Aalto University
School of Science
Master's Program in Human Computer Interaction and Design

Mao Lin

Augmented Reality Solution for Facilities Maintenance with BIM

Master's Thesis
Espoo, September 26, 2019

Supervisors:	Professor Yu Xiao, Aalto University Jean-Marc Vézien, Paris-Sud University
Advisor:	Petra Isenberg, Paris-Sud University

Table of contents

Acknowledgements

- Prof. Yu Xiao (Aalto University)
- Jean-Marc Vézien (Paris-Sud University)
- Tero Järvinen (Granlund)
- Juha Tuomainen (Granlund)
- Joni Turunen (Granlund)
- Veikko Martiskainen (Granlund)
- Jing Xu (Granlund)
- Mika Hakkarainen (VTT Research)
- Petri Honkamaa (VTT Research)
- Otto Juhala (ISS)
- Kim Nyberg (Trimble/Tekla)
- Jussi Ketoja (Trimble/Tekla)
- Teemu Berglund (Immersal)
- Dong Jiang (VimAI)
- Ismo Olkkonen (VimAI)
- Heidi Ali-Kippari (Granlund)
- Joni Turunen (Granlund)
- Francisco Forns-Samso (Granlund)
- Markku Vuori (Granlund)
- Davor Stjelja (Granlund)
- Anna Korolyuk (Granlund)

Abbreviations and Acronyms

AECO	Architecture, Engineering, Construction and Operation
AHU	Air Handling Unit
AR	Augmented Reality
BIM	Building Information Modeling
BIS	Building Information System
COM	Concurrent Odometry and Mapping
FoV	Field or View
FM	Facility Management
HMD	Head-Mounted Display
HVAC	Heating, ventilation, and air conditioning
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
MAR	Mobile Augmented Reality
OHSA	Occupational Safety and Health Administration (USA)
PLM	Product Life Management
PoI	Point of Interest
Pose	Position and rotation
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
UCD	User-Centered Design
UX	User Experience
VIO	Visual Odometry
WIFI	Wireless Fidelity

Summary

Aalto University

School of Science

Master's Program in Human Computer Interaction and Design

ABSTRACT OF MASTER THESIS

Author:	Mao Lin		
Title:	Augmented Reality Solution for Facility Maintenance with BIM		
Date:	Sep 26, 2019	**Pages:** 69	
Major:	Human Computer Interaction and Design (EIT-HCID)	**Code:** SCI3020	
Supervisor:	Professor Yu Xiao (Aalto University) Jean-Marc Vézien (Paris-Sud University)		
Advisor:	Petra Isenberg (Paris-Sud University)		

Building information modeling, or BIM is a term frequently employed in the architecture, engineering, and construction and operations (AECO) industry nowadays. BIM is widely accepted not only for the planning and construction phases of AECO projects, but also the maintenance phase for Facility Management. On the other side, Augmented Reality, as an emerging technology is considered having great potential to be applied in AECO context. However, how to combine BIM and AR technology together to design a practical solution for Facility Management is a challenging work. Most of the previous works either focus on specific type of user scenario or with various kind of limitation or compromise.

The goal of this project is to find out a practical Augmented Reality solution for Facility Management context, with the support of BIM. In this project, in parallel with a careful user study and requirement analysis following User-Centered-Design process, a comprehensive technical evaluation is performed for selecting the proper tools and technology. Based on the result of both tracks, we designed concepts and the technical solution, which includes both intuitive user interfaces and an architecture with good maintainability and extendibility. In the end, we performed an evaluation and discussed about the limitation and possible future improvement.

Keywords:	Augmented Reality, BIM, HVAC, Cloud-based, Marker-less, Facility Management, Facility Maintenance, Digital Twin, Anchors
Language:	English

1 Introduction

1.1 Background

Building Information Model, or BIM is a digital representation of physical and functional characteristics of a facility. As such, it serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its life cycle from inception onward[1]. It is not only a tool but also a process based on 3D representations of building objects with related information such as the properties and relationship with other components. It is a term that frequently employed in the architecture, engineering, and construction and operations (AECO) industry nowadays, and it is beneficial for the whole life cycle of an AECO project, which includes not only the planning and construction phases, but also the maintenance phase for Facility Management.

The Digital Twin concept, on the other hand, referred to a virtual representation of a physical product in the product lifecycle. (Grieve et al, 2011)[16]. In AECO context, digital twin refers to the digital representation of a building, including not only its static geometry properties but also its dynamic status information. It is considered as a beneficial tool for increasing productivity in the whole life cycle of an AECO project. BIM technology provides a practical solution to enable the digital twin concept, it let us define structured and rational building information into BIM properties.

Augmented reality (AR) is an emerging technology that provides users interactive experience of a real-world environment where the objects that reside in the real-world are “augmented” by computer-generated perceptual information. AR is considered to have a great potential to be applied for facility management scenario. For example, an AR application can be a portable application, running on hand-held mobile devices (Irizarry et al., 2012)[5], or AR head-mounted display (HMD). The application must be able to recognize the target object in the maintenance task and overlay the relevant structured information, which are defined in a sophisticated and comprehensive BIM model. The 3D geometry of the model can be used to overlay on the physical environment to provide an ambient intelligent environment for facility maintenance engineers (Gheisari, 2013)[6], especially in case the real object is invisible behind a wall or ceiling.

1.2 Objective of the thesis

The objective of this thesis project is to explore the feasibility to build a practical Augmented Reality solution for building facility maintenance scenario, with BIM technology. Combine BIM and AR technology together to design a practical solution is challenging, because most of the previous works either focus on specific type of user scenario or with various kind of limitation or compromise. This is mainly because AR technology as an emerging technology is still under rapid development.

One of the challenges is about the target recognition. The AR target recognition technology has been advanced marker-based to marker-less, and in facility management scenario, marker-less recognition & tracking on room-scale target is needed, rather than markers, because the maintenance engineers would naturally turn the camera toward the actual object for the maintenance task. Although markers can be attached on the equipment, on the wall or celling, but in practice, attaching and spreading markers everywhere in the building is not considered as a good idea, not only because it doesn't look elegant, but also its effort-taking to manage and maintain.

However, marker-less target recognition is complex to prepare because users need to somehow get the feature points of the physical environment, and the AR recognition quality is very much based on the quality of the feature points. Although with some AR tools, it is possible to generate reliable marker-less AR target, but the practicability is a great challenge on this topic, because it is obviously not practical to ask building facility engineers to maintain AR targets by programming.

Another challenge is that the virtual content in facility management is usually quite large, which can cover the area of the whole floor. Because in practice of an AECO project, the virtual content to be augmented can be the HVAC model of the whole floor. When AR is initialized, the model is visualized, the maintenance engineers might need to walk along the HVAC pipes without losing tracking. That requires not only object recognition but also motion tracking. Some new AR engines in recent years, such as Google ARCore or Apple ARKit support motion tracking by using the device sensors (gyroscope and accelerometer) to calculate its pose (position and rotation) in respect to some starting point, whereas the starting point can be the place where object recognition happens. Therefore, in our context, object reorganization and motion tracking must be integrated in the way that object recognition can be used to initialize the scene, and motion tracking can be employed as the extended tracking method for making sure that the model stay as it is, no matter how user move in the floor. How they can be perfectly combined is a challenge.

There are some other challenges such as the maintainability and shareability of the AR targets, the vision-based indoor navigation support and BIM integration. And finally, and most importantly, in this thesis project, we implemented a systematic solution architecture for supporting the whole workflow of three phases, including 1). the AR configuration phase for making a physical environment AR-ready. 2) the alignment phase for align the AR target and the virtual content. And 3). The AR viewing phase for the final users, which are the maintenance engineers.

1.3 Structure of the thesis

The thesis is structured according to the process of the project. Chapter two describes the background information, related works and challenges details. Chapter three describes how we start the project by doing user studies, following User-centered-design process. Chapter four describes the technical evaluation, which was executed in parallel with the user study. Chapter five describes the concept design based on the work of user-study and technical evaluation. Chapter six describes how the system is implemented in detail. Chapter seven describes the evaluation process and result and in the last chapter, we discussed about possible future improvements.

2 Background & Challenges

2.1 BIM technology

BIM technology are considered as crucial for AECO industry because they offer three useful characteristics (Wagner and Schmalstieg, 2003)[3]: (1) they store the digital information of building objects in databases. (2) the information is linked in the database so that the modification one object and trigger the update on all the relevant objects. (3) the information in the database can be accessed by other industry specific applications, such as the building information system (BIS).

There are also some other benefits in practice of BIM (Gheisari M. and Irizary J. 2014)0: (1) It makes the information processing faster and more effective. (2) It makes it easy for designers to achieve better design through comprehensive analysis, simulation and evaluation. (3) It takes care of the cost and environmental data of the whole life cycle of the building. (4) It makes the assembly of structure system with digital product data automatic. (5) It makes it possible to achieved better user experience with 3D visualization. (6) The data of the whole life cycle can be accessed for facility management.

2.2 BIM for Facility Management

The Digital Twin concept, on the other hand, is considered as a major tool for increasing productivity in the age of industrial digitalization in the engineering field. (Deuter A. and Pethig F. 2019)[14]. The concept was originally established by NASA, which refers to a simulation model that mapped the behavior of a physical spacecraft (Shafto et al 2010)[15] and then was employed also in industrial domain, in which it referred to a virtual representation of a physical product in the product lifecycle. (Grieve et al, 2011)[16]

The formal hypotheses of digital twin were proposed and discussed with industry professional forum held by engineering association OWL Maschinenbau e.V. in June 2018 and at the PLM Europe conference in October 2018 as following: (Deuter A. and Pethig F. 2019)[14]

1. A digital twin is a digital representation of an asset.
2. A digital twin is in several places simultaneously.
3. A digital twin has multiple states.
4. The digital twin has a context-specific state in a specific interaction situation.
5. The information model for digital twins is a real information model, which is infinitely large.
6. The real information model can be finitely approximated for a specific scenario, becoming a rational information model.
7. The rational information model can be stored in multiple places.
8. The rational information model can only be partially visible.

BIM technology provides a practical solution to enable the digital twin concept. In the above definition point out clearly that the ideal information model of digital twin is infinitely large, which is not possible. With BIM technology, the infinite real information model can be narrow down and structured and defined as BIM properties, which is the final rational information model.

2.3 Augmented Reality, BIM, FM and Digital Twin

Augmented Reality (AR) is an emerging technology that provides an interactive experience of a real-world environment where the physical objects in the real-world are overlaid with computer-generated digital data, which can be in the form of text, pictures, videos, 3D assets, or a combination of all the above. The digital data can be constructive (i.e. additive to the real environment) or destructive (i.e. masking the real environment) (Rosenberg, 1992)[17].

In the scenario of facility management, facility maintenance engineers are often needed to relate physical objects to extra information from either a printed material or digital system (e.g. a mobile phone or laptop screen), and they usually must shift their focus among those objects. (Gheisari et al)[2]. This makes AR a perfect user assistant solution for facility maintenance engineers to perform their tasks in an intuitive way. As Henrysson and Ollila (Henrysson and Ollila, 2004)[4] pointed out in their study on Mobile Augmented Reality (MAR) that AR can help solve the problem of "distracting domain switching" because AR makes it possible to directly overlay static or dynamic information on the physical object. The overlaid information can be from its BIM properties or other sources such as facility management system or building information system (BIS).

In practice, a simple AR application for facility management can be a portable application, running on hand-held mobile devices (Irizarry et al., 2012)[5], or AR head-mounted display (HMD). The application must be able to recognize the target object in the maintenance task and overlay the relevant structured information, which are defined in a sophisticated and comprehensive BIM model. The 3D geometry of the model can be used to overlay on the physical environment to provide an ambient intelligent environment for facility maintenance engineers (Gheisari, 2013)[6], especially in case the real object is invisible behind a wall or ceiling. In addition to the static information from the BIM model, dynamic information can also be integrated from facility management system or building information system. This information is beneficial for maintenance task. e.g. A maintenance manager might want to check the temperature of the Air-handling-unit (AHU) before making maintenance decision.

2.4 The Challenges

Although integrating AR and BIM for facility management is beneficial for facility maintenance engineers. It is still a challenging job to combine all these technologies together as a practical and reliable solution, which is the target of this project. The difficulties are from four different aspects.

