM module, must be clearly discussed and decided at the beginning of the phase 2.

4. BIM-based authorization procedure software

Since this is the first joint project of Future Insight and Reach-U building the first technical demonstrator, a great exercise to combine our expertise and technologies and experiment with the MKM components. It has delivered the first working demo comparable to the proof of concept, only with the new components and with the idea that it must ultimately work for real in the EHR. In this chapter we give a further explanation on the different parts and the capabilities of this first prototype.

4.1 Development of model upload and storage

The open source BIMserver project has not been in development anymore since early 2020 and the main developer has been working on a proprietary alternative. This product is called BIM.works and it too is based on open standards such as IFC and has an extensive API and SDKs in different languages (Java, JS, .Net) which makes integration quite easy.. The main developer is also part of the project, which ensures optimal flexibility. The agreement between MKM and Future Insight is that MKM gets a perpetual license for using the version of the software that is delivered at the end of the project for the purpose of doing building permit checking.



BIM.works does not only store the models, it also generates and indexes a lot of data so that queries can be executed efficiently and so that visualization of BIM data can be done in a browser. BIM.works uses both open source and other proprietary components for which all the licensing has been checked to be ok for the intended use.

4.1.1 Customizability

Since the source code of BIM.works is not open/provided and since MKM needs to have the ability to modify and/or add model checks, a special area has been reserved within BIM.works for plugging in custom code that can do model checking. For the original checks that will be written in this project the source code with documentation on how to change checks or add new checks will be provided. Any competent Java developer that has knowledge of IFC will be able to modify/create checks. The extent to which checks can be modified/added is limited to the types of functionalities that are already being used in the delivered checks.

For example, a new check could be added that checks whether all windows have a minimum width of one meter because another check that checks the same thing for doors already exists. However the method of calculating the actual floor area is not something that can be changed since this functionality is buried deep into BIM.works and also uses other proprietary software to calculate this.

4.1.2 Open Common Data Environment (CDE)

The progress on the OpenCDE initiative is closely monitored. When this becomes a finalized standard it will be integrated into BIM.works. This should make uploading to BIM.works directly from authoring tools much easier and perhaps will also play a role on inter-BIM system communication, but this remains to be seen.

4.2 Model visualization procedure in the building permit process

Just like in the Proof of Concept, we use the open source BIM Surfer as a visualization solution. As requested in the tender, various functionality has now been added. By this we mean the 'walkthrough' mode using keyboard keys to move around, multiple section planes, smooth transitioning of views for more context in navigation, touch screen navigation and the possibility to read additional geometry data using the standardized glTF format, the format also used by 3D Tiles for which we also implemented initial support. We expect more improvements to be developed throughout the rest of the project.

4.3 Model visualization solution in a 3D twin

A good integration between the BIM storage and the 3D Digital Twin is very important in the future and has therefore already been requested in the tender. BIM model visualization in a 3D digital twin can be useful for:

- Early contextual placement (get an idea of how the building to be built will integrate with the environment).
- Part of the building permit process, certain non-automatable checks are easier to be performed manually with the aid of contextualized visualization
- Building in-progress, for logistic planning purposes
- As-built integration into the 3D digital twin, so it serves as an up-to-date base layer for other processes

For visualizing BIM models in a GIS context (3D digital twin) the models need to be converted. The reasons for this are:

- The IFC format is not suitable for reading in directly, for example the geometry needs to be converted from the (many different) ways of describing geometry that IFC has to triangle meshes
- The models should be simplified because the amount of detail in IFC is too high for performant visualization in a GIS context (where usually you want to visualize much more than just the building).

4.3.1 Solution in BIM.works

BIM.works provides a good option for some of these functionalities.

1. Most IFC files contain information about what objects are "External" and which objects are "Internal", but often this information is not correct. BIM.works can calculate this information from the geometry
2. BIM.works can export any model or combination of models to GLTF2. GLTF2 is an open standard for geometry transmission on the web. Many 3D web viewers can work with this format including Cesium, ThreeJS, Babylon etc…
3. The JavaScript SDK of BIM.works can easily be embedded in any web application and helps the developer to manage tokens and URLs which are required for downloading the GLTF versions of models. This same API is being used in the EHR web application for building permits.

This allows a developer to quickly show simplified BIM models in a GIS context such as for example Cesium based software. The good thing about it is that this can all be done automatically and no layers/3D tiles need to be regenerated.

