D2.8 – Layer Services description

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 958450. This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

Project Title	Improving Building Information Modelling by Realtime Tracing of Construction Processes
Project Acronym	BIMprove
Grant Agreement No	958450
Instrument	Research & Innovation Action
Topic	Industrial Sustainability
Start Date of Project	1st September 2020
Duration of Project	36 Months

Name and Number of the deliverable	2.8 - Layer Services description
Related WP number and name	WP 2 - Components, technologies & functionalities
Deliverable dissemination level	Public
Deliverable due date	31 August 2021
Deliverable submission date	31 August 2021
Task leader/Main author	Dag Fjeld Edvardsen (CATENDA)
Contributing partners	ZHAW, FhG-IAO, USTUTT, SINTEF, ROBI, VTT
Reviewer(s)	Andrej Cibicik (SINTEF)

Abstract

This deliverable describes the BIMprove systems' basic layer services.

General structure, interface structure, data structure, expandability, implementation, data security, software documentation as it exists in early August 2021.

The document will be updated at the end of each Cycle at M18 (February 2022) and M24 (August 2022).

Keywords

Software development, Virtual Reality, User Interfaces, UxV, Augmented Reality, IFC

Revisions

Version	Submission date	Comments	Author
v0.1	21.6.2021	Initial version	Dag Fjeld Edvardsen (CATENDA), Ruprecht Altenburger (ZHAW)
v0.2	16.8.2021	Correction	Matthias Aust (FhG-IAO), David Salvo (ROBI), Kaj Helin (VTT)
V1.0	30.8.2021	Approved, final version	Andrej Cibicik (SINTEF)

Disclaimer

Acronyms and definitions

Acronym	Meaning
IFC	Industry Foundation Classes, the file format for open building information models
BCF	BIM collaboration format, an open standard task-format with strong industry support
API	Application Protocol Interface
AI	Artificial Intelligence
VR	Virtual Reality
MUVR	Multi-User-VR
AR	Augmented Reality
BIM	Building Information Model(ling)
DT	Digital Twin
HMD	Head mounted display: "Goggles" for VR/AR/XR
XR	Extended Reality or Mixed Reality – either VR or AR or something in between
MUXR	Multi-User-XR
EKF	Extended Kalman Filter

BIMprove project

In the past 20 years, productivity in the European construction industry has increased by 1% annually only, which is at the lower end compared to other industrial sectors. Consequently, the sector has to step up its digitization efforts significantly, on the one hand to increase its competitiveness and on the other hand to get rid of its image as dirty, dangerous and physical demanding working environment. Construction industry clearly needs to progress beyond Building Information Modelling when it comes to digitizing their processes in such a way that all stakeholders involved in the construction process can be involved.

The true potential of comprehensive digitization in construction can only be exploited if the current status of the construction work is digitally integrated in a common workflow. A Digital Twin provides construction companies with real-time data on the development of their assets, devices and products during creation and also enables predictions on workforce, material and costs.

BIMprove facilitates such a comprehensive end-to-end digital thread using autonomous tracking systems to continuously identify deviations and update the Digital Twin accordingly. In addition, locations of construction site personnel are tracked anonymously, so that **BIMprove** system services are able to optimize the allocation of resources, the flow of people and the safety of the employees. Information will be easily accessible for all user groups by providing personalized interfaces, such as wearable devices for alerts or VR visualizations for site managers. **BIMprove** is a cloud-based service-oriented system that has a multi-layered structure and enables extensions to be added at any time.

The main goals of **BIMprove** are a significant reduction in costs, better use of resources and fewer accidents on construction sites. By providing a complete digital workflow, BIMprove will help to sustainably improve the productivity and image of the European construction industry.

Contents

Index of Figures

Index of Tables

1. Introduction

This document is about the BIMprove systems' as it exists in early August 2021.

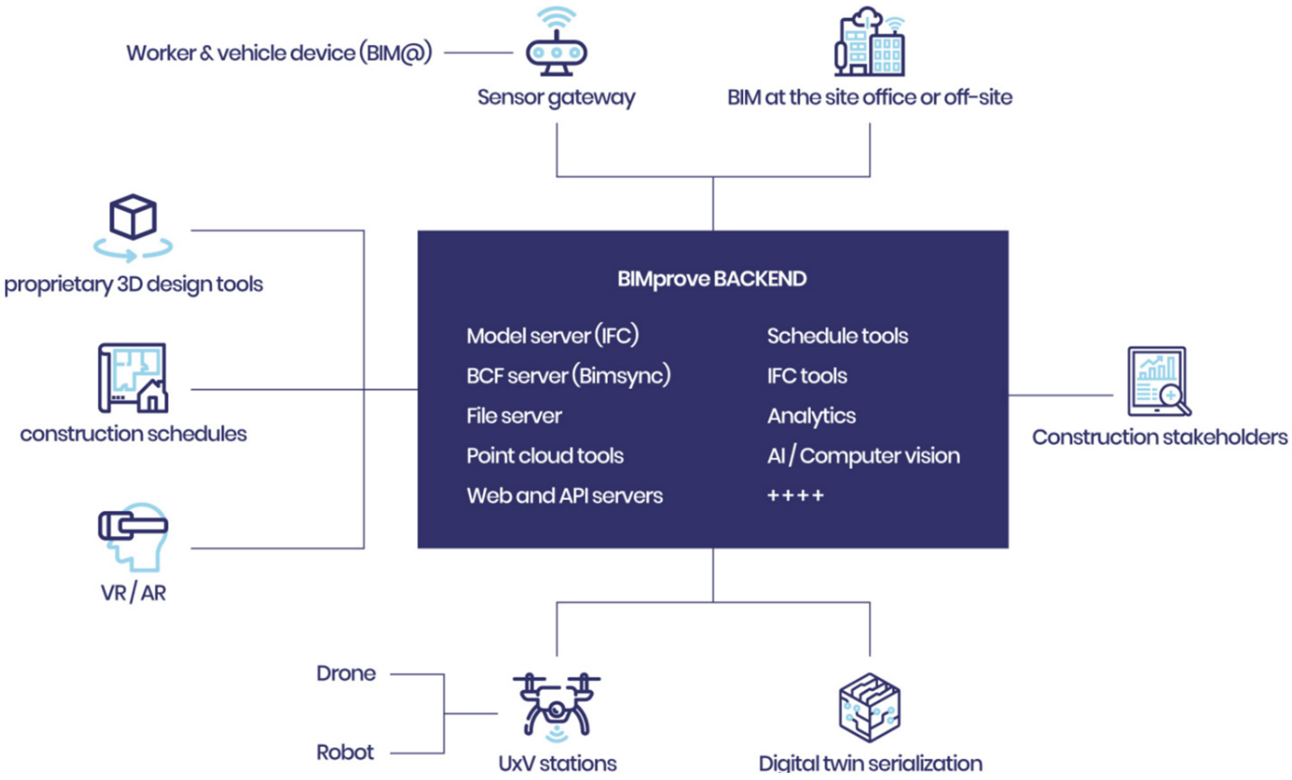

Figure 1 Overview of the BIMprove systems' satellites

2. BIMprove – development status and early documentation

In a future version of deliverable D2.8 a more detailed documentation will be presented. At the current point in time the system is still under development and many parts are in early versions and they are not yet integrated. For that reason, in this section we present a high level user journey instead of detailed documentation with screenshots. For each part of the system, we present a short development status.

2.1. Models and schedules uploading and viewing

The user uploads all IFC models and construction schedules to BIMprove using the web interface (or potentially via APIs). After doing this, the user can list his BIMprove projects and see a list of models and model revisions in the web GUI, and he can also see the uploaded schedule revisions.

Existing functionality: can upload IFC models to the model server via Bimsync directly or via Bimsync API. Can list BIMprove projects in BIMprove web GUI. Initial version import and viewing of schedule (MS Project MPP format).

Future functionality: Upload of IFC files and schedules via integrated web GUI.