2.4.1 Combining the design process and proper technology to build a practical solution, instead of a fancy prototype

The main challenge of this project is how we can combine the design and technology for making a practical solution for solving real problems, instead of only a fancy prototype. This is hard because it is not like most of the software project, as Augmented Reality is yet an emerging technology, it is not even a popular technology that everybody is using in daily life. Therefore, on one side, when doing the design following the User-Centered-Design or UCD process, asking users what they want with AR, we may get various kind of ideas but many of those are fancy but beyond feasibility, when it comes to technical implementation. On the other side, when performing technical solution design, we may realize that a practical AR solution requires users to perform some extra activities which they are not aware of at all, such as the activity to configure an environment to be 'AR-ready', or as known as 'AR configuration'. When talking about AR during the interviews, people may expect to open the camera and objects can be visualized at any position as they want, without any configuration beforehand, which is not feasible at this moment, at least. And it's almost impossible to get such information as 'requirement' from user interviews, just like it's impossible to ask people what they expect for a car in 1850s when everyone was still using horse-pulled carriages. Therefore, we must customize the standard UCD process, inserting technical evaluation activity and allocate most of the time before the implementation phase. In addition, considering that the goal is to implement a minimal product, five-month times is very limited, we must prioritize the time on the most critical features, and move others to the longer-term product roadmap.

2.4.2 Technical Challenge of AR Technology

2.4.2.1 Marker-based V.S. Marker-less

The maturity, specifically the accuracy and reliability of AR engine is the biggest challenge. AR technology has been developed for many years since 1999, when the first popular AR engine *'ARToolkit'* was released (Kato and Billinghurst, 1999)[18]. The supported tracking target has been expanded from AR markers or simple images, to physical small size objects such as a cup, and room-scale objects such a corner of a room. The recognition and tracking of markers or images are the simplest and but still the most robust and reliable option. Koch et al[19] already demonstrated a prototype using natural markers(images) in 2012. Today we can see many exiting research or projects following this approach. However, in the scenario of facility management, the recognition and tracking of room-scale object is preferred, rather than markers, because the maintenance engineers would naturally turn the camera toward the actual object for the maintenance task. Although markers can be attached on the equipment, on the wall or celling, but in practice, attaching and spreading markers everywhere in the building is not considered as a good idea, not only because it doesn't look elegant, but also its effort-taking to manage and maintain.

Object (either small-size or room-scale) recognition and tracking can be achieved by retrieving the 'point-cloud' which is a group of 'feature points', which are the visually distinct features in the captured camera image of the target object. There are several AR engines support it. e.g. *Vuforia* from *PTC*[20] support small object tracking, *ALVAR* from *VTT*[19] or *ARCore* from *Google*[21] support room-scale target. The key challenge of object recognition and tracking is about the generation of feature point. The point-cloud of a physical object is like the 'digital twin' of the target object in geometry. Usually the more accurate point-cloud lead to a robust tracking performance. On the other side, the complexity of the workflow of point-cloud generation varies a lot when using different AR engine. Therefore, how to generate a high-quality point-cloud efficiently is the major challenge. And for those who is building a practical solution, this is the main point to consider when selecting AR engine for the solution.

2.4.2.2 Motion Tracking, sensor drift and re-alignment

In the scenario of facility management for HVAC facilities, the HVAC pipes are usually quite long, usually when the location is recognized and pipe model is visualized, the maintenance engineers need to walk along the pipe without losing tracking. This requires the solution to employ not only object recognition but also motion tracking. Motion tracking technology can use the device sensors (gyroscope and accelerometer) to calculate its pose (position and rotation) in respect to some starting point, whereas the starting point can be the place where object recognition happens. In our solution, object reorganization must be integrated together, object recognition can use to initialize the scene, and motion tracking can be employed as the extended tracking method for the overlaid virtual object. How they can be perfectly combined is a challenge. In addition, as the motion tracking on mobile devices (phones, tablets) is fully relay on visual and sensor (gyroscope and accelerometer) data, things can get tricky due to the high framerate of the sensors, which causes small measurement errors to accumulate to a significant error over time, which is called 'sensor drift'. As a result of which, in practice, especially inside a large building, when the maintenance engineer holds the device and walk far away from the location when scene was recognized, the augmented HVAC model (e.g. the pipes) could significantly drift away. It is mandatory to have a solution to enable the user to re-align the augmented model correctly.

2.4.2.3 The Maintainability and Shareability of the Tracking Target

No matter the tracking target is a marker, image or point-cloud. It is important to make it easy to maintain. In practice, on one side, the maintenance engineer could have the need to update the marker, image or point-cloud. On the other side, it could also happen that the virtual object (i.e. the model) to be augmented need to be updated and re-aligned with the tracking target. The prototype in previous study are mainly for dedicated area with not so many tracking targets. But in practice this is a mandatory issue to be tackled.

Additionally, another practical issue is that usually there are multiple engineers responsible for one building, at least there can have multiple devices available for them to use. The sharing of the tracking target should be considered as part of the target management solution, so that when one engineer performs some change to the tracking target on one device, the change can be easily migrated to or even automatically applied on other devices.

2.4.2.4 Indoor Navigation Support is a Separate & Challenging Topic

In fact, indoor navigation is totally a separate topic. But in the scenario of facility management, indoor navigation is beneficial for maintenance engineers, especially when finding the target facility. The mature indoor navigation solutions are usually based on extra sensor data such as the signal strength of WIFI, Bluetooth or geomagnetic field information. But the main limitation of those solution is that they require installation of extra, physical sensors all over the building, whereas indoor navigation based on augmented reality technology along with a computer vision-based approach do not have such prerequisite, and there has been a hot research topic in recent years. In this project, indoor navigation is not the focus, but the extendibility for indoor navigation feature must be considered when designing the architecture of the solution.

2.4.3 The Technical Challenge of BIM Integration

BIM model can be used not only for visualization but also for showing component properties. But there are two practical problem to be tackled.

1) The BIM model needs to be managed systematically and preferably centrally. In practice, the BIM model can be changed by the model designer, and once that happen, the solution should support to apply the change easily to all the devices for all the maintenance engineers, not only that, on the other hand, the same model might also be used by other system, such as the web-based or desktop-based applications.

2) The size of the BIM model in AECO field is usually quite large, whereas the augmented reality devices are usually mobile, the hardware capability of which, especially the graphical

processor is not as powerful as the desktop computer which is used by the model designer. Therefore, the solution needs to enable model optimization for AR devices to some extent.

2.4.4 Lacking a systematic architecture for supporting the whole workflow

It might be not hard to make a simple prototype, which works for dedicated room or building. However, a practical solution should include tools for supporting the whole workflow, including (1) the tool for configurating the tracking target, (2) tool for the alignment between the tracking target and the virtual object (BIM model) and even model optimization, and (3) tool for the end user for visualizing the virtual object and supporting other type of interaction. And a systematic architecture is needed to not only enable the workflow, but also with good extendibility for the future.

2.5 Contribution

The target of this project is to find out a practical solution to apply augmented reality for facility management. There is no perfect solution available in the market, so far. For achieving the target, we start the study by following the User-Centered Design (UCD) process. In detail, we started by taking interviews with the target users and relevant professionals, then we summarize the use-case and define the personas, based on the use-case we define the detail feature list, and based on the personas we made the user interface prototype.

On the other side, for the technical solution, we took the method of 'review of existing relevant technology and product' to evaluate and select technical tools, based on which, we design the solution architecture and implement the working software, after which we perform evaluation for getting user feedbacks, based on which, we draw the conclusion and discussed the future works.

3 User Study & Requirement Analysis

This section describes the whole process for how the interaction design is done through a comprehensive process step by steps, from the interview to the final concept design.

3.1 User Study Following UCD Process

To find out the real use case and requirement, we make a comprehensive user study following the User-Centered Design process. The User-Centered Design, or UCD process is also called Human-Centered Design process, which is a systematic process that employs scientifically proven methodologies of the behavioral and cognitive sciences to optimize the user interaction design for achieving the best user experience (UX). (Cowen et al)[24] The UCD process includes four steps:

- **Step 1. Analysis:** The first step is for analyzing the user and requirement, it has two sub components.
  1) Specify the user group: Identify the people who will use the product.
  2) Specify the use case and requirement: Identify how they will use it in detail. In this step we will employ the method of Persona to create some virtual users to simulate the actual use case.
- **Step 2. Design:** The user interaction prototype will be designed according to the use case and requirement from analysis phase (step 1).
- **Step 3. Evaluation:** The design prototype should be evaluated with the actual users. Feedback should be collected, and this step can iterate back to the design phase (step 2).
- **Step 4. Implementation:** After the design is concluded, the implementation work can be started.

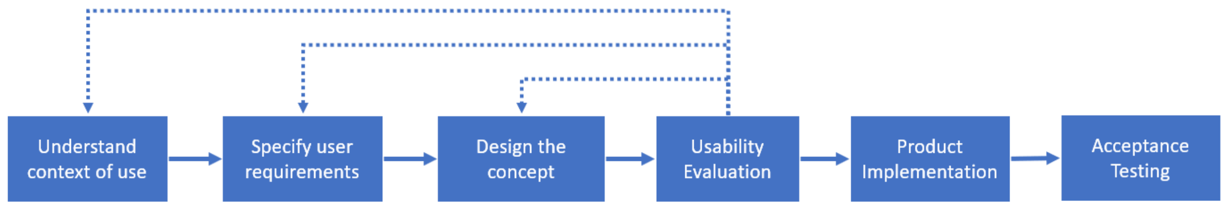

Figure 1. The standard UCD process

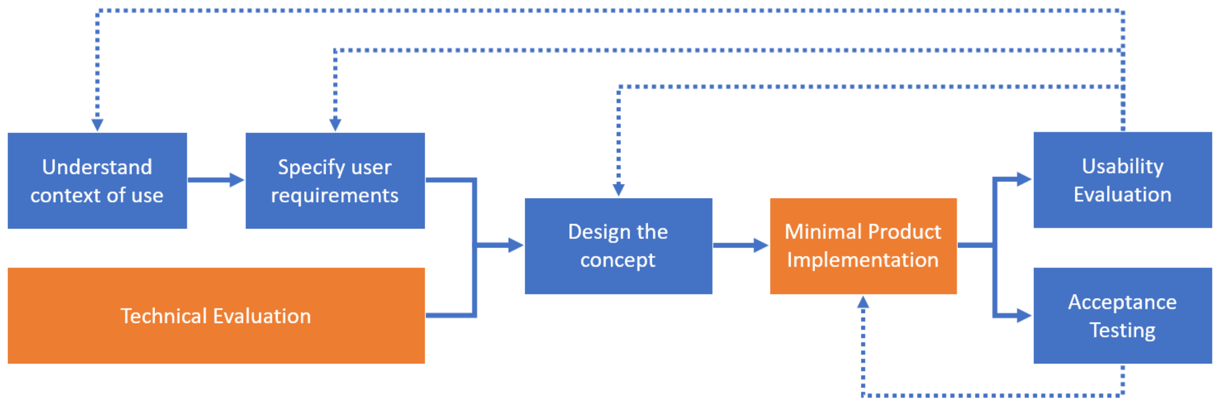

Figure 2. The customized UCD process in this project

In practice, the UCD process can be incorporated into higher level product development approach such as waterfall and agile, and it can be customized according to the project needs. The principle of the process is to involve users throughout the design and development process. However, in this project, we customize the UCD process and put the evaluation phase after the implementation phase, because of two reasons:

1) The standard UCD process is usually for UI design based on mature technology. In this project however, we are building a solution based on emerging technology (AR). In this case, the investigation of technical tools including the AR engine and the review of existing product takes majority of the time in the pre-study phase. Because the result may have great impact on the feasibility of the design. Therefore, in our customized process, the technical evaluation activity is done in parallel of design activities such as user study and requirement analysis. The design concept should be done based on the result of both design track and technical evaluation track.
2) For the prototype, we must implement a minimal working software rather than a user interaction mock-up, within a limited time of six months (Mar-1/2019 - Aug-31/2019). And the time period of evaluation for the UI prototype conflict with the summer holiday period in Finland and it must be postponed.

3.2 Interview

For the analysis step, we start with five round of interview with the target users and relevant professionals. The goal of the interview is for improving the understanding of user behavior to influence the interaction design. In total, six people participate the interview and one out of six is female. All the interviewee has the background of facility management and from relevant companies in Finland including *Granlund* Oy (AECO consultancy), *ISS Palvelut* (property and office service provider), and *VimAI* (indoor augmented intelligence solution). Among the interviewees, three out of six have strong background of BIM, and four out of six has experience of AR while two out of six have well understanding of AR.

The main goal of the interview is for understanding the problem or challenging in the current workflow of facility maintenance. According to the UCD method (Gulliksen et al, 1999)[25], although the final target is to collect ideas for AR solution, in the beginning, the questions should be asked only for understanding how the current workflow works and the problem or challenging during the work. Their opinion or idea about apply AR can be asked in the second part of the interview. A question list was prepared for the interview. The listed questions are categorized in three parts. Part of the typical questions are list below:

1. **Questions about the current situation:**
    - What do the maintenance engineers usually do during the maintenance work? Can you describe one or a few typical maintenance tasks in detail?
    - What is the major inconvenience/challenges during the maintenance work? And how did you overcome them? (please give some example of 'typical hard moment')
2. **What would you do in case a perfect AR glasses is available for maintenance work:**
    - How helpful would it be to visualize target facility models in AR, when they are invisible physically?
    - How helpful would it be to show the relevant BIM properties of the target facility model?
    - Which BIM properties are the most useful ones? What would you do with that information?
    - What kind of interaction would you expect to do with the visualized facility models in AR?