4.3.2 Future developments

To make the integration even better, BIM.works is planning on implementing the server side of the 3D Tiles Specification [1] which is also based on GLTF. This will allow integration into any 3D Tiles supporting viewer (such as Cesium) to be as simple as adding another layer with just one URL and no external dependencies.

Part of this implementation will also include generating different LOD (Levels of detail) to make viewing many BIM smooth and performant.

For the final bit (integrating as-built simplified models into the 3D Digital Twin more permanently) a conversion to CityGML might be very useful. The advantage would be that certain queries make more sense to be done in CityGML, which also allows to visualize query results by changing the appearance of buildings. Depending on the development on the next 3.0 CityGML specification and its reception, BIM.works contemplates implementing CityGML export as well.

[1] https://docs.opengeospatial.org/cs/18-053r2/18-053r2.html

4.4 First technical demonstration

4.4.1 Main purposes of the demo

Besides showing that the different parts actually work, the demo has a number of important other goals, these are:

- to demonstrate how the most important functionalities - file uploading, displaying check results and the 3D model, as well as interaction between the results and 3D-elements work with the renewed technology compared to PoC (e.g. switching from BIM Server to BIM.Works, using improved BIM Surfer etc);
- to start using EHR codebase and getting acquainted with its available components;
- to get actual technical hands-on cooperation started between Future Insight and Reach-U;
- to set up a basis for productive communication between all the project parties for planning the next steps in Phase 2.

For building the demo, EHR code-base and it's UX components were used as a starting point. For this, a local branch of the ehr-ui project was created. We added the BIM module to it as a React component. We currently use the following EHR components:

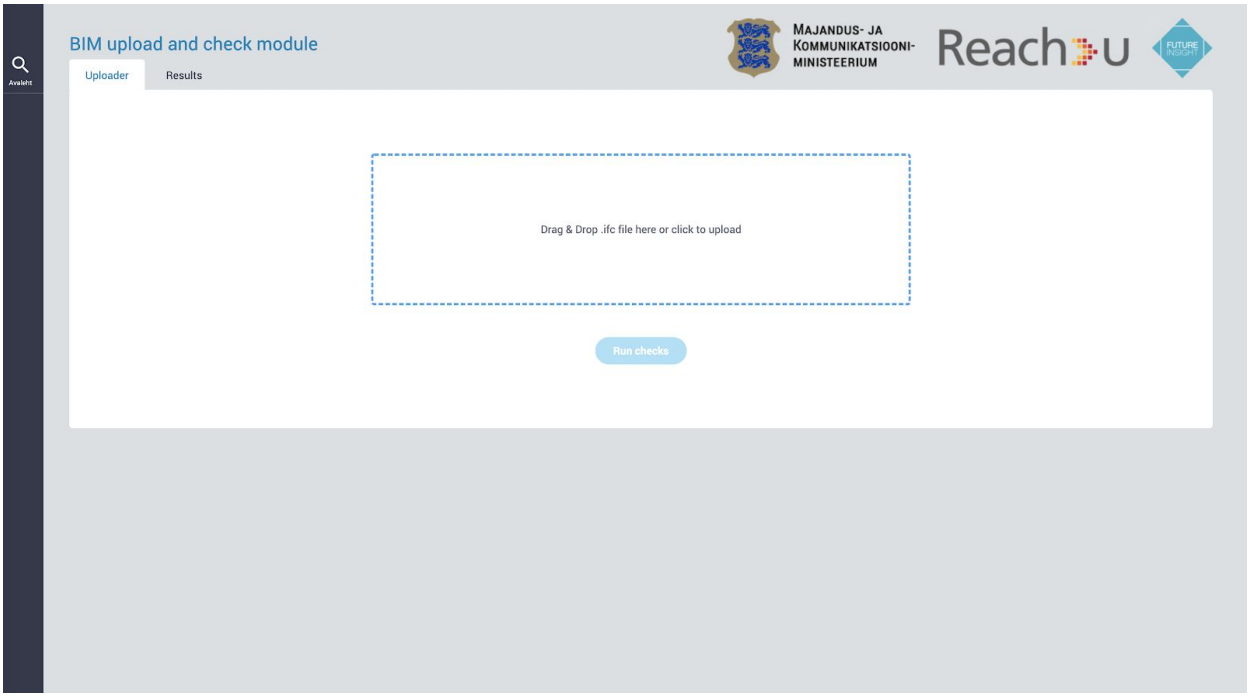
- Section, SectionTabs
- Loader
- Button
- Accordion, AccordionBlock

We deployed the code to Reach-U infrastructure. The BIM module executes API calls using the BIM.works JavaScript SDK. These API calls communicate with the BIM.works server.