2.2. Scanning planning, executing, processing and uploading

Day x (a random day) of construction a BIMprove operator reads that day's schedule, and extracts the model, the 2D and 3D geometry and the other information. This is used to plan today's scanning. After ordinary construction is done for the day, the ground robot and drone scanning takes place. The operator then generates the registered and BIM-aligned point clouds and uploads them to the BIMprove system. During scanning the semi-real-time position of the robot/drone can be seen via the web GUI. If there are interruptions or the robot/drone needs help they can “call home” so that an operator can help it (during development an operator will always be physically near the working drones).

Existing functionality: drone flying / positioning has started working, same with the ground robot.

Future functionality: the planning and point cloud cleaning, registering and sending.

2.3. Daily decisions

The purpose of this is to use information in the BIMprove system to follow up yesterdays activity regarding safety, accuracy and progress. The morning after the scanning (day x + 1) the BIMprove system is based on the BIM models, schedules and the point clouds created a list of tasks for the scheduler role to look at. There will be an “overall” task (in BCF format) with links to all the scheduled activity for the day being considered (day x). This “overall” task will always be created, and can be the basis for “free form” follow up tasks. In addition the system will auto-create issues based on the related construction tasks (from the schedule), related BIM building components and point clouds from the drone/robot. Each task can be followed up with considerations (decisions) related to safety, progress and accuracy. It can also be followed up with a reschedule request, a fix-accuracy request, fix safety request or a need-more-information request. Notice that this can be linked with (4) below, AR/VR for some innovative workflows.

Existing functionality: list, create and model-link BCF issues in model server (Bimsync). Initial version of comparing BIM building component with point clouds.

Future functionality: Automatically generate tasks based on BIM, schedule and point clouds.

2.4. Augmented (AR) and Virtual (VR) Reality

BIMprove will provide both an Augmented Reality (Hololens 2) and a Virtual reality (many compatible headsets) functionality. The user can start these systems either independently (just selecting the BIMprove system and activating device in that context), or from a BCF issue indicating location, object selected and text describing tasks to be done. The main input for the VR is the geometry for the BIM model(s) of the area you want to investigate, and the Point clouds of the same area. The main input for the Augmented reality system is the geometry from the BIM model, but it might be that Point clouds can also be utilized in VR and AR. In addition, the AR system greatly benefits if there are QR markers in the area you want to investigate. Ideally these markers are also in a BIM model so that the positions are easily verified. The geometry for AR/VR is being planned made available from the backend, with a possibility of partial models being investigated (for speed and memory purposes).

An advantage with AR/VR is that they can be very useful related to (3), the “Daily decision”: Comparing as-planned vs as-built using very immersive technology. In AR/VR you can open a BCF issue and comment/conclude it or create a new one if something is detected then and there. The linking to the backend is done using the standard BCF API to the BCF-/model server.

Existing functionality: View BIM-model in Multi-User-VR and in AR via intermediate format (IFC converted to a 3D geometry format); first preliminary interaction techniques and user interfaces for issue-creation.

Future functionality: Getting BIM/PCs from the backend, reading/updating/creating BCF issues in AR/VR.

2.5. AI and computer vision

The user uploads labelled images to the backend and initiates the machine learning training phase. After this is done the system can estimate what is seen in the new image and provide a probability/certainty number. When presented with a picture of a safety net, the system will – given enough training data – estimate that the image contains this with a high probability. Current tests indicate that it can recognise a safety net with high probability.

Existing functionality: Training and predicting system works with test data.

Future functionality: Integrate with the backend and create a GUI.

2.6. Worker / machine positioning

The workers and machines carry Bluetooth low energy based beacons. Based on the signal strength measured from a set of receivers, a position estimate can be created using triangulation. This can be visualized in 2D or 3D in near real time.

Existing functionality: Early version of positioning estimate based on signal strengths and known receiver positions.

Future functionality: Integrate with visualisation and send/receive on the backend.

3. The satellites of the BIMprove system

3.1. End-Users' Point of View: UI concept

The BIMprove UI concept has been detailed in Deliverable "D1.3 – Concept descriptions of user interfaces and human-robot interaction". The different UIs are categorized by their respective context of use. There are six UI categories:

- BIM@Construction

Figure 2 BIM@Construction visualization concept

- BIM@SiteOffice

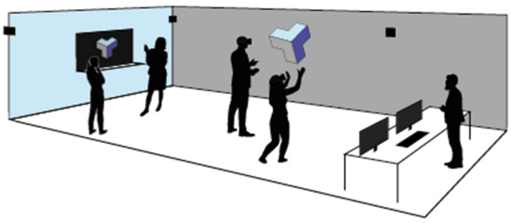

Figure 3 BIM@SiteOffice visualization concept

- BIM@OffSiteOffice

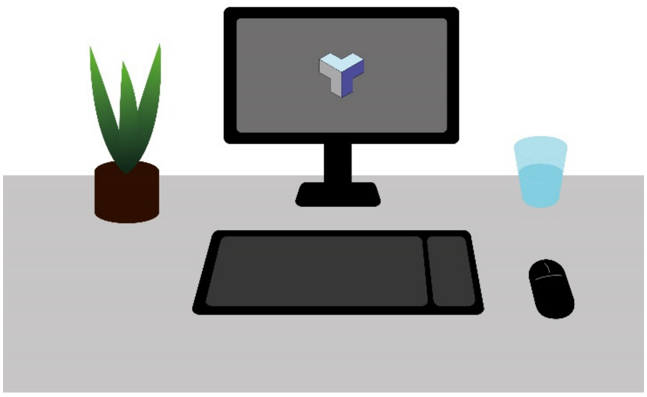

Figure 4 BIM@OffSiteOffice visualization concept

- BIM@Emergency

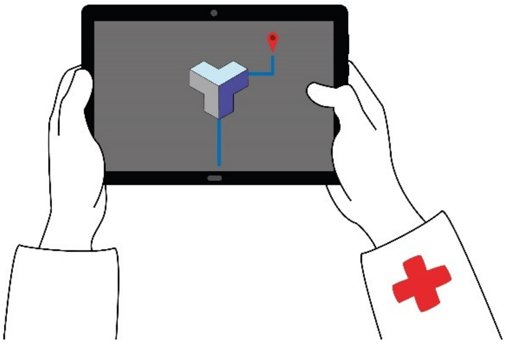

Figure 5 BIM@Emergency visualization concept

- BIM@Vehicle

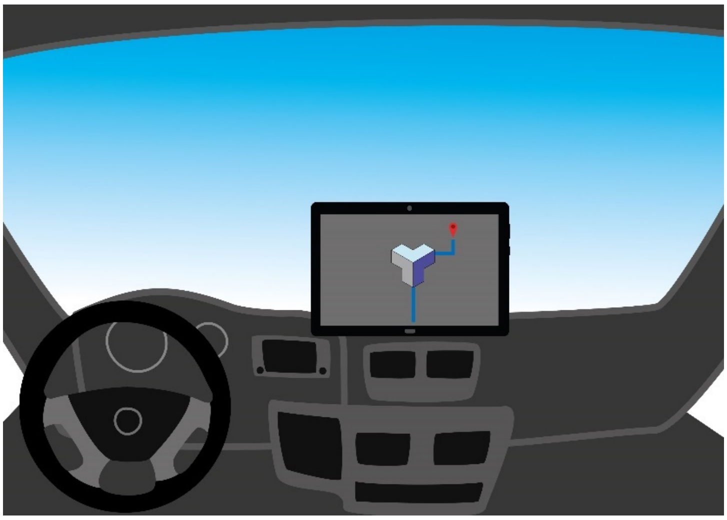

Figure 6 BIM@Vehicle visualization concept

- BIM@Anywhere

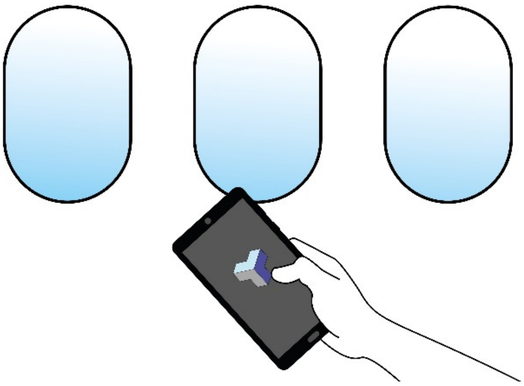

Figure 7 BIM@Anywhere visualization concept

The different functional prototypes of BIMprove-satellites, which are described in the sections below, and are to be used by end-users, fall into those categories:

Proprietary 3D design tools will mainly be used by engineers, such as architects, outside the construction site in remote offices -- BIM@OffSiteOffice. Construction schedules and cost tools will most likely be used at the construction company's premises or at the construction site office -- BIM@OffSiteOffice or BIM@SiteOffice. XR-tools and -devices will mainly be used at, and on the construction site -- BIM@SiteOffice or BIM@Construction.