- What else activities would you like to do in the AR view. (e.g. object measurement)
- What kind of concern do you have for applying AR for facility management? (for those who have experience or understanding of AR)

3. **Please rate for the below ideas for how true they are, in the maintenance scenario** (1-5, 1: not true at all, 5: completely true)
    - As a maintenance engineer, when arriving the building, you want to be able to easily find or guided through indoor navigation (if possible) to the maintenance target facility.
    - As a maintenance engineer, when you reach the target position, when you are not sure of what 's going on, you would like to check and confirm the facility object information (from BIM properties)
    - (continued from the last item) When the static information is not enough, you would like to check and confirm the instant status information of the facility object.
    - As a fresh maintenance engineer, a step-by-step AR instruction for the maintenance task would be helpful (step 1 do this, and step 2 do that..., etc.).
    - As a maintenance engineer, sometimes you would like to place notes on the facility object, or in the areas around it, for you or the next maintenance engineer to take as reference later.

3.2.1 Highlights from Interview Record

1. Maintainability/Updatability:
    - It would be the perfect if the AR feature doesn't require any preparation or configuration, but if that's not achievable, the configuration should be easy to setup and easy to maintain or update.
2. Navigation:
    - It would be the perfect if the AR feature doesn't require any preparation or configuration, but if that's not achievable, the configuration should be easy to setup and easy to maintain or update.
3. Checking object information:
    - Sometimes the engineer would have the need to check the BIM model to understand how the equipment's are connected.
    - The information in the BIM model might not be up-to-date, therefore usually engineers do not relay on the BIM properties only, they usually check from the facility management system.
    - Sometimes the engineer might want to check instant status information for diagnosis, usually they must go to the computer room where they can access the building automation system, or they might call somebody (remote support engineers) to do check it for them.
4. Features that 'good to have'
    - The maintenance engineers sometimes would like to leave a note for the next engineer. As it's not allowed to attach a sticker physically, usually they write notes in the report system, but the notes can be easily ignored. It would be helpful if they can attach virtual notes in AR.
    - If it's possible to enable maintenance engineers to measure the length, area or even volume of the object in AR, that would be quite useful in case the object is hard to reach or difficult to estimate.
    - Remote assistant would be useful.
5. Other notes:

- If the AR device is a hand-held device, maintenance engineers need a pair of touch-screen friendly cut-resistant gloves. If it's AR glasses, the glasses should be integrated with the existing eye-protection safety goggles with industrial safety certifications.

3.3 Breakdown Analysis

After gathering the interview records, we employed the research technique of 'breakdown analysis' to find out the problems. Breakdown analysis is a technique to identify situations where the user does not complete the desired task or activity.[26]

Based on the information collected from the interview, a break-down analysis is performed and below are the break-down cases:

- Problem in navigation: For the maintenance engineers, usually the first thing that they must do when they arrive the building is to locate the maintenance target object inside. They need to know which floor it locates and how to reach there. A breakdown at this moment is that they could spend a lot time on the finding the correct place. Not only in the beginning, indoor navigation is a typical breakdown throughout the whole maintain acne task, as they usually must switch among different places inside the building, for example switching between the control room, where they have access to the computer, and the object under maintenance or problematic area. It is a problem until one day they get familiar with the building.
- Inconvenient to check BIM properties: It is quite often that the maintenance engineer need to check objects' properties, which are defined in BIM model. However, usually they need to walk to the control room, or computer room to access the computer, or sometimes they even must call the support engineer for help.
- The BIM properties are not up-to-date. Even when engineers have easy access to the BIM properties, they might hardly relay on it without confirming the correctness, as it could be not up-to-date. Usually they confirm it by accessing other facility management system or calling the support engineer.
- Dynamic information is needed for diagnosis. During the maintenance work, usually the engineers need not only the static information from BIM properties, but also dynamic information, such as instant status or data from the sensors, which can be accessed through building information system, or BIS. They usually retrieve this information by accessing the control room or calling the support engineer.

3.4 Personas

In UCD process, 'personas' are a means of summarizing a user segment targeted by a product development effort as a fictional character. They are helpful to simplify the vast amount of information from the interview and streamline the design process. Personas are usually summarized from the interview record, it represents the type if user in detail, including who they are and how they behave, especially how they handle the current job. Usually, multiple persona can be defined, and they can be used in the user scenarios stories. In this project, we summarized and defined three personas for three type of users: A senior maintenance engineer, a junior maintenance engineer and a support engineer who works remotely.

Persona A

Janne

47 years-old

Senior Maintainance Engineer, JTT

Janne has been working in company 'JTT' for 20 years, JTT is providing building maintainance service in Helsinki area. Janne knows well of the building and he can easily locate any facility inside. He is also very professional that usually he can find out the root cause of any problem easily.

Persona B

Tommi

25 years-old

Junior Maintainance Engineer, JTT

Tommi joined JTT six months ago and Janne is his mentor. Sometimes Janne take him along during routine maintainance tasks and teach him not only about the task but also the story of the buildings and everything that happens through out the years.

Persona C

Juhanna

34 years-old

Manager and Support Engineer, JTT

Juhanna has been working in JTT for 7 years. She is the operation manager for building maintenance of Helsinki area, responsible for assigning and managing the tasks, as well as giving support to the maintenance engineers remotely.

Figure 3. Personas

3.5 User Scenario

Use scenario is like a tiny one-act play, subdivided into one-paragraph micro scenes (interaction points) that describe how users currently perform one or more tasks.[27]

We summarize user scenario as tasks[28], each task can be considered as a maintenance task that can happen in the real life, and it could involve multiple person, which are defined in 'Persona' phase.

Task 1:

Janne as a senior maintenance engineer received a challenging task, he is asked to go to the building located in 'Malminkaari 21, Helsinki, Finland' for fault investigation. It is reported that some hot and cold spot are found throughout the space in the second floor of the building. Janne's task is to check the situation and find out the root cause.

- **Event 1**: Since Janne is already familiar with the building, when he arrived, he directly went to the second floor and checked one of the reported cold spots. He found the temperature is indeed noticeably lower than other rooms. Then he checked the 2D floor plan in his hand, trying to find out the pipe location in this area. But as the room is furnished so differently than the shape on the 2D floor plan, it is a bit difficult for him to alight the position of the pipe precisely. But he still roughly locates the pipe, although it is invisible behind the ceiling.
- **Event 2**: Now, Janne would like to check which Air Handling Unit (AHU) is connect to this room. For that, he must go to this build's control room to check from the computers there, although he has a printed version of the HVAC design diagram from the maintenance handbook, but he ignores it because the information could be out-of-date.
- **Event 3**: Finally, from the computer in the control room, Janne found the connected AHU. He has a hypothesis that it's caused by the blower motor failure, but he is not sure. He would like to check the instant temperature and other information from the sensors. For some buildings which has building information system, or BIS integrated, he can check it in the control room. But this time, as most of the times, he called his colleague Juhanna, who works as a support engineer and can check this information for him remotely. After a few discussions on phone, Janne confirmed that his hypothesis is correct that the blower motor needs to replace.

Task 2:

Tommi is assigned a task for regular filter maintained for the AHU in the building located in 'Malminkaari 21, Helsinki, Finland'.

- **Event 1**: The task is just a simple routine maintenance check, which is usually assigned to junior engineers like Tommi. As he has never been to this building.
- **Event 2**: Although it's a routine maintenance check, Tommi still spend much time on the handbook for the operation steps.
- **Event 3:** During the last task, Tommi noticed that the machine is having a weird sound, but it seems everything are all right. To make it easier for the next maintenance engineer to understand the situation, he decided to leave a note in the physical environment but so far there is no obvious problem found, please keep an eye on it.

3.6 Brainstorming

A brainstorming session was executed to generate ideas for solving the problem defined in the two tasks mentioned in user scenario chapter. For task #1, the general idea is to enable Janne to be able to see the virtual facility directly in AR, and access to all the information needed within the AR view, so that he doesn't have to switch to the control room and back and force. The general solution for task #2 is providing AR-based indoor navigation to help engineers who is not familiar with the target building, and AR can also provide a step-by-step visual guidance for the detail maintenance task.

In parallel of the internal brainstorming session, my college in *Granlund* Juha Tuomainen launched an innovation campaign among all *Granlund* employees to collect ideas on how AR technology can be beneficial for facility management. Here are some of the ingesting ideas collected from the campaign. Although it is pleasant that 21 ideas are collected from the survey, but many of those are either too general or too ideal and even beyond feasibility. That is because AR is an emergent technology and many of the participant do have enough knowledge of AR, therefore, most of the people are not sure of what is possible and what is not. For example, when asking people, the question 'how would you imagine AR could be beneficial for you as a maintenance engineer?', most of the people would expect that they open the camera and the facilities behind walls or ceilings can be visible, no matter from which position, and without any configuration or preparation, since the model is ready. However, the reality is that it is not possible for AR camera to recognize an environment only based on its 3D model, without configuration in the physical environment.

This kind of feasibility knowledge gap problem exist not only for the ideas that collected from the campaign, but also our own idea from the brainstorming session. We realized that we cannot simply treat this project as a design project, following the UCD process. Instead, we must perform a comprehensive technical investigation before the concept design, then process with UI prototype, solution design, implementation and evaluation.

4 Technical Solution Investigation & Selection

One thing special in this project is that the we are building an AR solution, whereas AR technology is an emerging technology comparing to the software development based on traditional technology such as web applications, desktop applications or mobile applications. Therefore, the most challenging work in this project is not only designing the UI interaction following UCD process, but also technically make it happen. In another word, building the solution with proper technology and tools, making sure that it is not only a fancy prototype but a practical product which can be used by real users. To achieve that, it is mandatory for us to perform a systematic investigation on AR technology for choosing the proper tools to build our solution.

We employed the research method of 'reviewing existing solution', spend 70% of the time during the pre-study phase on the investigation of existing software and hardware, for picking the best tools, based on which, we design our own solution. In practice, the UI interaction design following UCD process is executed in parallel to the technical solution investigation, because sometimes the concept design needs to be adjusted according to the feasibility from what we understood from the technical investigation. The technical investigation is done for three different aspects: Augmented reality, model management and system integration.

4.1 Technical Investigation for Augmented Reality

Building an AR solution is a complex task, let's split the investigation on three different topics: The AR target recognition and tracking, motion tracking and cloud-base AR solution.

4.1.1 AR Target Recognition & Tracking

Based on our context, the most basic requirement is to visualize the facilities such as HVAC equipment's behind the wall or ceiling. For such particular purpose, the relative technology in AR is 'target recognition and tracking', which means that the application should be able to recognize the physical target object, so that it can overlay virtual objects related to its pose (position and rotation). There are mainly three type of AR solution, categorized by how target is recognized. They are 1) Marker-based AR, 2) Marker-less AR and 3) Location-based AR.

4.1.1.1 Marker-based Target Recognition & Tracking

Marker-based AR is the most popular and simple form of Augmented Reality today. It requires a visual marker or an image that you can find in real life. The black and white marker below is an example of markers, supported by the well-known, early AR engine: *ARToolkit* (Kato and Billinghurst, 1999)[18]. For marker-based AR, computer vision and image processing are the key technology behind, when image is recognized, the pose of the image is also calculated, so that the pose of the virtual object to be augmented can be adjusted accordingly. That is why for AR engines, especially the ones in early days, the black-and-white markers are considered as the easiest and most reliable AR target. However sometimes it is considered not practical or not elegant to put markers everywhere. Images in real life can also be used as markers. But not all images work perfect. The recognition and tracking quality very much depends on the mount and quality of the 'features' on the image. Some AR engines such as *Vuforia* from *PTC*[20] provides tools or service that can calculate a score (e.g. from one star to five star), showing how well the image can be used as a marker, which makes it easy for developers to choose the proper images.