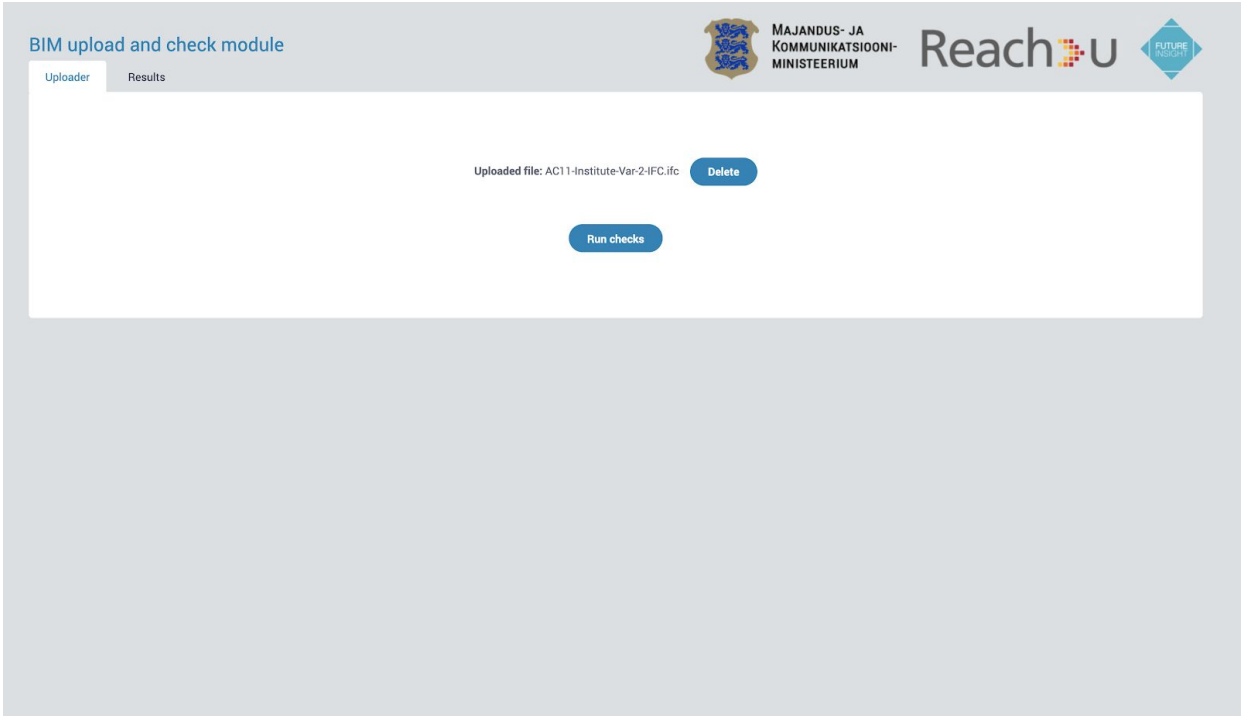
Most of the EHR components worked out-of-the-box and are suitable for fulfilling the requirements of the project. However, there were some minor issues with using the Accordion component and we foresee even more issues when we move forward with the development process. Therefore, this component either needs to be developed further or we will have to create a new component which more readily caters to the requirements of the project. It is also likely that other technical nuances will emerge during the development that cannot yet be foreseen, but a more detailed technical analysis at the beginning of the second phase will provide a more detailed overview in this regard.

At this stage, we have intentionally not contributed much to the visual side of the application UI, because we currently find it more important to move on with functionalities. Final structure and design of the user interface is planned to be carried out step by step in phase 2.