As Digital Twin serialization is semantically at the core of the BIMprove-System and all BIMprove-related systems will have interfaces (direct or indirect, in some way) with the Digital Twin, this could be seen as falling into all UI-categories.

There will also be a web-based UI to the BIMprove-System based on Catenda's Bimsync-UI. It will include a model viewer, point cloud viewer, schedule viewer, issue tool, etc to provide stakeholders with a very rich view of the project without having to be on the construction site physically. This web-based UI will mainly serve for the categories BIM@SiteOffice and BIM@OffSiteOffice, but probably all other categories as well, as most devices will be able to display a web-based UI. This is especially relevant for BIM@Anywhere, which contains mostly mobile devices.

3.2. Proprietary 3D design tools

These days, before physical production starts on the construction site, a digital model is created showing the planned final shape of the building and the main systems. These digital models are more and more often BIM models (replacing 2D tools like AutoCAD), and we assume in BIMprove that the project actively uses open BIM (IFC) in its collaboration process. At the time when construction physically starts, we will typically have a site model, an Architecture model, a Structural model, and several models from the HVAC and other domains. It is not unusual that larger construction projects have 25 or more domain models. Ideally all the design would be completed before physical construction starts. In practice, this is almost never the case. Sometimes requirements like short deadlines make it beneficial to start construction before all design is ready, because that makes economic sense. Other reasons could be that errors are discovered in a domain model or in how domain models work together. In addition, external factors for the project can change. The client can change anytime the design, regulations can change, unexpected conditions might be discovered on the ground, or some building product might not be available. In reality, there are practically always some changes in the design models during constructions, and in BIMprove our system will be prepared for this.

The Architects, engineers and other domain experts design the BIM models in their own CAD tools. There are some very well known ones for the major trades, and some that are not so well known and might be mostly used in a few countries, and also some new software appearing. An advantage for the BIMprove workflow is that the quality of the open BIM (IFC) export of the most used CAD tools has become much better in the last few years. It is expected that this will further improve with stricter certification processes at buildingSMART International (voluntary but gives a certificate that is important to some customers).

BIMprove does not have an ambition to (re)create these CAD tools, but the existing ones are extremely important as their exports are (together with construction schedules) our main input. The different domain models will be imported in IFC format into the model server (Bimsync). After this the models are available via API for data queries and webGL visualization in the other parts of the BIMprove system.

The main flow of data from the CAD tools to the BIMprove backend will be through IFC models. Depending on feedback detailed 2D Drawings might be supported. The BIMprove systems and its satellite might send BCF issues ("please fix this") to the domain engineers via the CAD tools.

Dataflow out: IFC models

Dataflow in: BCF issues

3.3. Construction schedules and cost

This part of the system supports importing schedules and storing them in a database (MongoDB). Today import from the native Microsoft project files (.mpp) is supported. The plan is to add support for IFC files with tasks (Synchro can produce these). These schedules are visualized in an interactive Gantt chart. Each line of these charts is a task, and the GUI will support linking a task with a set of building elements. This is an important part of the BIMprove flow, since it is from this task-BIM relationship we know what is being worked on in a day-to-day perspective, and this is key input for planning the Robot / Drone scanning routes.

Dataflow out: Json file of schedule tasks

Dataflow in: IFC-file with tasks or native MS Project (.mpp), schedule file enriched with links to BIM objects

3.4. eXtended Reality (XR)

3.4.1. XR UI concept

This section describes BIMprove's XR based graphical user interface (GUI) concept. More details can be found in the public deliverable "D1.3 – Concept descriptions of user interfaces and human-robot interaction".

The construction Digital Twin Model has various layers and users with different needs and requests for information. There is no 'one-fit-for-all' GUI for all user roles and needs. The BIMprove's general UI concept is shown in Fig. 8. The UI concept has four modes:

- Digital Twin
- Immersive Digital Twin
- Mobile
- Notification and Warnings

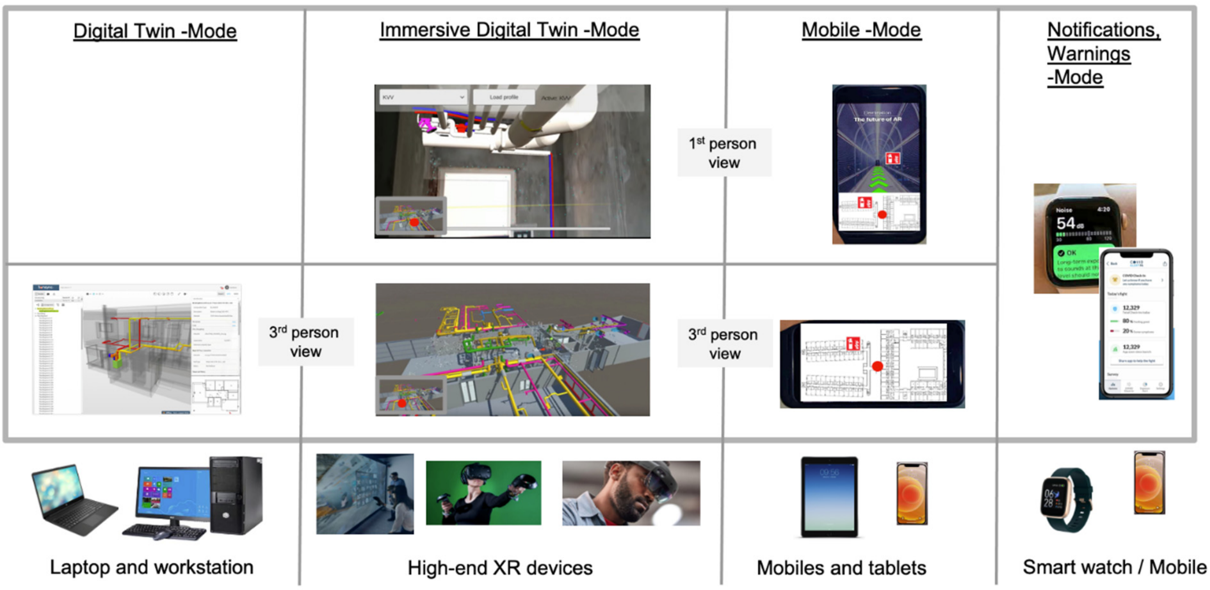

Figure 8 BIMprove's general UI concept

Users will be able to change modes based on context, device features, and user profile based access rights. They can have the 1st person or the 3rd person view in the Immersive Digital Twin and Mobile modes. The Digital Twin mode allows users to exploit just the 3rd person view. In the 1st person view, users will see information from the digital twin in real scale, e.g., can feel as if being located in a real room in Virtual Reality. In the 3rd person view, user can see information from digital twin at a selected distance, e.g., zoom in and out in a 2D map. There will be small sized overview maps on the 1st person or the 3rd person views, which support users' orientation in the Digital Twin model and give better situation awareness. The most urgent and relevant information will be given to the user via the Notification and Warnings mode.

3.4.2. VR

At the time of the first version of this deliverable (M12, Aug. 2021), the BIMprove Multi-User-VR-User-Interface looks as follows: Users log in to the system giving themselves usernames and entering a certain "room" when connecting to the system (see screenshot below). These steps, later in the project, might be handled via the BIMprove - Backend (semi-)automatically. As many as 20 users theoretically could enter the same room at a time with different devices. The system (and the UI) has been tested thoroughly with two users entering from two remote locations with VR-HMDs or using a PC-screen. The tests and their results are reported on in deliverable D3.2. It is also questionable and a part of the research over the course of the project, for how many users it makes sense to join in one XR session, depending on the specific use case.