Figure 4. AR Marker (*ARToolkit*)

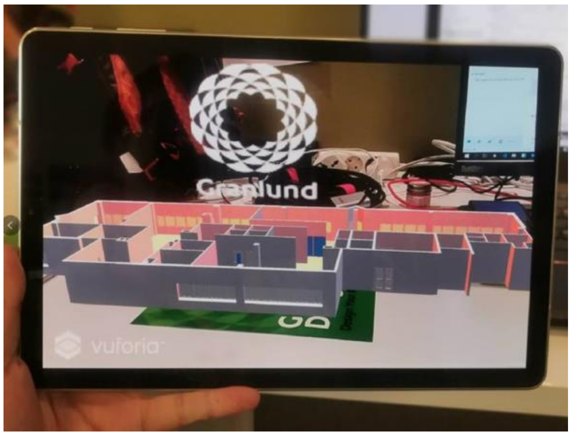

Figure 5. Image as Marker (*Vuforia*)

In the context of AEC and facility maintenance, there are many previous studies about using images as AR markers. The images can be a physical wallpaper, poster on the wall or signs inside the build. Koch et al (Koch C., Neges H-M., 2012)[29] discussed about the possibility of using signs as markers, such as the sign of exit. We also considered to use the sign of fire extinguisher, which sounds like a good option, because usually according to the safety policy, it is mandatory to distribute portable fire extinguisher in every area. For example, according to Occupational Safety and Health Administration (OSHA of USA), "The employer shall distribute portable fire extinguishers for use by employees on Class A fires so that the travel distance for employees to any extinguisher is 75 feet (22.9 m) or less." (OSHA 1910.157.d.2)[30]. However, a practical problem for using signs is that they are all the same. This is not a problem when the goal is to show some content for the sign itself, such as a few text for describing the policy or guidance of the fire extinguisher, but if the goal is to visualize large facilities across the whole floor, such as the long HVAC pipes in our context, the signs has to be unique, because for different signs, when the HVAC pipes are visualized, the pose(position and rotation) should be different, although the sign is the same. This makes it hard to simply take natural signs as marks, although one workaround solution can be attaching more signs (such as QR code) around the fire extinguisher sign, but that is again considered not an elegant solution.

Figure 6. Natural sign as markers

4.1.1.2 Marker-less Target Recognition & Tracking

In recent years, the development of AR has led to 'Marker-less AR', which allows the use of a physical object or physical environment as the recognition and tracking target. In this case, for generating the target, it is required to scan the physical object or environment to find the 'point cloud', which is a group of feature points. The process is also known as 'map construction'. It works by finding the same feature points in multiple images (either from multiple pictures or multiple camera frames) from different viewpoints and estimating the 3D structure of scene by triangulating those feature points.

In this project, *Vuforia* and *ALVAR* are evaluated as marker-less AR engine. *Vuforia* from *PTC* support 'object target', which is for small size objects, such as a cup. The feature points of the object target can be generated by scanning around the object while the object is on a scanning template paper, which has a lot of features and is used to establish the position of the object target relative to its local origin. However, this is not an option for facility management context because we need to recognize not small-size object but room-scale objects, such as a corner of the office room.

Figure 7. Object tracking target (*Vuforia*)

ALVAR from *VTT*[33] supports to track room-scale point cloud targets. To generate a point-cloud resource, it is required to take 30 to 100 images of the target or the environment. It is important to make sure to have a good baseline between the images and that the images cover most of the target or environment from many different angles, perspectives and distances. Those images need to be imported into point cloud generation software like *Agisoft Metashape*, before the point-cloud can be used by *ALVAR* engine, some post work is needed, such as converting the point cloud to be *ALVAR* compatible format, manually adjusting its orientation, calculating and changing the scale to be metric-scale. In this solution, the AR tracking quality is very much depending on the quality of the point-cloud. But from the practicability's point of view, the whole process is hard to be practical. Because of two reasons. 1). In the picture taking step, the maintenance engineer needs to take many pictures for one corner, and some skills are required to make sure of the quality. There is a workaround solution, which can save the time-cost of picture taking. That is using panorama camera (360-degree camera). Taking panorama images is easy, and one panorama image can be extracted into multiple normal image. But it brings another problem that usually the panorama images have less distortion and the quality is not as good as the normal image. 2). The post-works are complicated with a lot of manual steps. For example, for modifying the scale to metric scale, the maintenance engineer needs to calculate a scale factor. He/she need to find an object inside the point-cloud, which is obvious enough from its shape. Then the size inside the point-cloud needs to be measured and compared with the actual size in physical world, thus the scale factor can be calculated. Those two steps are too complicated to be handled by maintenance engineers in practice. Some level of automation tools needs to be developed to make it practical.

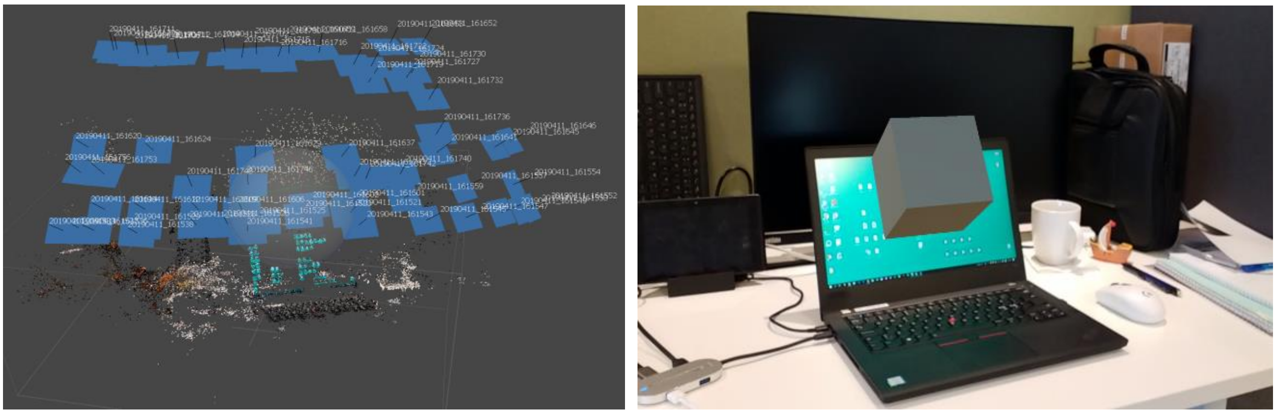

Figure 8. Point-cloud recognition & tracking (*ALVAR*)

'Anchor' is another important concept related to marker-less AR and point-cloud. Point-cloud is only a group of feature points of specific physical environment. An anchor is a point with its own pose related to the point-cloud. Anchor is for enabling user to simply anchor virtual objects to the real environment. This ensures that the virtual objects stay glued to the physical location where it was put. In addition, it enables objects stay in place even if the system learns more about the environment over time. This feature is provided by all the modern AR engines such as *Google ARCore*, Apple *ARKit* and *Microsoft* HoloLens. But usually the complexity is hidden from the users, users only knows that they can place virtual object in the real environment, the objects are related to the created anchors.

4.1.1.3 Location-based Tracking

Besides the marker-based and marker-less AR, location-based AR is another type of augmented reality, which overlays virtual content to a specific location. For example, users may see virtual road sign displaying the street name through the phone's camera when walking in a city they are not familiar with, or another example, the well-known AR game ' *Pokemon Go'*[32]. This solution relies on a combination of GPS and the compass sensor on the phone. In AEC context however, it is considered a good choice for out-door, construction phase, but not for indoor, maintenance phase. For example, the field workers can preview the architecture in the real place before or during the construction. One of the exiting products is *Trimble SiteVision*[31], which is for exact the same purpose.

Figure 9. Location-based tracking (*Trimble SiteVision*)

4.1.1.4 Summary

In the scenario AEC environment, marker-based AR can be considered as a good option for the construction phase, because in the construction field, the gray, plain floor and walls do not have enough feature for AR to recognize, and the environment changes every day in the construction field. However, in the scenario of facility management, which is in maintenance phase, markers can be used but cannot be considered as a common solution for AR recognition & tracking, because usually it is not practical to place markers everywhere in running buildings. Instead, it can be considered as a complementary workaround solution for some situation when it's hard to use marker-less target. The marker-less AR is the best choice in this situation. Additionally, the location-based AR is good for out-door environment such as the out-door construction field.

AR Type/Scenario	Construction phase (out door)	Construction phase (in door)	Maintenance phase (in door)
Marker-based AR		X	
Marker-less AR			X
Location-based AR	X		

4.1.2 Motion Tracking

For achieving a comprehensive AR solution for facility maintenance, especially for HVAC equipment maintenance, having AR target recognition is not enough, which is only the solution for initialization. Because the HVAC pipes model to be visualized is usually as large as floor scale, and the maintenance engineers need to walk along the pipe while keeping the visualization staying in the correction pose. For such case, combing motion tracking is the solution. Motion tracking technology can use the device sensors (gyroscope and accelerometer) to calculate its pose (position and rotation) in respect to some starting point.

ARCore on Android, or *ARKit* on iOS are the most popular AR engine which supports motion tracking. Which means that it can use the device sensors (gyroscope and accelerometer) to calculate its pose (position and rotation) in respect to some starting point in space. In detail, as the hand-held device moves through the world, *ARCore* for example, uses a internal process named 'concurrent odometry and mapping', or COM[22], to understand where the device is relative to the world around it. It detects feature points and uses these points to compute its change in location. The visual information is combined with inertial measurements data from the device's Inertial Measurement Unit (IMU) to estimate the its pose relative to the world over time.

Uimonen et al (Uimonen M. and Hakkarainen M., 2018)[23] demonstrated a prototype using ALVAR engine to recognize the location, then employ *ARCore's* motion tracking capability as 'extended tracking'. In this way, the augmented model can stay its pose as it is recognized when the maintenance engineer moves away, even though the ALVAR already lost tracking on the target.

However, the motion tracking of *ARCore* or *ARKit* is fully relay on visual and sensor (gyroscope and accelerometer) data, things can get tricky due to the high framerate of the sensors, which causes small measurement errors to accumulate to a significant error over time, which is called 'sensor drift'. As a result of which, in practice, especially inside a large building, when the maintenance engineer holds the device and walk far away from the location when scene was recognized, the augmented HVAC model (e.g. the pipes) could significantly drift away. It is mandatory to have a solution to enable the user to re-align the augmented model correctly.

The sensor drift problem is not as prominent as on *ARCore* or *ARKit* devices, for those devices with additional depth camera, such as *Google Tango*[35] devices or HoloLens, in order to correct errors caused by sensor drift, they utilize depth sensor to measure the exact position of visual characteristics in the camera picture (a.k.a. Visual Odometry). With the additional hardware it can ensure the correct positioning of these features. However, *Tango* project was unfortunately dropped and replaced with *ARCore* by *Google* in 2017, and as for AR HMD like HoloLens, the hardware and training cost is also something must be considered in practice for facility management, and as the user interaction pattern is very different from traditional hand-held mobile devices, therefore the user interface needs to be designed separately.

4.1.3 Cloud-based AR

Even though we have the solution ready for target recognition and tracking, there is still a practical challenge, which is the management of the AR targets. i.e. How we make sure that AR targets can be easily managed and even shared across different session, or even different devices.

Shading the tracking target information (point-cloud and anchors) across multiple session is a mandatory requirement for a practical AR solution. In the AEC context, that means, as a maintenance engineer, when you place a virtual object in the physical space, the object is expected to stay in the exact same position even after you reboot your application or device. To achieve that, it is needed to somehow save the information of tracking target to somewhere (local or remote), so that it can load it from a local file or a remote server next time. As we already concluded in the paragraph 'AR Target Recognition & Tracking' that we need marker-less AR solution in our context (facility management), the tracking target in our scenario is the point-cloud and anchor in physical environment, such as a corner of an office room. Therefore, we need to find a solution to somehow save the tracking target information to local file system or remote server. In addition, it is also needed to make it possible to load it (from local file system or remote server), and then align the two point-clouds: one point-cloud is the one which is saved in the last round, another point-cloud is the new one instantly retrieved from the camera frames. Once those two point-cloud can be aligned, that means the target is recognized. A popular point-cloud alignment algorithm is 'Iterative closest point', or ICP algorithm, which is widely employed in many AR engines. In the context of facility maintenance, as one building can be maintained by different maintenance engineer and they can have different devices, it's crucial to make it easy for them to share the same tracking target information. In detail that means, if maintenance engineer A and B are using different devices, and when engineer A create a point-cloud and anchors for area 1, the data should be automatically available also for engineer B, or it can be easily transferred (for example exported from A's device then imported to B's device) to engineer B.

Additionally, sharing tracking target information across multiple devices at the same time is also a valuable feature for our context. Based on the interview feedback, usually there are one or two maintenance engineers for one building. Sometimes collaboration cloud be needed among multiple engineers.

There are a few existing solutions which support saving tracking target information to the local file system. *VTT ALVAR* supports loading point-cloud resource as a zip file that contains a tracker configuration file, a point cloud file and either images of the target or pre-computed cache files, which can be generated and saved by the point-cloud preparation process. *Google Tango* supports loading 'Area Description Files' which is the point-cloud and anchors information for a target area. However, in practice, to make the tracking target information easier to be managed, it is preferred to enable saving the tracking target information to a server or cloud, instead of saving to local file system. With *ALVAR* or *Tango*, although we can build our own server for the storage, but it is more straight forward to employ a cloud-based solution, considering the server performance, scalability and server maintenance effort. Apple *ARKit* supports cloud-based anchors, but it works only for iOS devices. *Google ARCore* also has its own cloud solution named *ARCore* cloud anchor, which makes it possible to save the locate anchors to *Google* cloud. However unfortunately, there is a limitation is that it only allows to store anchors for 24 hours, which is totally unacceptable when making a real product. Therefore, many third-party cloud anchor services came out, such as *Immersal* and *VimAI*, both supports storing the tracking target information to their private cloud.