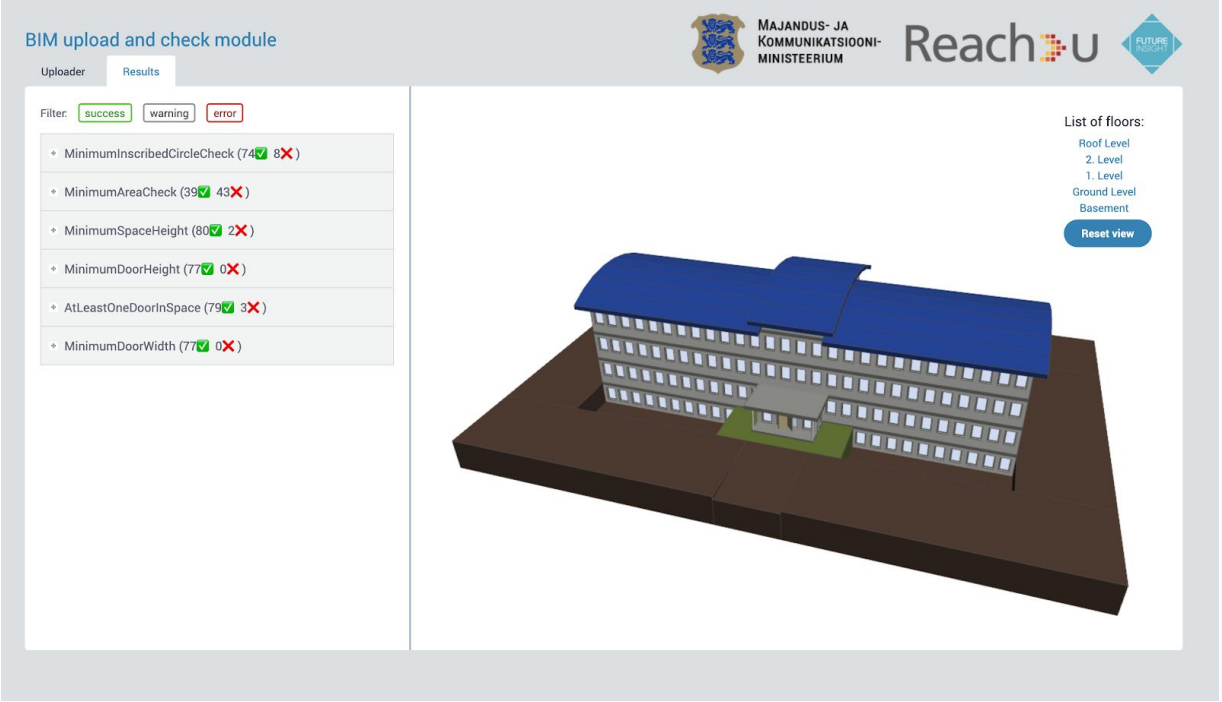
Screenshots from the demo



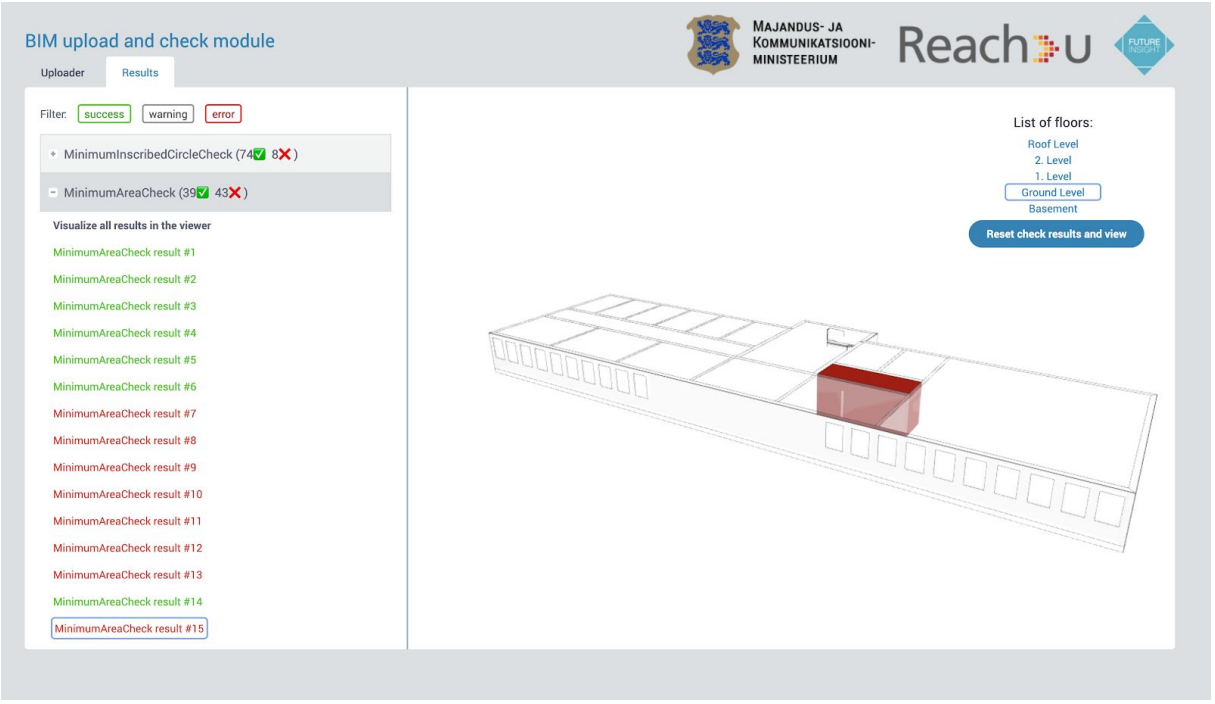
Add files



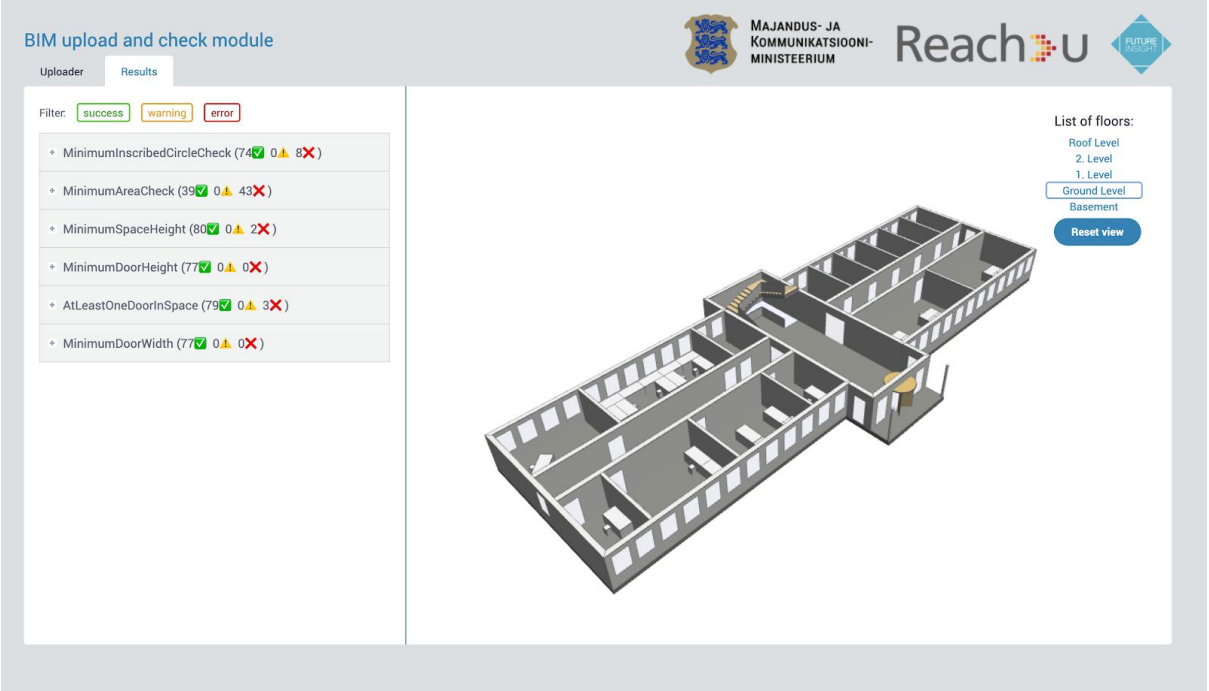
Uploaded files & execute checks



Check result list and 3D view



Selected check result and the corresponding element in 3D-view



Just one floor selected

4.4.2 Functionalities of the demo

When setting up the functionality of the demo, we took into account both the requirements from the tender and the wishes we saw from the user workflow analysis. The demo is currently available online via https://bim-demo.reach-u.com/ui/ehr/v1/ [1]. The following functionalities are currently available:

- Model upload
  - It is possible to upload either one or multiple aspect models of a design. Both for the Maleva as the Kari example design we used during the proof of concept these aspect models are available. However, only the Kari model also contains the Spaces required for multiple checks, making it more suitable for the demo.
  - It's possible to either add files by drag-and-drop or choose from folder Currently, the models are still stored in our BIM.Works cloud. In the course of phase 2 this will become a BIM.Works instance in the EHR infrastructure.
  - It is possible to delete the uploaded model
    We haven't paid too much attention to the file system yet, because this must be further organized in close connection with the EHR in phase 2.