After entering the VR environment, the users are able to view, inspect, and discuss the preloaded BIM model.

Figure 9 VR room connection

3.4.2.1 User's point of view

In VR the users (in the current state of development) have the following functionalities available to them. Shortly, here, are the functionalities HMD-users can use. PC-users, in principle, have the same options.

Navigation: The users can travel ("fly") through the model using their controllers (see the screenshot below) and a point-and-fly interaction technique with a rubberband-physics-metaphor.

Figure 10 Navigation setup

(De-)selection of models: Depending on how the model is subdivided, e.g. into discipline-models or by floors, the users can hide and unhide these sub-models to inspect different aspects, using the hand-UI and the controllers (see Fig. 11.).

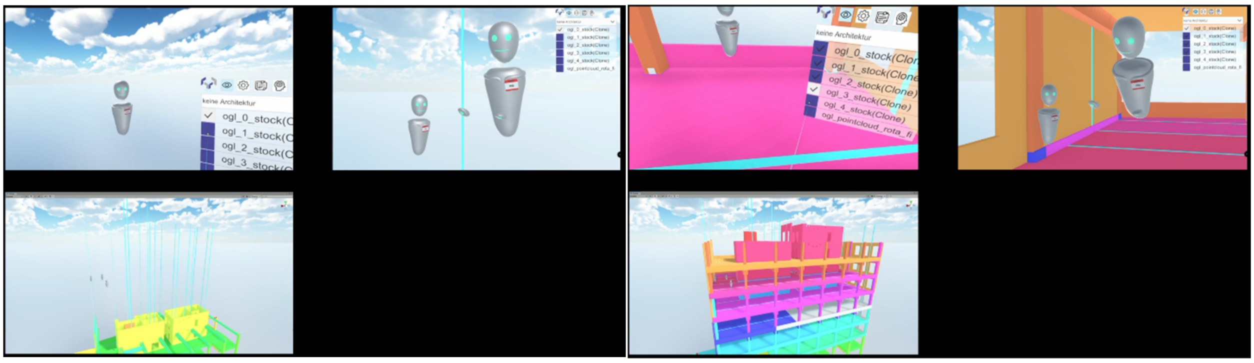

Figure 11 Selection of models

Communication: The users can see each other's avatars (see Fig. 12.), you can also see, when an avatar is talking, hear each other via voice chat and use a virtual laser pointer to show each other things in the model.

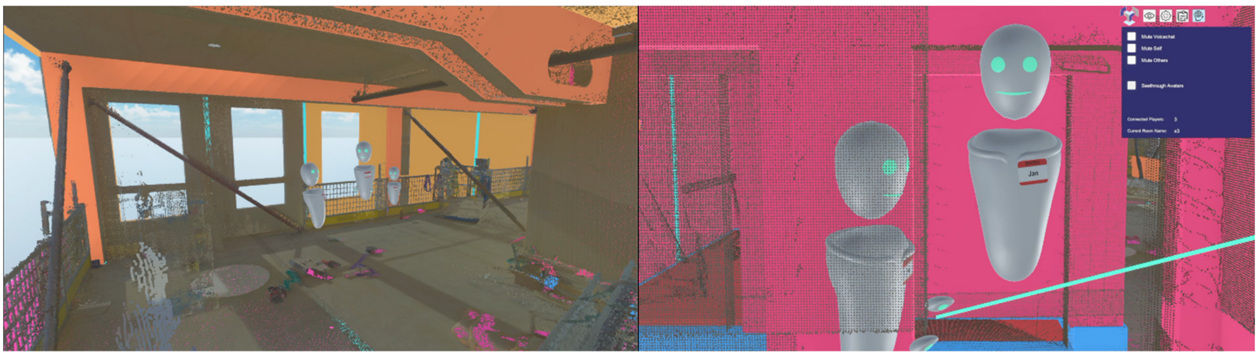

Figure 12 Avatars communicating

Issue management: The system also contains the first version of functionalities for issue creation and management. With their controllers and the hand-UI, users are able to create and position question marks in the model that represent issues -- these (and most likely other symbols, depending on the nature of the issue) will, in the future, be connected to BCFs. Currently, an issue can be created, given 3D-coordinates by positioning the question mark, edited via menu, and deleted again. The functionalities related to issue management will surely be revised, amended, and be subject to user tests many times throughout the project, as they are among the most important features of this user interface.

Communication about issues: Since a likely use case for Multi-User-VR will be, to, as a group, go through the list of issues one by one to discuss them, there is also the functionality to:

- click on an issue in the list to jump (navigate) to it directly, and
- summon everyone in the session to an issue in the list, in order to discuss this certain issue. (this functionality is new and has not yet been user-tested)

The HMD-MUVR-system will be for co-located and remote use, so that users at the site office can discuss the model, e.g., comparing the as-designed-model to the current as-built-model. Who can also be joined in their session from a remote location, e.g., for a planner engineer to give their opinion from their remote office. The motivation for a MUVR-system is the conviction that VR can a) ease communication about complex geometrical issues between users of different disciplines and b) induce a sense of presence for users which in return facilitates a state of flow. This will hopefully increase satisfactory task completion, overall usability and UX.

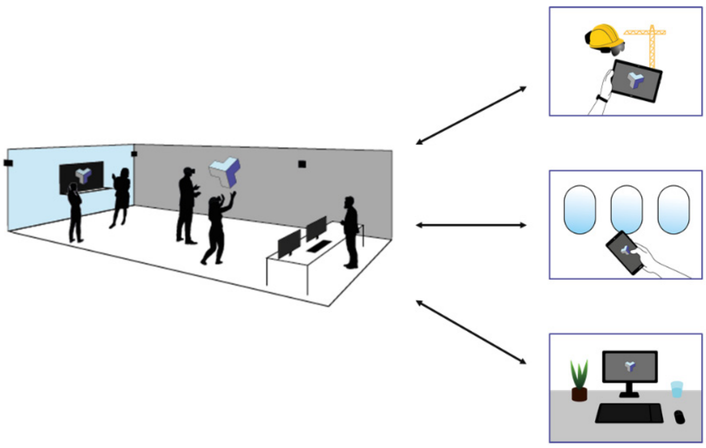

Figure 13 Visualisation of HMD-MUVR-system for co-located and remote use

MUVR will also be used when considering the "daily decisions" -- follow up on what was built yesterday (schedule + design vs point clouds) and considerations about accuracy, progress and safety considerations.

3.4.2.2 Multi-User-XR-Viewer

The XR-Viewer is used for Multi-User-VR, as it now stands. It is planned to facilitate joining a Multi-User-session with AR-HMDs and other devices, as well -- hence the name XR-Viewer. It has been developed using the unity game engine[1]. This has the benefit of being able to

- develop game-like applications fairly quickly,
- deploy to many different devices, and
- taking advantage of many plug-ins and SDKs that are developed specifically to be used with the engine.

The implementation of the Multi-User-functionality is an example for the last bullet point in the above list, since it has been developed utilizing Normcore[2], a multiplayer networking plugin for Unity. Only the usernames, users' positions (6 degrees of freedom), the voice chat, and the UI-menu are synchronised through Normcore. Which also means only this data runs through Normcore-Servers. The BIM-Models and point clouds will be loaded from the BIMprove-Backend to each user's individual machine/device.

3.4.2.3 Data interface to the BIMprove-backend

In the current state, by the time of the first version of this deliverable in M12, there is no true data interface between the BIMprove XR-Viewer and the Backend. This will be one of the main next steps in the development. The process of loading data still involves some manual steps which will be automated in the future and the XR-Viewer will evolve from a stand-alone application to a satellite of the BIMprove system, with an interface to load all data from the Backend, and feed results back.