Immersal is a Helsinki-based startup company, providing their own SDK for developers to build AR solution. The solution has a comprehensive process supported by a few tools, including anchor configuration, anchor alignment and anchor recognition. Following the process, at first, developers may take several pictures for the target in a physical environment to generate the point-cloud, then they can create an anchor related the point-cloud. After that, they may save anchors to the cloud server. On the other side, they may download the target tracking data (the point-cloud and anchors) from the cloud and import into software development tool such as Unity, where they can align the anchor to virtual content, and build the final application. In the user time, when the physical target is recognized, the related anchor can be loaded from the cloud and generated, so that the virtual content can be also created related to the pose of the anchor, according to how it was aligned in Unity. Comparing to the previous evaluated process of *ALVAR*, which includes many manual operations, the *Immeral* process is significantly easier and more practical.

Although the *Immersal* solution is not chosen for this project because of the cost consideration. But we learned a lot from the *Immersal* solution.

- It shows how the automation can be done to simplify the AR configuration work. The *Immersal* solution is very much the automatic version of *ALVAR* process with better user experience.
- The *Immersal* solution demonstrated a full AR configuration process, which includes three steps: 1). Collecting environment data, 2). Offline alignment for virtual content, 3). Locating & visualization. This kind of 'scan->align->locate' process is a typical process for AR solution in our context.
- The *Immersal* solution demonstrated a smart solution for achieving indoor navigation. As the alignment is done offline within Unity editor, indoor navigation is implemented with a popular Unity plugin named *'NavMesh'*. *NavMesh* support building path intelligently for a given 3D architecture model. This is considered as a smart way to implement indoor navigation without inventing wheels.

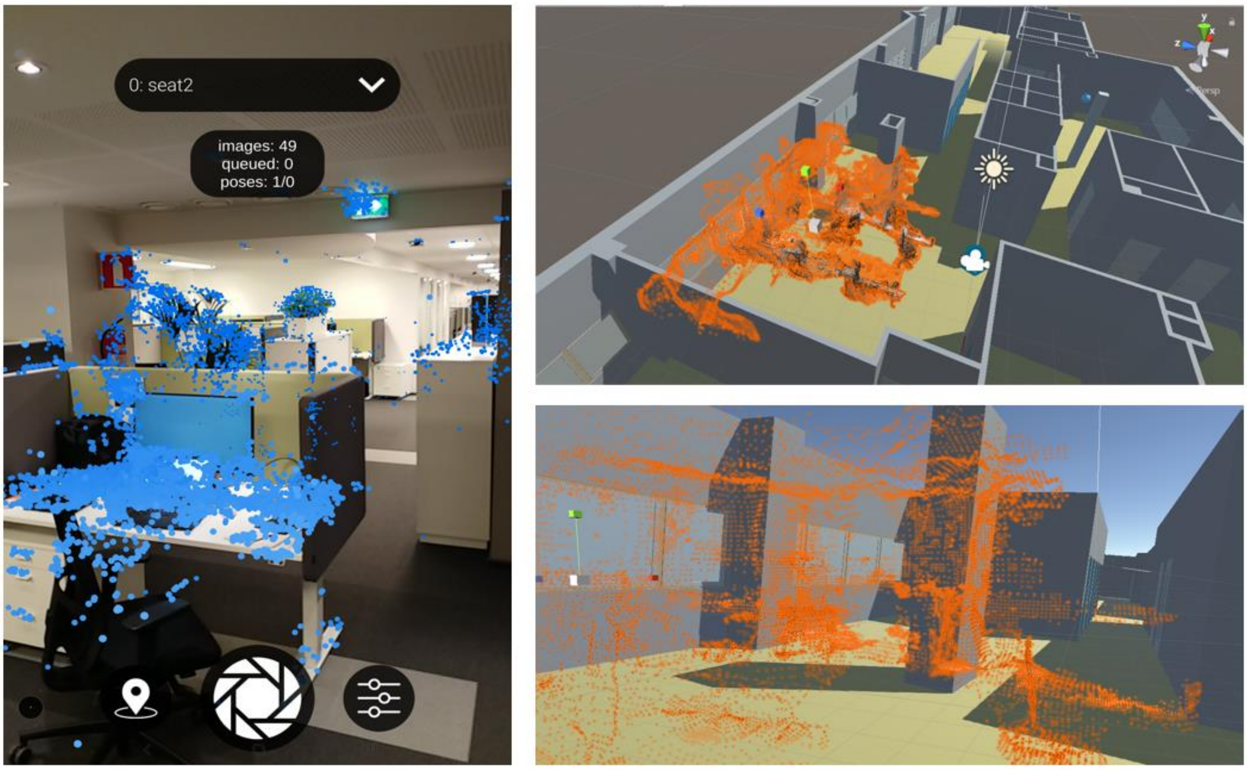

Figure 10. Environment feature point collection & offline alignment (*Immersal*)

VimAI is also a Helsinki-based company, focusing on computer vision-based indoor navigation solution. Although indoor navigation is not the focus of this project, the same technology can be employed for building AR solution. With the tools provided by *VimAI*, users may configure a whole floor of a building to be navigation-ready. The configuration process is easy that users can take a walk in the office while holding the camera, and pictures can be taken automatically, in this way, a 'route' can be configured, and one floor can contain multiple this kind of routes. Then the images can be uploaded to their cloud-based training process for calculating the point-cloud and other related data. In the user time, user can locate him/herself in any position on any route of the floor. Based on our evaluation, the computer vision-based navigation works very accurately, and theoretically based on the accurate positioning, it is possible to add virtual content to achieve Augmented Reality. However, the current *VimAI* solution is mainly designed as a one-stop solution for those who owns or manage the building. There is not SDK available for higher level development, and it's compatible with Apple iOS device only, while we prefer cross-platform SDK, so that we can easily extend it to be compatible with or can be easily migrated to not only phones, tablets but also future potential AR glasses.

4.2 Technical Investigation for Virtual Content Management

Besides the AR technology, it is also critical to discuss how to properly manage the virtual content, and how the AR application can be integrated with other systems.

The management of the virtual content is important. In the case of facility management, the virtual content is the model of HVAC equipment's or other facilities inside the building. Since our AR application should be a common solution, which supposed to be working for any buildings, it is not possible to simply bind the virtual content models into the application because of the number of model can be infinite and the model size is usually quite large in AEC industry. Instead, the model must be somehow retrieved dynamically from the server side, which can also make it easy for users to share the same content. This approach or solution is usually known as cloud-based content management solution, such as *Autodesk Forge*[36] and *Tridify*[37]. This kind of solution makes the life easier for AEC companies from three aspect:

- **Easy to manage models**
  With the cloud-based content management solution, companies can upload their models to the cloud, and access them easily. For example, Autodesk Forge cloud enable users to access the models from all sorts of clients, including web browser or web application, Android application, iOS application and Unity application. In the easiest way, user may use internet browser to access the models, for other clients, user may need to developer their client with the corresponding API. This also enable companies to integrate it with their existing product.

- **Easy to share models**
  Since it is possible access the content through different type of clients. It makes it possible to share content among different type of application or devices. In the context of this project, one model can be shared among the AR application and other applications, such as *Granlund* Manager or *Granlund* Designer. Once the model is updated, it can be reflected automatically on all clients.

- **Model Optimization Support**
  For AEC content, one thing critical is about the model size: The size of the models in AEC context are usually quite large. The cloud service usually includes model optimization. Because the cloud service needs to support different type of clients, their architecture supports a process converting the original model to an optimized format, which can work with all different clients, and different clients can adapt to the different platform to achieve the best performance. For example, Autodesk Forge converts the uploaded model to SVF format from 60 types of formats with Forge model derivative service. The model downloading can be optimized by the corresponding clients.

4.3 Technical Investigation for System Integration

Our final AR solution should be able to not only visualize the geometrical information of the model, but also enable users to see the static and dynamic properties of the model. The static information is from the BIM model, while the dynamic information can be from various kind of sources. For example, usually the BIM model is not changed frequently after it's released. The information can be still edit and changed in *Granlund Designer* and stored in *Granlund Designer* database. In the final AR solution, it is valuable to provide a merged view from both the original properties inside the BIM model and the modified information in the database of Granlund Designer. This integration can be easily achieved through the REST API of *Granlund Designer*. Additionally, the dynamic information such as the instant status of the equipment can be integrated with Building Information System, or BIS. For example, one of the buildings that we experimented is using IoT sensors from Siemens, the instant statistics information of the sensors can be retrieved through the REST API from Siemens BIS.

4.4 Combined Technical Solution Options

During the investigation for technical tools, we evaluated many different options, and the five options as below are considered typical ones as the candidates:

4.4.1 Solution Candidate 1: *ALVAR + ARCore + Forge*

In this solution, *ALVAR* is used as for AR recognition & tracking to initialize the virtual content, and *ARCore* is employed as the extended tracking for the content. The whole process is described as below:

- **Configuration-time**: Maintenance engineers create the point-clouds for the target area by taking pictures and some post-work process with *ALVAR* tools.
- **Authoring-time**: Support engineers align the virtual content such as the HVAC model to the point-cloud of an area.
- **Using-time**: Maintenance engineers can view the virtual content in the physical area.

Advantages:

- The AR recognition & tracking works well if the point-cloud is in high quality. No marker is needed. *ARCore* can work with *ALVAR* for extended tracking.

Disadvantages:

- The point-cloud preparation is laborious and error-prone. To make this solution practical, automation tools for making this process easier is needed.
- In post-work phase, support engineer needs to use several different tools or scripts to to generate the final point-cloud.
- In post-work phase, support engineer needs to manually calculate the scale factor based on the 'obvious object' and transform the scale and orientation with a matrix.

Conclusion: Technically feasible but not practical, because of the complexity of AR tracking target configuration process.

4.4.2 Solution Candidate 2: *ARCore + Google Cloud + Forge*

In this solution, *ARCore* is used for both the initialization (target recognition & tracking) of virtual content and extended tracking. The target tracking information, which is the ARCore anchor will be stored in *Google* Cloud, and the model is also stored and managed with Autodesk Forge cloud. The whole process is described as below:

- **Configuration-time**: The maintenance engineer scans the target area to collect feature points and save it to *Google* cloud.
- **Authoring-time**: The virtual content can only be placed within the AR view, Offline authoring is not possible, as it is not possible to download the point-cloud information from the cloud. However, as the raw camera frames and feature points are retrievable with *ARCore* API, some third-party AR solution enable developers to retrieve the feature points and store in their private cloud. But that is not an option, considering the solution reliability and extendibility in longer term.
- **Using-time**: Maintenance engineers can view the virtual content in the physical area.

Advantages:

- The AR tracking target configuration is easy, no pictures taking nor manual post-work are needed.
- As the anchor is stored in cloud, it is possible to share the anchor among multiple devices.

Disadvantages:

- *Google ARCore* Cloud Anchor is free, but it is available for only 24 hours. In our case, once an anchor is created and saved, it must be permanently available until it is deleted by the maintenance engineer, in case the physical environment is changed.

Conclusion: Not practical, because of the limitation of *ARCore* anchor storage time.

4.4.3 Solution Candidate 3: *HoloLens* Spatial Anchors + *Forge*

HoloLens is a powerful mixed reality head-mounted display by *Microsoft*. The first version of *HoloLens* was introduced in 2016. It has all the necessary features including the AR target recognition & tracking, and motion tracking. *HoloLens* has a complex camera system, which support instantly re-construct the physical environment to retrieve its feature points and let users to place virtual content in the physical environment with the support of feature 'Spatial Anchor'. The whole process is described as below:

- **Configuration-time**: The maintenance engineer put on *HoloLens*, scan the physical environment by looking around, placing an anchor in the environment.
- **Authoring-time**: The maintenance engineer can directly load the model from Forge to *HoloLens*, and align it with the physical environment, this approach is so called 'online alignment'. Also, offline alignment is possible, that the maintenance engineer saves the position information of anchors and import it into Unity and align virtual content to anchors in Unity.
- **Using-time**: Maintenance engineers can view the virtual content in the physical area. The interaction depends on the capability of *HoloLens*. For example, the interaction of *HoloLens*1 is limited to 'Air-tap' gesture and 'Bloom' gesture, which is why in many cases it must be assisted by voice command. HoloLen2 supports accurate hand tracking and enables users to touch the virtual content intuitively, which also make it possible for developers to design more type of novel interaction patterns.