- Execution of automated checks
  The first basic check functions that emerged from the check analysis have also

[1] It must be kept in mind that in the current demo solution each model uploaded is available for every user simultaneously.

been prepared. These have also been made available in the demo, so you immediately get a sense of how this could work with the available components. The checks that are already available are:

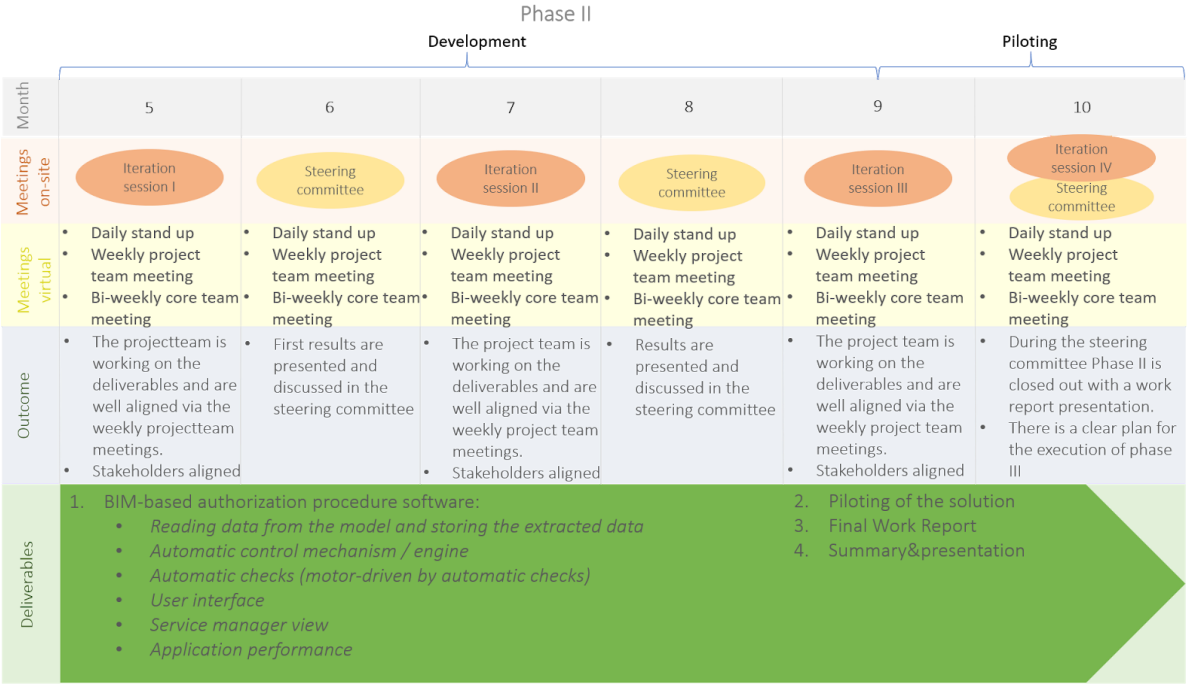
  - MinimumAreaCheck
    Check whether a Space is larger than a certain value (20 m2), 20 square meters was chosen for demonstration purposes, the actual value will be 8m2.
  - MinimumDoorWidth
    Check whether a door is wider than a certain value (600 mm)
  - MinimumInscribedCircleCheck
    Check whether a circle of a certain diameter fits within a space (2,4m)
  - MinimumDoorHeight
    Checks whether a door is higher than a certain height (1950 m)
  - MinimumSpaceHeight
    Checks whether a space is higher than a certain value (2500 mm). This does not yet take into account other elements in the IfcSpace or the fact that some spaces have sloping ceilings.
  - AtLeastOneDoorInSpace
    Checks if a Space at least had one door.

- List of check results
  - Grouping by checks
    The results are currently grouped by type of check and can also be filtered depending on the result. This way you can easily show only the objects which failed the check.
  - When clicked on a check result, the corresponding element is highlighted in 3D-viewer and only the floor of that particular check result is displayed. What exactly happens when a result is clicked can be set per type of check and depends on the information requirement.
    Technically, if clicked on a check in the list of results, RecipeExecutor component is called with the corresponding check ID, and RecipeExecutor itself knows that according to the check ID, the corresponding visualization logic must be run.
  - The grouping or filtering by legislation and by discipline is not implemented yet and will be done in phase 2.