There are two main ways of how data is loaded to the XR-Viewer: One concerns the as-designed BIM models (IFC) and, as the case may be, BCF-issues. The second one concerns point clouds (PLY) generated by UAV- or robot-scans, representing the current as-built status.

For IFC-models the current process is as follows:

- ifcConvert[3], an open-source command line application, is used to convert the IFC-geometry into Collada-DAE-files. The non-geometry BIM-parameters are written into XML scripts. These XMLs are not yet usable in the XR-Viewer.
- These DAE-files can then be imported in unity (see next step). But currently, it makes sense to have another manual step in between: Select and separate the geometry, e.g. to separate

[1] https://unity.com/
[2] www.normcore.io
[3]http://cad-3d.blogspot.com/2018/09/getting-bim-data-into-unity-part-9.html, https://forum.unity.com/threads/running-ifcconvert-exe-command-line-from-c-script.576859/, https://github.com/IfcOpenShell/IfcOpenShell/issues/397

the storeys of a building, to have useful "parts" of geometry to view in XR. This step can be completely omitted, or automated in some way, or made obsolete by delivering sensibly separated models from the Backend.

- The files are then imported into unity and, using a unity-editor-script, turned into unity AssetBundles[4]. This makes them more usable in unity-XR – prefabs[5] can be instantiated, they are more perfomant, they can be made collidable, and they could be (re-)loaded at runtime.
- This step is completely automatable and can be triggered by a command line.

One important reason why, currently, the geometry is manually separated (see the second bullet point in the above list), is one of the main neat features of the BIMprove XR-Viewer: The user-menu shown in the screenshots above is generated automatically from the data. The geometry files are named according to a certain convention, and by these filenames the software will structure the menu.

For point clouds in PLY-format the current process only differs from the above in the first step: We use an open source shader (provided via GitHab by user Keijiro Takahashi[6]) that turns PLY-formatted point clouds into meshes of squares that can be rendered performantly by Unity applications. Then, again, AssetBundles are created from these meshes by unity editor script.

Dataflow out: BCF issues

Dataflow in: IFC models, point clouds

3.4.3. HMD-AR

The Main idea behind the UI-category BIM@Construction is the use of BIM as digital twin data in immersive XR devices on the construction site. While AR devices like Microsoft HoloLens 2 are already built for interaction with 3D models (or “holograms”, as shown in the figure below (Building 3D model used in this concept application: "PROJETO HIDROSSANITÁRIO - BARRA VELHA / SC"[7] by Diago Fagundes, licensed under Creative Commons Attribution[8]), the scale of those models is usually dictated by the FOV of the HMD.

[4] https://docs.unity3d.com/Manual/LoadingResourcesatRuntime.html
[5] https://docs.unity3d.com/Manual/Prefabs.html
[6] https://github.com/keijiro/Pcx
[7] https://skfb.ly/6SBw9
[8] http://creativecommons.org/licenses/by/4.0

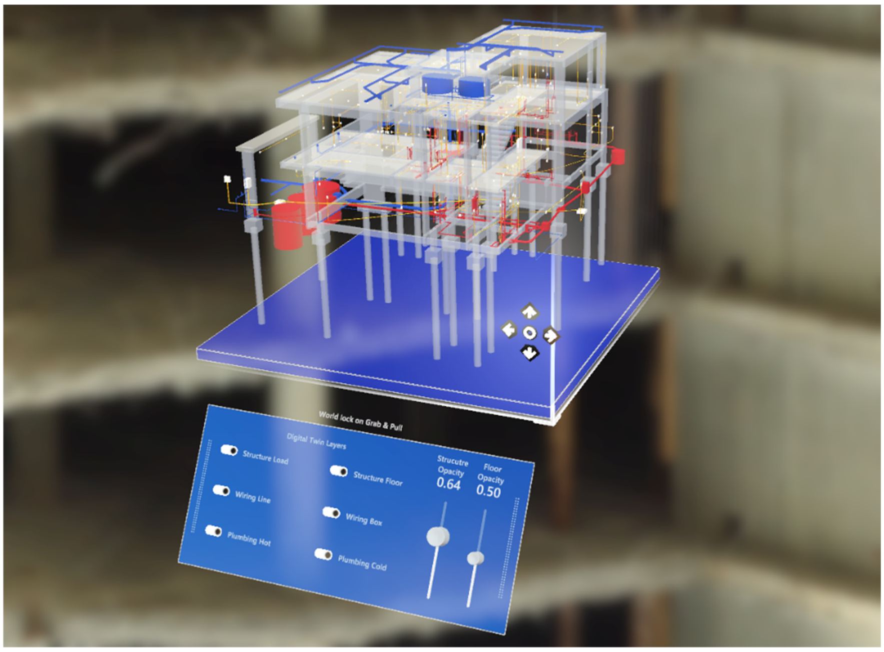

Figure 14 Augmented Reality demonstration

In order to overlay digital twin information on top of the real building, additional pivot objects should be created. Naturally, pivot object placement should match the placement of the fiducial makers in the real world. That way the digital twin and its real-world counterpart will be located in the same coordinate system (see Fig. 15.). Multiple markers and anchors are required because tracking precision and hologram stability will degrade with increasing distance to the marker.

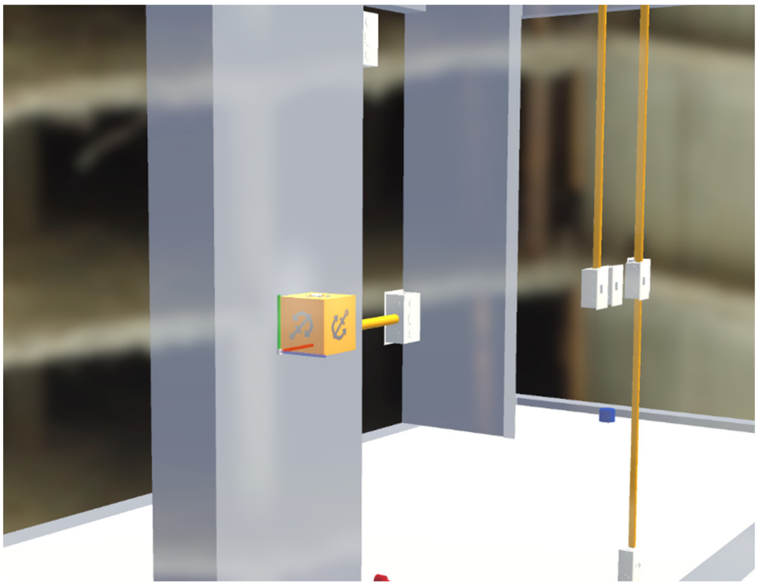

Figure 15 Markers and anchors

3.4.4. Handheld AR

Mobile mode version of BIM@Construction concept application (see Fig. 16.) is using computer vision for feature and plane detection in order to place AR object in the scene. It doesn't necessarily need the fiducial marker but works well with them since they can be used for on-site tracking and contain additional image features (texture and contrast variation). In the same way as a HoloLens application, the tablet version of the mobile mode provides both the 1st and the 3rd person views.

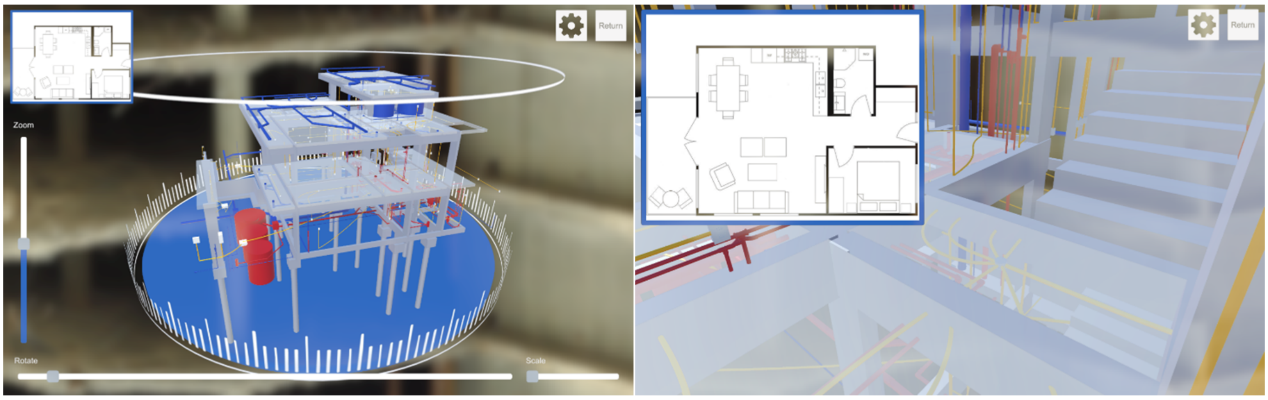

Figure 16 Mobile mode version of BIM@Construction

Dataflow in: Geometry of BIM showing relevant building geometry, marker positions, schedules tasks with location information/geometry

Dataflow out: BCF issues.