Advantages:

- The AR target configuration is easy. No picture-taking, just putting on the glasses and looking around.
- Thanks to its depth camera, the anchors accurately quite good, when comparing to mobile AR such as *ARCore* and *ARKit*, which fully relay on the computer vision and sensors. The motion tracking is also more reliable, and the drifting issue is less prominent comparing to mobile AR platform.
- Thanks to its environment understanding camera set, the AR tracking is robust comparing to mobile AR. In contrast, *ARCore* is easier to lose tracking.
- Since it's hand-free, the interaction in AR view does not block the interaction on other applications on the phone or tables. It also enables developers to design novel interactions across the phones/tablets and glasses.

Disadvantages:

- For the first generation of *HoloLens*, the performance of the computing hardware is low, since it's released in 2016. This does not impact the AR tracking performance, but brings problem when loading models, especially large models, even though Autodesk Forge API has a *HoloLens* version that allows to load models directly into *HoloLens*. The performance of interaction with the model is still quite low, which leads to bad user experience.
- The hardware limitation including the narrow Field-of-View and limited interaction is a

problem. Especially, the interaction capability of *HoloLens1* is quite limited. Therefore, a training session might be needed for the maintenance engineers to get familiar with the interaction, especially if they need to align virtual content within *HoloLens*.

- As a gadget with a price tag of more than 3000 US dollars, *HoloLens* is expensive. The hardware price is not critical but still an important item to consider when building a solution, since it will eventually impact the price of the product.

Conclusion: Technically feasible but not practical with *HoloLens* 1. However, since almost all the disadvantage above are for *HoloLens*1, *HoloLens2* could be a practical option but as it is not yet available at this moment, more evaluation needs to be done later when it's accessible. On the other side, it is promising to see that there are many alternatives of *HoloLens* are coming out every few months, which of those could have better FoV, performance or a cheaper price tag. Therefore, as a solution builder, it is wise to keep an eye on new options and make sure that solution architecture is flexible enough to be easily migrated to new platforms.

4.4.4 Solution Candidate 4: *Trimble Connect* (*HoloLens*) + *Trimble* Cloud

The *Trimble* solution is quite special. *Trimble* is a US-based company focusing navigation and tracking solutions. *Trimble* cloud is a private cloud service for AEC companies to store content models. *Trimble* provides a series of software named *Trimble Connect*, which are clients of the cloud for different terminals. It has a *HoloLens* client which support to download model into *HoloLens* from *Trimble* cloud. The whole service and tools are like what is provided by Autodesk Forge. The main difference is that *Trimble* solution is a closed eco-system, which means it provides usable product, instead of development platform for developers to build higher level applications. Nevertheless, the reason why we are interested on *Trimble* solution is to study how it works and learn the technical feasibility.

The process is very simple and it's different from other solutions. It does not have cross-session AR recognition & tracking. Users may put on HoloLens, start *Trimble Connect* software and in the user interface, download the corresponding model from *Trimble* cloud. The model downloaded is only a thumbnail, floating in front of the user. Then users may scale it to 1:1 scale and align it to the physical environment by indicating the mapping between a wall in the physical environment and the one in the model, then the model can be correctly rotated to fit with the real environment. The indicating of wall in physical environment is achieved through the plain detection feature of HoloLens. However, since there is no AR recognition & tracking, users must do the same in every session.

Based on our evaluation, the most interesting feature in *Trimble* solution is about how users can align model to the physical environment, which can be referred when building online alignment feature for *HoloLens* or other AR-glasses.

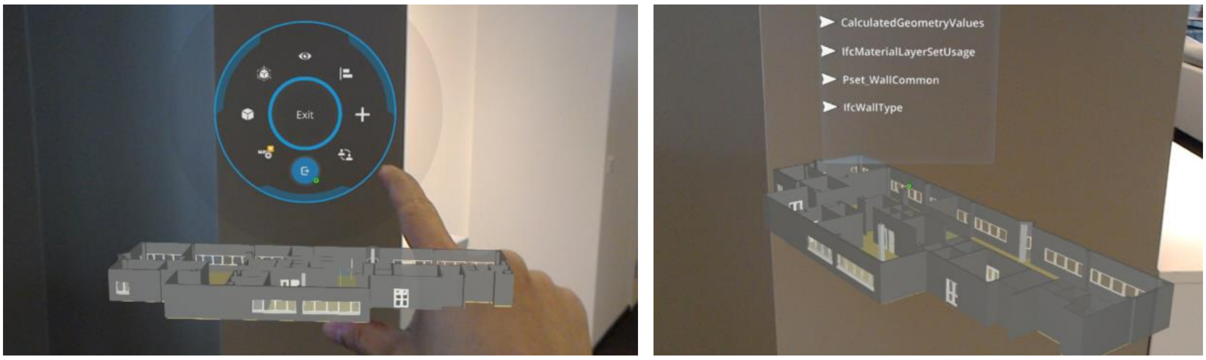

Figure 11. *Trimble Connect HoloLens* client.

Another thing that can be learned from *Trimble* solution is about its integration with existing safety helmet. *Trimble XR10* is the product which integrates *HoloLens* in a helmet. *HoloLens* is directly amounted on the helmet without the head band, and it's possible to flip-up. There is also a balance weight attached to the back side of the helmet, which makes the AR glass in front lighter and easier to wear. During our previous interview with maintenance engineers, practically engineers should be wearing a pair of eye-protection safety goggles which is certified by industrial safety certifications such as OSHA[30], ANSI Z87[38] and CSA Z94.3[39].

Figure 12. *Trimble XR20*[40] with *HoloLens2*

Figure 13. Industrial Safety Goggle (*Neiko 53875B*)

Conclusion: It is not an option to directly use *Trimble* solution in this project, mainly because it is not an open platform for application development. *Trimble Connect* for *HoloLens* is likely only designed as another client of *Trimble* cloud, instead of independent AR solution designed for users in building facility maintenance scenario. However, it presents a good example showing how model alignment can be done within *HoloLens* view and how AR glasses should be integrated with existing safety helmet.

4.4.5 Solution Candidate 5：*ARCore* + *Azure* Spatial Anchor + *Forge*

As stated in previous paragraph, the cloud-based AR solution is considered the trend soon, but unfortunately both Apple *ARKit* and *Google ARCore* has limitation in term of the cloud solution: *ARKit* supports iOS devices only, while *ARCore* cloud anchor can be stored only for 24 hours. Although there are many third-party cloud AR service provider available, the compatibility and extendibility in long term is an important criterion to consider when choosing the low-level technology. So far, the AR hardware is not yet mature enough, and new hardware are coming out every few months. Under this situation, as a solution builder, it is wiser to build the solution upon cross-platform tools, which can be compatible, or at least can be easily migrated to be compatible with new hardware platform.

Microsoft release Azure spatial anchor in February of 2019, which is a cross-platform augmented reality anchor solution. It enables developers to create AR apps that map, designate, and recall precise points of interest that are accessible across *HoloLens*, *iOS*, and *Android* devices. Azure spatial anchor enables users to easily scan a physical environment to collect its feature points (i.e. the point-cloud), and create an anchor, which represents the coordinate system of the point-cloud. Then virtual content can be aligned related to the anchor position.

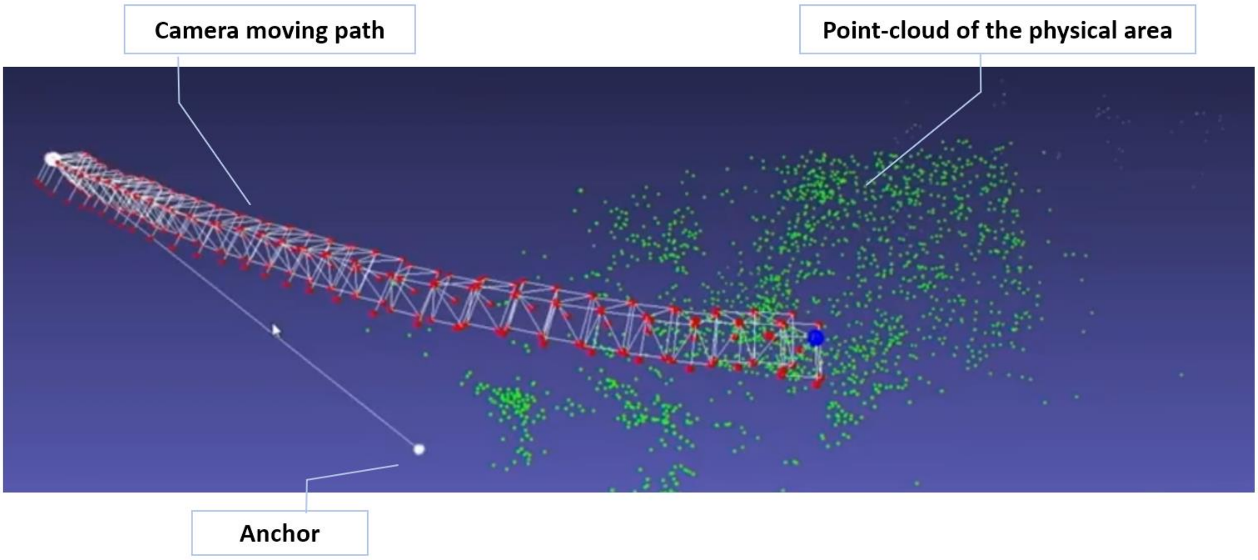

Figure 14. The relationship among point-cloud, anchor and camera positions [41]

The magic why it works for all platform is that it collects raw camera frames from the corresponding platform and retrieve the point-cloud. On the other side, it aggregates the native anchor, such as *ARKit* anchor, *ARCore* anchor and *HoloLens* anchor to retrieve the pose (position and rotation) related to the point-cloud. Then, both the point-cloud and anchor information can be saved to Azure cloud. When user tries to recognize a target area, it instantly compares the point-cloud downloaded from Azure cloud and the point-cloud instantly retrieved from the camera frames. If the two point-clouds can be matched (aligned), it creates a native anchor according to the pose information of the Azure spatial anchor in the cloud. In short, Azure spatial anchor is a higher-level anchor which aggregates the existing ones for specific platform.

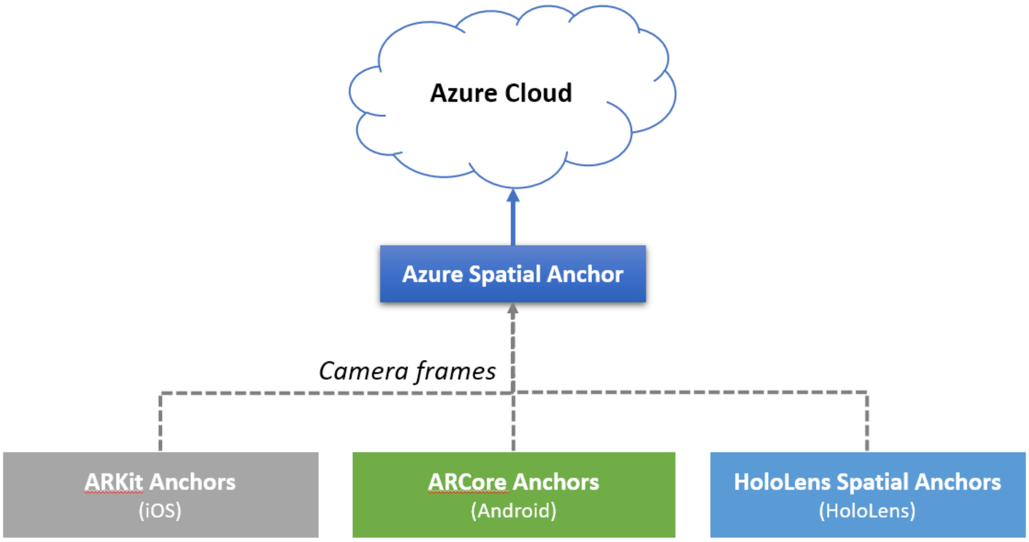

Figure 15, *Azure Spatial Anchor* is a cross-platform cloud-based anchor solution

In this solution, the whole process is described as below:

- **Configuration-time**: Maintenance engineers may use Android/iOS device or *HoloLens* to create Azure spatial anchors (they create native anchors, then convert it to Azure spatial anchor). They can save the created anchor to Azure cloud.
- **Authoring-time**: Support engineers may import the anchors into *Unity* and align the virtual

content with the anchor. Model optimization can also happen in this phase.

- **User-time**: Maintenance engineers can view the virtual content in the physical area.

Advantages:

- It is easy to prepare the environment. To configure the tracking target, maintenance engineers only needs to scan the environment by moving the camera, without picture-taking.
- The offline alignment in authoring tools (*Unity*) is accurate and it enables other beneficial actions, such as model optimization and adding indoor navigation support.
- It is possible to support content sharing and collaboration among multiple persons at the same time. The *Azure* cloud has not only the storage for point-cloud and anchors, but also the Azure application service can be used for synchronizing status to enable collaboration.
- It is easy to integrate other application which also use *Azure* cloud, especially traditional applications which might use *Azure* application service.
- Most importantly, it is easy to migrate to support other platforms such as *iOS* and *HoloLens* in future. This enable us to plan a reasonable and flexible product roadmap in longer term.