- 3D-view of the model
    - Using the BIM surfer component the full BIM model is shown directly from the database. The requested WASD navigation has been implemented. A first implementation of displaying the immediate environment has now also been added using 3D Tiles. This has not yet been configured in the demo because further coordination with the 3D Digital Twin project is required here.
    - Making the outer shell of the BIM model available is implemented and described in paragraph 4.3. This will also require further coordination with the Digital Twin project.
    - It is possible to switch between the floors. Technically, if the 3D visualization component activates a floor, the React application is notified via listener, which in turn will activate the corresponding floor in the list.

5. Phase 2: The real (agile) development

The second phase is of course about the further development and design of the platform and the checks. At the same time, we also see from the first phase that there is still enough research and choices to be made. The agile approach proposed is extremely suitable for this, whereby it is still necessary to look carefully at which developments are given priority and which are pushed back a bit further. This is also reflected in the original planning from our project proposal.



However, based on the experiences from the first phase, we tend to work with more and shorter sprints, for example three weeks instead of 2 months. This helps to make the steps smaller and more concrete and at the same time to regularly adjust the results and choices to be made. As soon as the planning for these sprints is clear, we can start planning and preparing the iteration sessions.

At the beginning of January, after acceptance of phase 1, we will draw up a first backlog with all parts that need to be developed, both from the technical infrastructure and the checks. We will then prioritize these, after which a broad approach and prioritization for the sprints will be created. It is good to have the 'must haves' and 'should haves' at the top of the planning, so that they get the attention they deserve. One of the points of attention that we must also take into account here are dependencies and choices to be made on the side of the Ministry. We need to identify these as soon as possible and put them in motion in order to get them sorted out. After this we should be able to start with the first sprint at the end of January. As requested in the tender, it will be possible to pilot in the last two months, whereby we will have to further coordinate the exact details.

Annex 1: Overview of checks

Annex 2: Technical description of integration in 3D Digital Twin

To load simplified BIM models as GLTF from BIM.works, the easiest way is to use the BIM.works JS SDK. The SDK can be downloaded from https://bim.works. (Account required, anyone can create an account).

If IFC files have been geolocated, the geo location is automatically extracted from the model. If no geo location was provided in the model itself, the BIM.works API allows the user to change the geo location.

The generated default GLTF model is generally pretty small and only contains the triangles that are visible from the outside. This can be configured to:

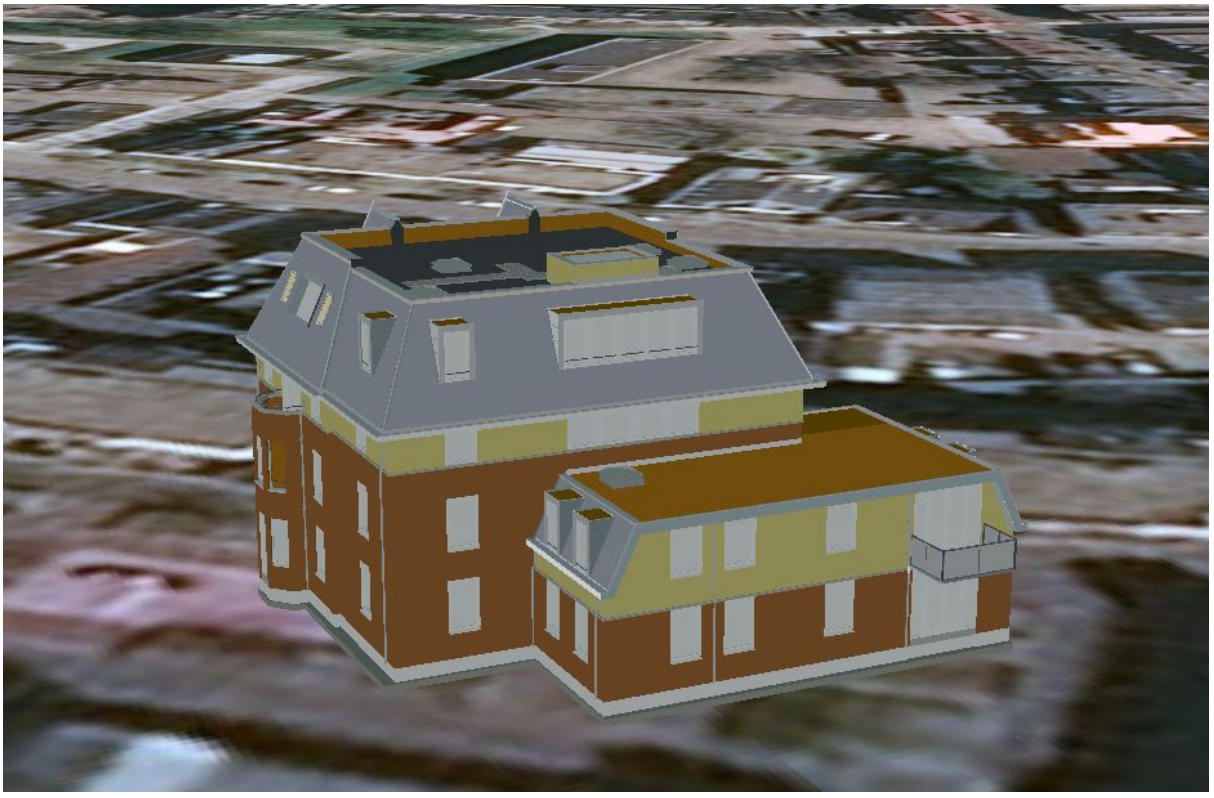
- Not only contain the external triangles of the external objects, but also contain all triangles of all external objects (usually solids). This results in larger files, but makes the models nicer (from the inside)
- Contain all objects, this increases file size a lot, but results in a complete model.
- Include all objects as GLTF nodes, this greatly reduces render performance, but allows the user to pick individual objects from the GLTF.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <script src="https://cesium.com/downloads/cesiumjs/releases/1.76/Build/Cesium/Cesium.js"></script>
rel="stylesheet"><link href="https://cesium.com/downloads/cesiumjs/releases/1.76/Build/Cesium/Widgets/widgets.css"
  <style>
        html,
        body,
        #cesiumContainer {
          padding: 0;
          margin: 0;
          height: 100%;
          width: 100%;
        }
  </style>
</head>

<body>
  <div id="cesiumContainer"></div>
  <script type="module">
        import { BimWorksClientFactory } from './sdk/bimworkssdk/bimworksclientfactory.js';

        const getPosition = (geoReference) => {
        const cartesianPosition = Cesium.Cartesian3.fromDegrees(
        geoReference.longitude,
        geoReference.latitude,
```

```
geoReference.elevation,
);
return new Cesium.ConstantPositionProperty(cartesianPosition);
};

const getOrientation = (geoReference) => {
const cartesianPosition = Cesium.Cartesian3.fromDegrees(
geoReference.longitude,
geoReference.latitude,
geoReference.elevation,
);
// https://github.com/CesiumGS/cesium/issues/6713
const rot = geoReference.rotation - 90;
const rotation = rot > 180
? rot - 360
: rot < -180
? rot + 360
: rot;

return new Cesium.ConstantProperty(
Cesium.Transforms.headingPitchRollQuaternion(
cartesianPosition,
new Cesium.HeadingPitchRoll(
Cesium.Math.toRadians(rotation),
0,
0,
),
),
);
};

const main = async () => {
const viewer = new Cesium.Viewer("cesiumContainer");

const auth_token = '[API_TOKEN]';
const versionId = '[MODEL]';

const factory = new BimWorksClientFactory('https://dev.bim.works');
const bimWorksClient = await factory.createClientWithApiToken(auth_token);

const token = await bimWorksClient
.tokens
.generateNewDownloadGeometry3DToken([versionId], { version: 1 });

const uri = await bimWorksClient
.executePreSignedDownload(token);
const ref = await bimWorksClient
.gis
.getGeoReference(versionId);

const position = getPosition(ref.geo_reference);
const orientation = getOrientation(ref.geo_reference);
```

```
        const entity = viewer.entities.add(new Cesium.Entity({
        name: 'test',
        orientation,
        position,
        model: {
        uri,
        }
        }));
        viewer.trackedEntity = entity;
        };

        main();
  </script>
</body>
</html>
```

Standard Schependomlaan model loaded in a default Cesium viewer (2.5 MB):



Annex 3: Draft extensive specification checks description