3.5. UxV stations

A request for spatial data is typically initiated by an end-user with some graphical input. Fig. 17. shows such a data capture schematically.

The input is done with a graphical user input on an end device: Browse to the specific room/region/part of the building, use different selection tools like: spot, line or area to specify the area that has to be scanned. The request is scheduled on the backend and sent to the data-capture system (DCS). Some analysis is performed to decide which system is asked to do the job (drone, ground robot, manual).

This information is sent and approved by a user via the backend. The DCS requests the latest BIM-data of the specific area/room. Currently this is given in a .svg format (2D-data). Based on that data the DCS calculates trajectories / waypoints for the specific drone or robot.

A permission to start is requested from the backend, together with an estimate of expected mission time. After approval the mission can start.

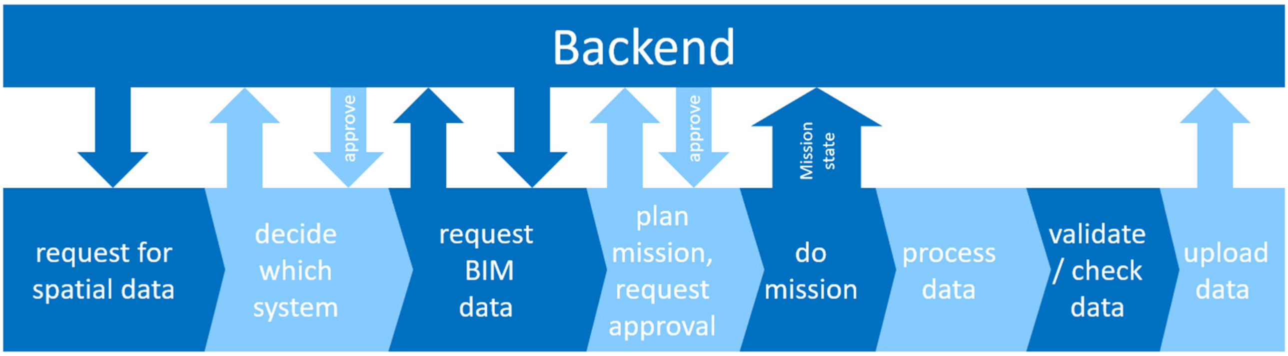

Figure 17 data capture schema for request of spatial data

Dataflow in: List of objects being worked on, 2D (svg) and 3D geometry from the BIM models

Dataflow out: Registered and cleaned point cloud, real time positions of drone/robot.

3.5.1.1 Drone based Data Capture System (DCS)

A more detailed description of the Drone part of the system is to be found in deliverable D2.2. Here is a short summary about the current status.

The Drone based DCS consists of 2 main systems: the drone itself (UAV) and the ground station (GS). Both contain several subsystems:

Ground Station (GS)

Table 1 Ground Station subsystems

Sub-system	Description	Progress
Ground Control	Control and visualize drone on ground, main connection to drone	first draft in Matlab (see below)
Pix4D-System	Converting image data to point clouds	commercial system, API under dev.
Connection Node	interface between backend and ground station	---
Mission Planner	planning missions, based on BIM Data	first draft
Charging System	Landing base, charging batteries of drone for long autonomous operation.	first prototype

Fig. 18. shows a snapshot of the Ground Control software. It shows a 2D view of the drone and further information of it (position, orientation, flight state, basic information from sensors etc.). Currently the position data is given in a local frame, initialized at startup.

The connection from GC to the drone goes via a 5.8GHz WLan with a lean UDP-protocol@1.25Mbaud. In general the drone does not depend on this data connection. After uploading the mission and arming it is fully autonomous and does all calculations on-board.

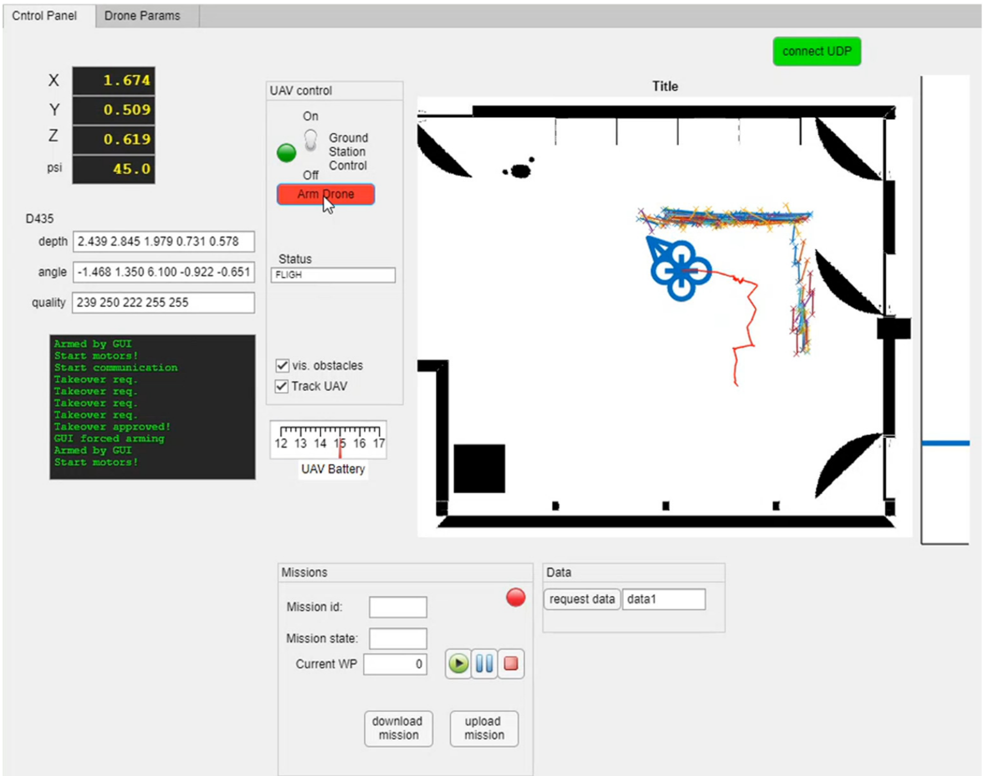

Figure 18 Ground Control

UAV / Drone

The drone itself consists of several hard/software subsystems, the main are:

Table 2 Drone hard/software subsystems

Sub-system	Description	Progress
Flight controller	The main system that guarantees a stable flight	90%
Pathplanner and Communication (RPI4)	corrects / recalculates flight paths based on measured obstacles	20%
further specific software (e.g. RPI0)	further specific software esp. on sensors	80%

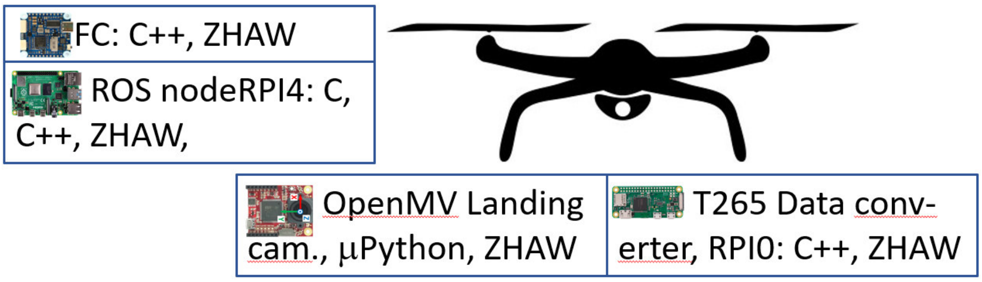

Figure 19 UAV / Drone subsystems

Fig. 20. shows a very simplified abstraction of the main threads, running on the flight controller (FC). There are several connections to external devices where a fast and robust data connection is needed. Several connection types are used: UART, SPI, I2C, UDP. The main motors are driven by a standardized PWM at 200Hz.