Disadvantages:

- The Azure spatial anchor is just released this February and it's still in preview phase. So far it only has basic features with limited document and support.
- Azure cloud service is not free but considering the cost will depends on how much resource is consumed, it is flexible and low for an experimental product.

Conclusion: We concluded that the *Azure* spatial anchor solution is a flexible and smart solution. It enables us to plan a reasonable product road map, i.e. starting with one platform such as *Android* while keeping the possibility to support other platforms such as *HoloLens*.

4.4.6 The Complete Technical Evaluation Table

The table below lists all technology that we evaluated and their pros and cons.

How To Read: For each column(dimension): BEST | LEAST
Author: Mao LIN
Version(Date): 03-08-2019

Technical Tools Selection for Granlund AR (Pros/Cons of different AR platforms)

Technical Solutions	Support Area Point-cloud	Is AR recognization Cloud-based?	Compatability	Configuration Process Complexity	AR tracking reliability	SDK available?	Development difficulty	Offline alignment support	Indoor navigation possible?	Integration with Forge possible?	Price
Vuforia	No	No	Android, iOS, HoloLens	N/A (no area point-cloud support)	N/A (no area tracking support)	Yes	Medium	Yes	No	Yes	Free
ALVAR	Yes	No	Android, HoloLens	Complex (taking pictures, then post-work done with many tools)	Medium	Yes	Medium	Yes	No	Yes	? (not free)
HoloLens Spatial Anchor (local)	Yes	No	HoloLens	Easy (walking & scanning)	High	Yes	Medium	No	No	Yes	Free
Trimble Connect HoloLens client	N/A (no AR recognization)	N/A (no AR recognization)	HoloLens	N/A (no AR recognization)	High	No	N/A (not supported)	No	No	No	? (part of Trimble connect cloud package)
ARCore (only)	Yes	Yes (but 24 hours only)	ARCore(Android)	Easy+ (walking & scanning)	Medium	Yes	Medium	No	No	Yes	Free
VIMAI (ARKit integrated)	Yes	Yes	ARKit(iOS)	Easy++ (walking & scanning, pictures are taken automatically)	Medium+(iOS)	No	N/A (not supported)	No	Yes+ (designed for indoor navigation)	Yes	? (not free)
Immersal (ARCore/ARKit integrated)	Yes	Yes	ARCore(Android), ARKit(iOS)	Easy- (taking picture but with easy tool)	Medium(Android), Medium+(iOS)	Yes	Easy (easy tool and API provided)	Yes	Yes (with extra development)	Yes	High
[SELECTED] Azure Spatial Anchor	Yes	Yes	ARCore(Android), ARKit(iOS), HoloLens	Easy+ (walking & scanning)	Medium(Android), Medium+(iOS), High(HoloLens)	Yes	Medium	Yes	Yes (with extra development)	Yes	Flexible (according to how much it's used by users)

4.4.7 Conclusion

Through a comprehensive analysis, it is concluded that the best solution is candidate 5: using *Azure* Spatial Anchor as the AR engine and Autodesk Forge as the model management platform. There are three reasons for this choice:

1. **Cloud-based**: In this solution, both the AR tracking target and the virtual content are store in the cloud. The AR target (point-cloud and anchors) are stored in *Azure* cloud, while the virtual content (e.g. HVAC models) are stored in Forge cloud. *Azure* cloud for AR tracking target: Azure spatial anchor is cloud-based and supports saving the AR target tracking data (point-cloud and anchor) into *Microsoft Azure* cloud. This makes it easy for maintenance engineer to configure and share AR target. Additionally, the cost of *Azure* spatial anchor depends on how much is used, which is flexible and perfect for building experimental solution. On the other side, the Forge cloud makes it easy to manage and share models between the AR application and other traditional type of applications.
2. **Compatible with all main-stream AR platform:** *Azure* cloud anchor is compatible not only with Apple *ARKit* for *iOS* devices, *Goolge ARCore* for *Android* devices, but also *HoloLens*. This is extremely important because the AR technology today is still an emerging technology, new hardware devices are being released every few months. Therefore, when building a commercial solution upon, it is important to stay on flexible platform, so that we can easily migrate to different platform as needed. Although currently this project targets to *Android* platform, but it is wise to keep the possibility to be compatible with *iOS* devices which has many users, and most importantly, we must make it possible to be migrated to *HoloLens* for future exploration, although for *HoloLens* the user interaction needs to be re-designed completely.
3. **Easy to scale and maintain:** The scalability is mainly for the cloud platform. Even though it is possible to build our own server for storing point-cloud and anchors, it is wiser to employ *Azure* cloud directly, as it is a powerful and reliable cloud service, which is already widely used by enterprise for years. In the current stage, we are still building an experimental product, the cost is low because the expected resource consumption is low, but it is easy to be scaled when the experimental product is upgrade to a commercial product, which would have much more users. In addition, employing third-party cloud service can save the effort and cost for maintenance of the *Azure* cloud server. On the other side, the *Autodesk Forge* cloud is also easy to scale and maintain for the model resources.

5 Design Concepts

5.1 Requirement Summary for a Practical Solution

Among the feedback that we got from the interview, most of them are about how they expect to use the AR application (for solving their problem in daily maintenance work), and most of the user did not mention anything about AR configuration, in detail, how we configure and maintain the AR target. That is understandable because most of the participants during the interview do not have enough expertise about AR technology, even a few people only heard of the concept AR from other movies, books or other sources. However, for building a comprehensive and practical AR solution. It is crucial to consider the maintainability of the solution. During our investigation for technical selection, we found that some options has a complex process which requires many tools or manual steps, or it is effort-taking to manage the AR target. Those options cannot be considered as the solution, at least not until one day they are complemented with automation and tools to be mature enough for a practical solution. Let 's summarize the most core requirement before designing the solution as below:

- **Core Requirement 1: Easy to configure**: The solution must be easy enough for maintain ace engineers to create the AR tracking target, which includes the point-cloud and anchor information.
- **Core Requirement 2. Easy to add virtual content**: The solution must be easy to align the virtual content to the anchor. For the alignment, there are two options: Online alignment and offline alignment. Online alignment means aligning virtual content to the AR target within the AR view, such as the case of *Trimble Connect*. Offline alignment means shifting the alignment work to authoring time with authoring tools such as Unity. Based on the interview feedback, a practical consideration for facility management scenario is that it would be beneficial to shift complex tasks to support engineers (in remote) from the maintenance engineers (in field) as much as possible. This is true especially when the interaction of digital products is not intuitive enough that maintenance engineers needs some training. Therefore, offline alignment is preferred, because the alignment with traditional software in Unity would be easier and more efficient than performing within the AR view. However, this can be changed if more intuitive interaction in AR can be achieved in future.
- **Core Requirement 3. Easy to maintain**: This indicates how easy it is to maintain the AR tracking target. In the simplest but worst case, maintenance engineers must manage AR target manually. For example, engineer A creates an AR target, then he/she needs to export and import to the device of engineer B to enable B for viewing the same content. This kind of inconvenience must be technically avoided. In addition, the solution should also support to manage virtual content easily. In AEC context, especially in facility maintenance context, the virtual content is usually quite large, such as the HVAC model of a whole floor. Therefore, it is not possible for one application to have all the models built-in, and it would be inconvenient if engineers have to export/import models among devices.

5.2 Design Concepts

The concept design is defined as the solution proposals for the requirement. In a typical interaction design project, a concept is mainly about how user interact with the system. However, in this project, as the technical feasibility could be a blocker for the implementation. A design concept without proper technical feasibility evaluation doesn't make any sense. Every concept below has two description, one is from the end users' perspective, with user-friendly words, another is from the technical feasibility 's perspective, with technical words.

5.2.1 'Point-of-Interest' and 'Anchors'

It is mandatory to introduce the terminology of 'Point-of-Interest' and 'Anchors' before designing any concrete concept. Ideally, people may expect to view virtual facilities in any position from any angle as they want, which is too ambitious to be practical. Technically speaking, as the AR recognition relay on the visual 'features' of the physical environment, the feature points or point-cloud of the environment is mandatory for the recognition. Usually it is too ambitious and complicated to generate the point-cloud of the whole building covering every single corner (which might the ideal geometrical 'Digital twin' of the building). In contrast, it is more realistic and practical to choose some places inside the building as the 'Point-of-interest' or PoI, and only make those PoI recognizable by the AR application. However, users can setup as many PoI as they want and there is no limitation. The more PoI is setup, the easier to initialize the AR view. But at least, a good practice is to set up PoI based on the 'critical area' of the maintenance work, for example, there should have PoIs around important components such as Air-handling unit (AHU).

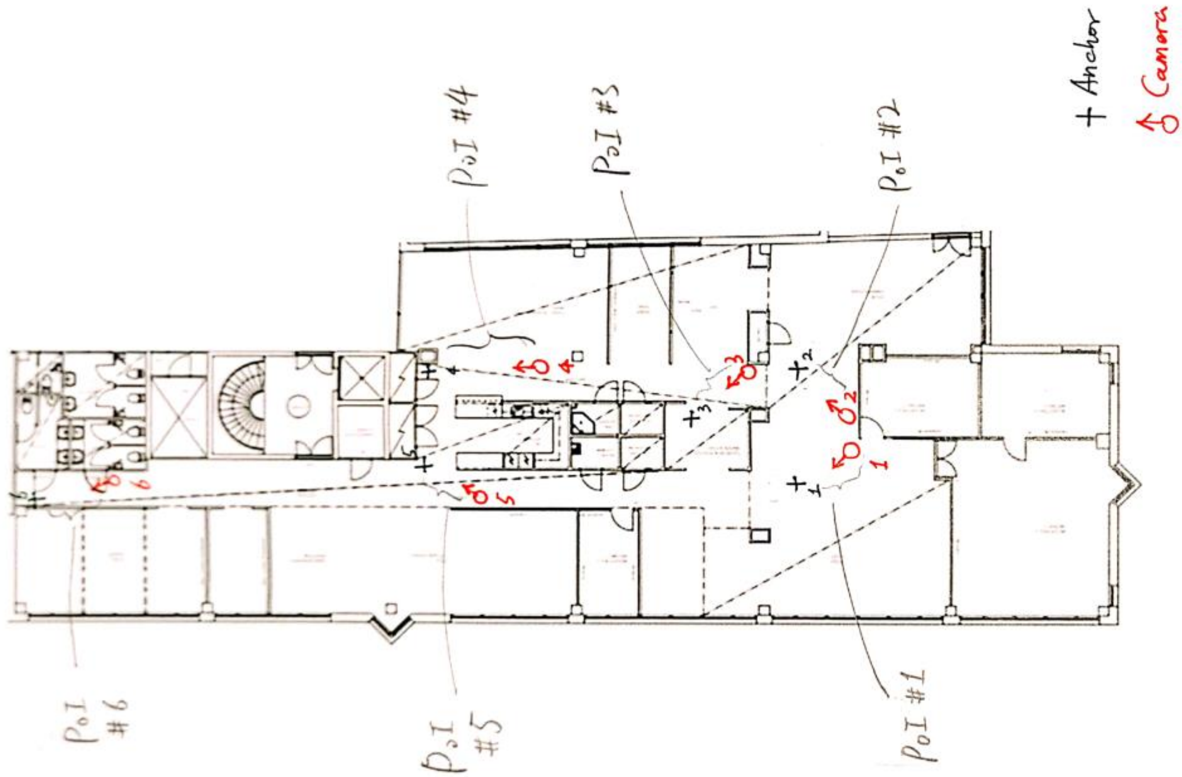

Figure 16 'Point of Interest' and 'Anchors'

For every PoI, the point-cloud is the critical element for AR recognition, which contains all the feature points of the physical area. As the point-cloud is a set of points which is hard for human to identify and align the virtual content which should be attached to it. The concept of 'Anchor' is introduced, which represent the coordinate of one 'point-cloud'. As anchor is the representation of the pose (position and rotation) of a point cloud, the task of attaching virtual content to the point-cloud is now simplified to attaching virtual content to an anchor, which has a clear coordinate, visually. And in the user time, when an physical environment is recognized successfully, the related anchor is discovered. In another word, the task of recognizing a physical area is transferred to discovering the anchor objects. When anchors are discovered, that mean AR view is successfully initialized, and all the virtual content are ready to be visualized. The figure below shows the relationship among those three elements: the point-cloud, and anchor and virtual content.