Controller Thread: The thread running with the highest priority at 200Hz is the controller thread with attached EKF (18 states at 50Hz). This thread calculates 3 torques (Mx, My, Mz) and thrust Ft, that are given to the motors. The calculation is based on desired values for position and orientation and on different sensor signals. The fastest sensors (gyro and accelerometer) are read in this thread and processed directly.

Main State Machine: runs at 50Hz with high priority and is responsible for the general flight management, like, arming/disarming motors, enable controllers etc. It gets direct input from a standard remote control (RC) but also via the WLan data link from the GUI.

External Sensors: runs with high priority and reads other sensors like z-Lidar, optical flow etc.

UDP communication: runs at lower priority. This thread sends and receives data via the WLan module to the ground station and gets data from there.

Data logger: Runs at medium priority. Here on-board data (approx. 65 values) are logged to an on-board SD-card at 50Hz. It is mainly for data analysis and optimizing controllers, analysing failures etc.

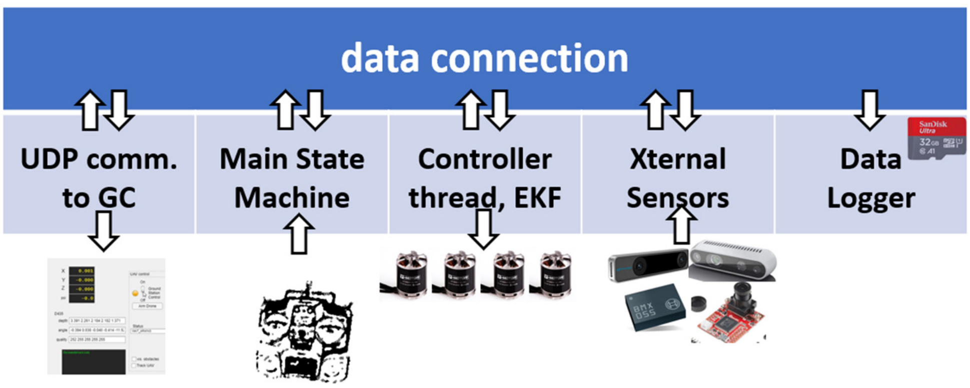

Figure 20 Flight controller

Dataflow in: Geometry of BIM showing relevant building geometry, marker positions, schedules tasks with location information/geometry

Dataflow out: scan plan, semi real time position of UxV, Point clouds, images, (maybe) thermal images

3.5.2. Robot

A much more detailed description of the Robot part of the system is to be found in deliverable D2.2. Here is a short summary about the current status.

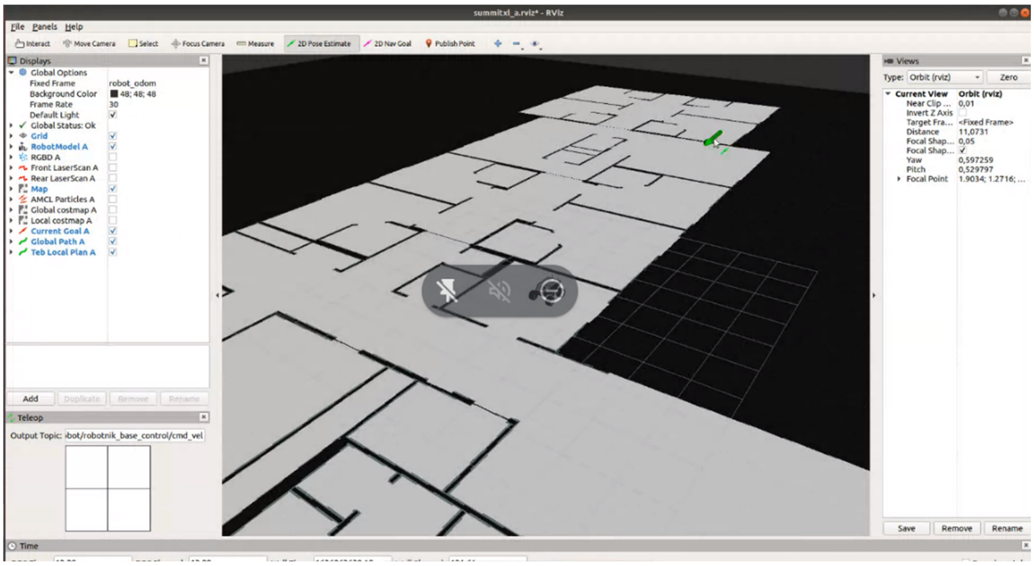

Figure 21 Floor plan in Rviz

Fig. 21. Shows a floor plan that is exported from the model server (Bimsync) as svg and imported into Rviz. Both this 2D floorplan and a 3D geometry are sent from the model server to Rviz/ROS via a conversion script.

Based on this we have simulated the robot scanning the building storey, and it is shown on top of the floor plan where the scanner detects hinderances (typically walls). This is done using the scanner the robot uses for autonomous navigation, so it has significantly less resolution than the BLK360 laser scanner.

The current state is that we can simulate with BIM-based data the robot navigating in the Rviz software. With this integration, it is possible to obtain a 2D map and 3D model of a scene from BIM-sync and to be able to simulate the operation of the robot in it without the robot physically being in that space.

Initially, the robot will perform the first scan of the place to generate the 2D map and be able to navigate and locate itself correctly. Thanks to this integration, it will allow to include new information about the previous map without the robot having previously scanned it. In order to carry out the study of this integration in simulation, the 3D model associated with the 2D map has been used to test the navigation and location of the robot.

The next steps will be to improve the compatibility between 2D and 3D maps, correctly setting the scale of both to fit a correct representation of the scenario in the virtual environment. On the other hand, the robot's laser data will be integrated with the Bimsync backend in order to be able to interact with the information from the maps generated in real time.

3.6. Digital Twin serialization

The Digital Twin (DT) is a core concept of BIMprove. In a sense the DT is the current status of all data in the system, since that is our full digital representation of the buildings current status and history. It is used by many parts of the system, and these part puts updated information back. So the DT is what makes the BIMprove functionality possible. But in a building life cycle perspective, it also has great value for those who will maintain the building. Among many other things it can show what actually is inside walls.

To export the digital twin we want to use open standards as much as possible. BCF for the issues, IFC for the models, e57 (or other open formats) for point clouds, pdf for building drawings, jpg for images etc. Further research inside BIMprove is needed and will be done regarding how to make it possible to make the DT as accessible as possible after the construction process has ended.

Dataflow in: Models, schedules, BCF, real time positions, point clouds etc.

Dataflow out: The same as in, but also enriched during the BIMprove process.

3.7. BIM@SiteOffice or BIM@OffSiteOffice

This will use the Model viewer, point cloud viewer, schedule viewer, Issue tool, etc to provide stakeholders with a very rich view of the project without being physically on the site. The basic interface will be web based, but a very important part will be using Virtual Reality to get an immersive experience of the design models and point clouds. It will also be used when considering the "daily decisions" - follow up on what was built yesterday (schedule + design vs point clouds) and considerations about accuracy, progress and safety considerations.

3.8. Sensor gateway and user locations/warnings


```
[{
        "MACid": "C1:E3:6D:10:E3:98",
        "x": 925,
        "y": 438,
        "timeStamp": "2021-06-29T10:10:14.1084832+02:00"
    }, {
        "MACid": "C1:E3:6D:10:E3:98",
        "x": 1140,
        "y": 1113,
        "timeStamp": "2021-06-29T10:10:17.8574419+02:00"
    }, {
        "MACid": "C1:E3:6D:10:E3:98",
        "x": 447,
        "y": 155,
        "timeStamp": "2021-06-29T10:10:18.6244034+02:00"
    }, {
        "MACid": "C1:E3:6D:10:E3:98",
        "x": -2147483648,
        "y": -2147483648,
        "timeStamp": "2021-06-29T10:10:21.1516369+02:00"
    }, {
        "MACid": "D3:C8:C1:79:8D:B3"
```


Figure 22 Json format for BLE marker positions

Based on Bluetooth low energy devices, worker and machine positions can be calculated based on signal strengths received. The Json format from the sensor gateway to the backend is simple, it is an array of MACid, x-position, y-position, and timestamp. The positions are in centimeters. The" MacId" is really referring to the Bluetooth Low Energy Adresse.

Notice that the beacons are "discovered" by the receivers at different intervals, and triangulation is done by taking the signal strength and doing calculation against a calibrated grid. A Kalman Filter is used.

Current estimation of the sensors precision seems to be about +/- 1.5 meters (after filtering obvious outliers, like the last one in the screenshot above), but that was measured in a test with two receivers and also with no information gathered about cross-sender information exchange. We will learn more about the actual precision when it is estimated on a construction site. Then there will be both larger distances but also more receivers. The system also supports sending real time warnings to users if for instance a machine/vehicle (a truck) is backing up close to a worker, but we have to consider if this will work well in practice.

The data from this will be presented in "near" real time on one of the web pages. This can be shown in the 2D visualization widget that Bimsync API provides. If we get access to history through a time period (like a day) we can show a heat map of positions. This is also a question of what is possible related to privacy policy and rules.

Dataflow in: List of active devices.

Dataflow out: semi real time position of workers/machines.

4. The Backend of the BIMprove system

4.1. The main part of the backend

The main part of the backend is developed in Python (version 2.8, which is a long time release), and uses FastAPI for Web/API serving. A benefit with FastAPI is that it has very strong support for OpenAPI / Swagger. Related to web, it uses Jinja2 for templates and Bootstrap 5 as a html/css presentation library. In the development phase, the different parts of the backend are developed/tested on local machines. There is no new cloud server yet, the reason is that in this phase of the development that makes coding faster. The idea is that until the parts are ready for deeper integration, fake/mocking objects with correct shape are used to simulate integration. When deployed on a cloud server, it will probably be on Linux - but have as an ambition that the backend and as much as possible other parts should be cross platform (Windows / Linux / Mac). The model and BCF server (Bimsync) are cloud based, but the other parts should be able to run locally on "any" machine.

4.2. IFC tools

Open BIM / IFC is a very central part of the BIMprove workflow. The model server (Bimsync) has strong support for IFC, but it is not designed to do a lot of direct manipulation of IFC files. But it has an API that allows for uploading, downloading and querying of IFC files. So it is designed to work well with other tools, and in BIMprove we will use IfcOpenShell (by Thomas Krijnen) for IFC manipulation, comparison and explicit mesh extraction

4.3. Point cloud tools

BIMprove will to a high degree make use of and also create tools related to point clouds. Among these will be tools that can compare 3D geometry from an IFC design model with observed point clouds data. So this works together with the IFC tools mentioned above. We use Open3D, Trimesh 2 and other existing libraries to support the development. Based on this we are developing tools that can color points based on (signed) distance to nearby meshes, supporting the "daily decisions" regarding progress and accuracy. The tools can provide suggestions, but the actual decisions will be made by a human.

4.4. Model server

Bimprove uses Bimsync as its model server.

4.5. BCF server

BIMprove uses Bimsync as its BCF-server.

4.6. Web, File and API servers

The D2.4 and D2.5 deliverables describe this in detail. FastAPI is used as the Web, file and API server. One big advantage is that it has strong support for OpenAPI / Swagger and the related ecosystem of tools.

4.7. Schedule tools

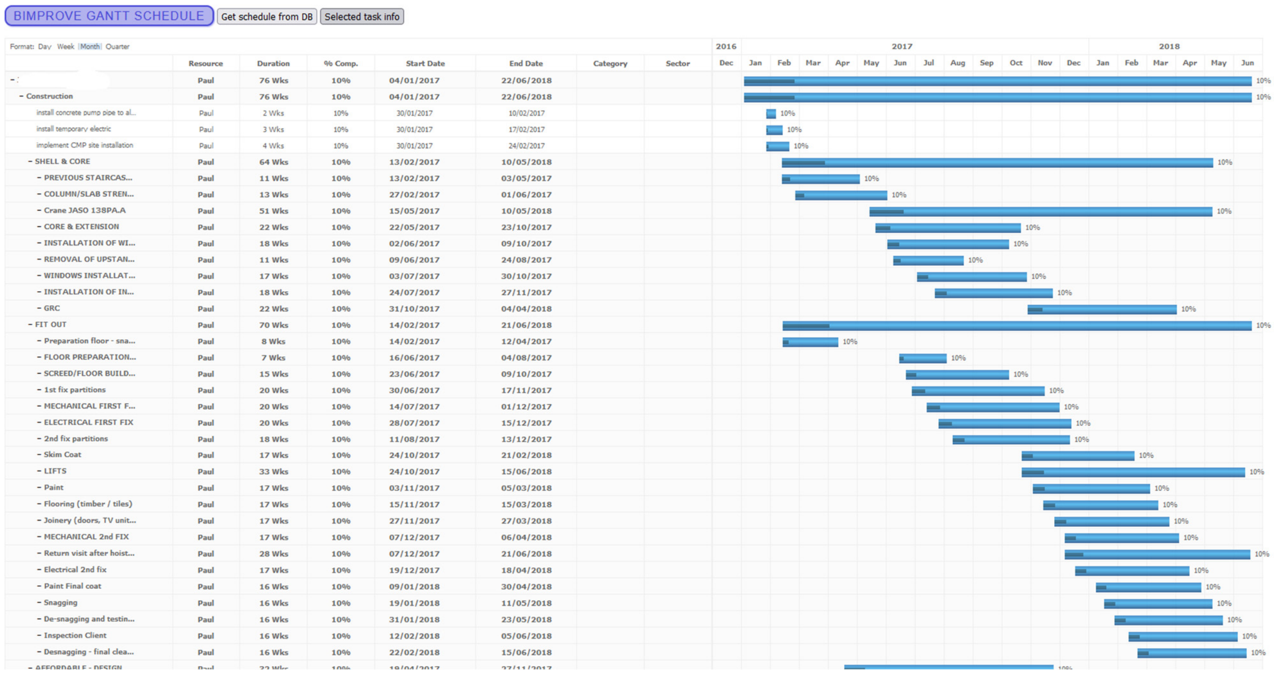

Figure 23 scheduling tool demonstrated

This tool uses the open source "JsGantt improved" as a GUI to visualize the interactive Gantt chart. Before this can be shown schedule revision is imported in native Microsoft project format (4D IFC will be supported later). The screenshot is from a "random" project MPP file we got access to for internal testing (project name deleted). In the future we also want to be able to import from IFC files with 4D data embedded. This makes it possible to examine the schedule in the GUI, but we will add support for linking a task with a set of IFC building components via a GUI where both GANTT and the BIM-models are shown at the same time.

4.8. Analytics

The core concept is to use the rich set of data (BIM, schedules, Point clouds, sensor data, worker and machine positions etc) to get new insight. The plan is to use a statistical approach and to test hypothesis that we formulate. Examples: One hypothesis could be that if a task is added to the schedule a short time before it is initiated on the construction site, it will more often lead to rework (since it might not be as well defined and discussed in a multi-discipline setting). Another hypothesis is that there will be fewer "undesired events" when using the BIMprove system historical company averages for similar systems. It might be that the statistical analysis would happen offline in tools like R, Stata or a Python based tools - but the required data should be available, preferably via the API. There can also be "soft" analytics, where we present in graphs relations between two or three dimensions so that it might trigger the development of hypothesis by graphing combinations that might be correlated.