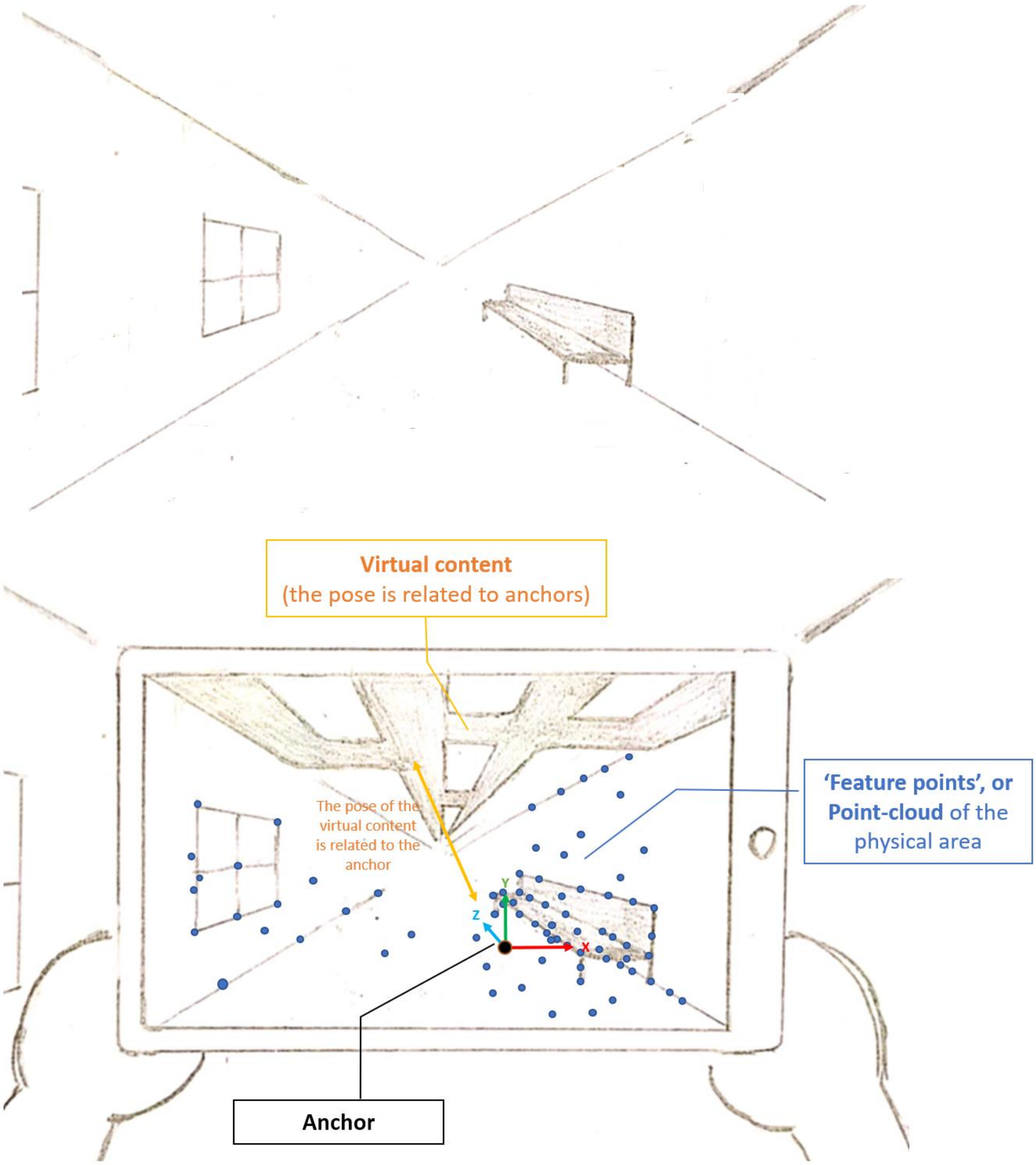

Figure 17. The relationship among 'Point-cloud', anchor and the virtual content.

One thing special in this project is that the virtual content is huge and shared among multiple PoI. In detail, rather than visualizing different virtual content separately for different Point-of-Interest, the only model (architecture model, electricity model or HVAC model) of the floor is expected to be visualized, but we must make sure the relative pose is correct according to user's position. Also, it's important to keep the tracking stably when users are walking inside the building. The maintenance engineers may need walk from one PoI to another, the HVAC model are expected to stay in the right position stably. The ARCore motion tracking can be employed for enabling that. But the drifting problem in mobile device is a serious usability problem, especially when user walk 20 meters away from one PoI. The solution is to make it possible to switch the focus anchor and align the virtual content according to the current focus anchor.

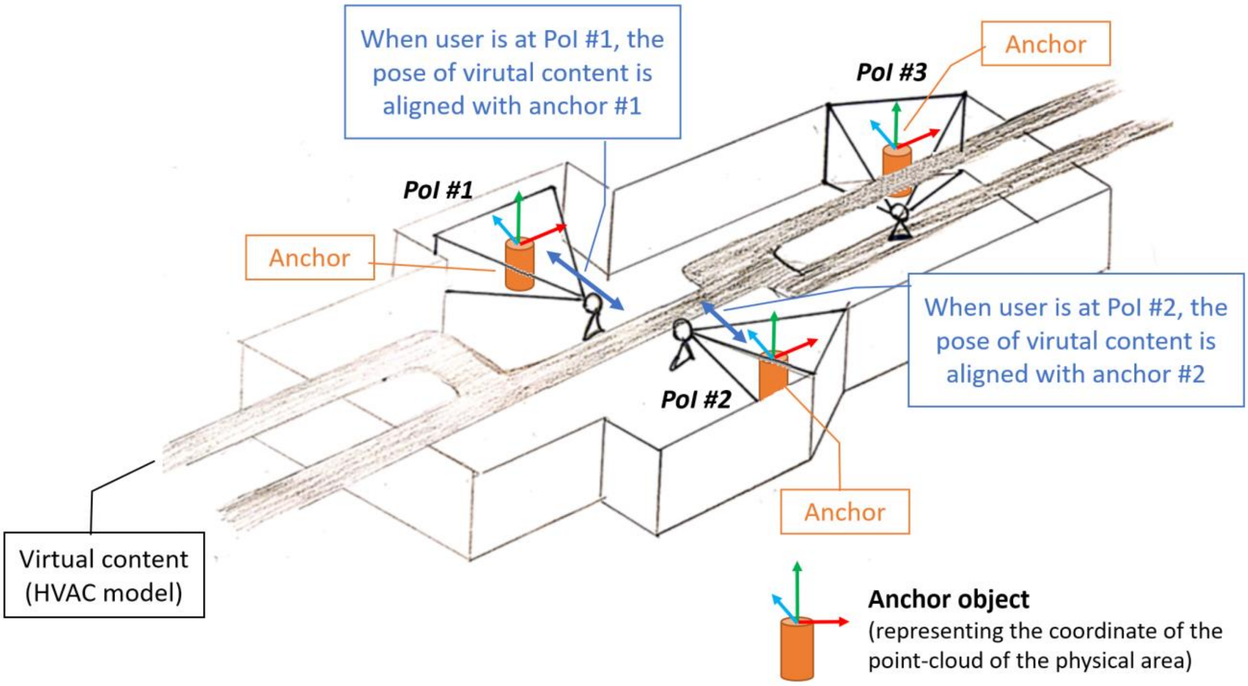

Figure 18. The pose of the virtual content is aligned with the focus anchor, according to user's position.

5.2.2 Design Concept 1: Preparing 'Point-of-Interest'

Summary		
Who is it for?	Facility maintenance engineers and support engineers	
To do what?	Making a physical floor as 'AR-ready'	
One sentence summary	The people doing maintenance work for a building are able to manage the AR point-of-interest inside the building as well as the related virutal content.	
Details	**User Interaction (user-friendly words)**	**Technical Translation (technical words)**
User Scenario	1. The maintenance engineers are able easily create a 'point-of-interest' by scaning a room-scale physical environment to make it 'AR-ready'. They may create as many 'point-of-interest' as needed for a floor. They may also update or delete a 'point-of-interest'.	1. The maintenance engineers creates Azure Spatial Anchors for a room-scale physical environment to make it 'AR-ready'. They may create as many anchors as needed. They may also update or delete an anchor.
	2. Once a point-of-interest is created and saved, the support engineers can easily attach virtual content to it, so that the virtual content can be visualzed in the AR view later. In practice, the on-site maintenance engineers can be responsible for collecting envinronment data of the physical environment and the support engineers in remote can be responsible for attaching virtual content.	2. One an anchor is created and saved, that means the environment data are saved to Azure cloud. Then the support engineers can attach virutal content to the anchor, so that the virutal content can be visualized in the AR viewer later. In practice, the maintenance engineers are responsible for createing anchors on-site, while support engineers in remote are responsible for align anchors to virtual content, such as HVAC models.

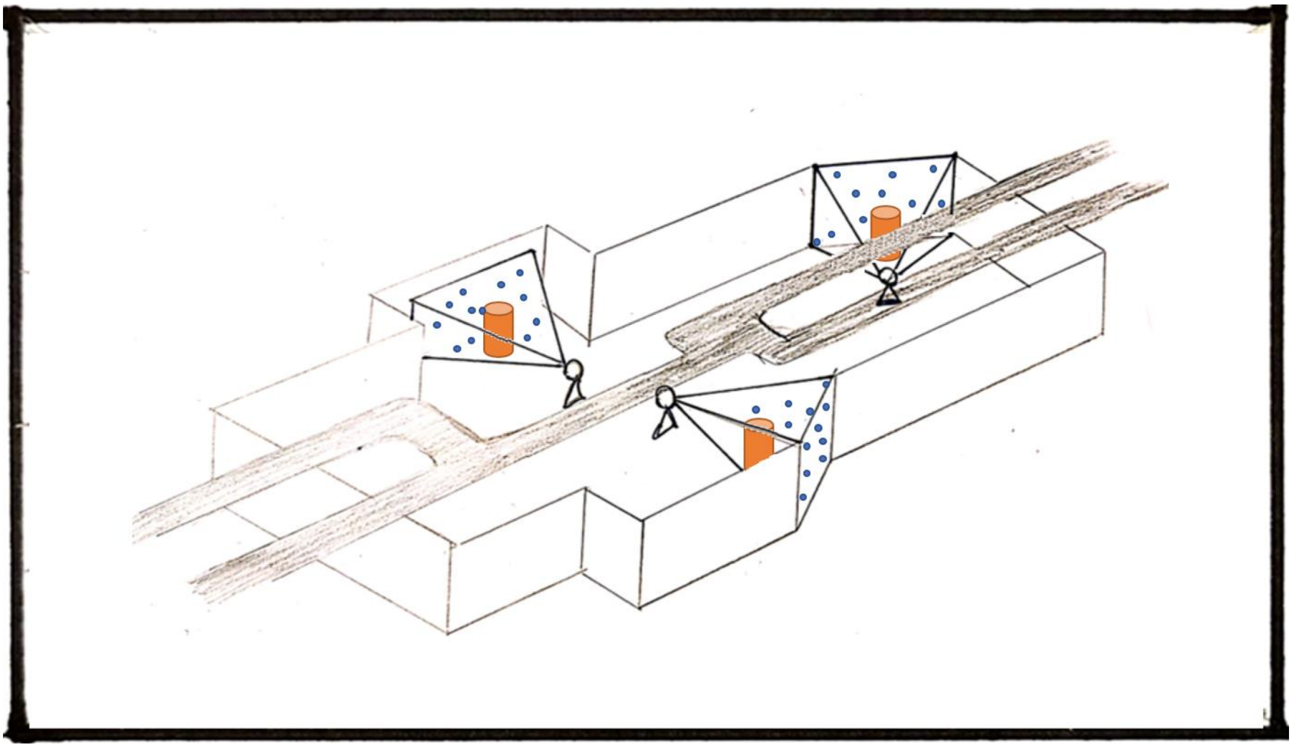

Figure 19. Design concept 1.

5.2.3 Design Concept 2: Visualizing virtual facilities

Summary		
Who is it for?	Facility maintenance engineers	
To do what?	Viewing the facilities behine walls or ceilings	
One sentence summary	When the maintenance engineer arrive one of the point-of-interest location, he/she can view the facilities behind the ceilings or walls in the AR view.	
Details	**User Interaction (user-friendly words)**	**Technical Translation (technical words)**
User Scenario	1. The user already know the position of the 'point-of-interest', where AR view can be launched. They may arrive the place directly or with the help from a 2D floor plan, where all the 'point-of-interest' are marked.	1. The user already know the position of anchors in the floor, where AR view can be initialized. They may arrive the place directly or with the help from a 2D floor plan, where all the anchors are marked.
	2. When they arrive the point-of-interest, they hold the device and move around viewing the area with the camera, until the area is recognized and virtual facilities are visualized.	2. When user arrive the position where there is an anchor, they hold the device and move around to collect enough feature points, so that the AR viewer can recognize the point-cloud of the area. When AR recognition succeeded, the anchor object shows up, and virtual facilities can be visualized

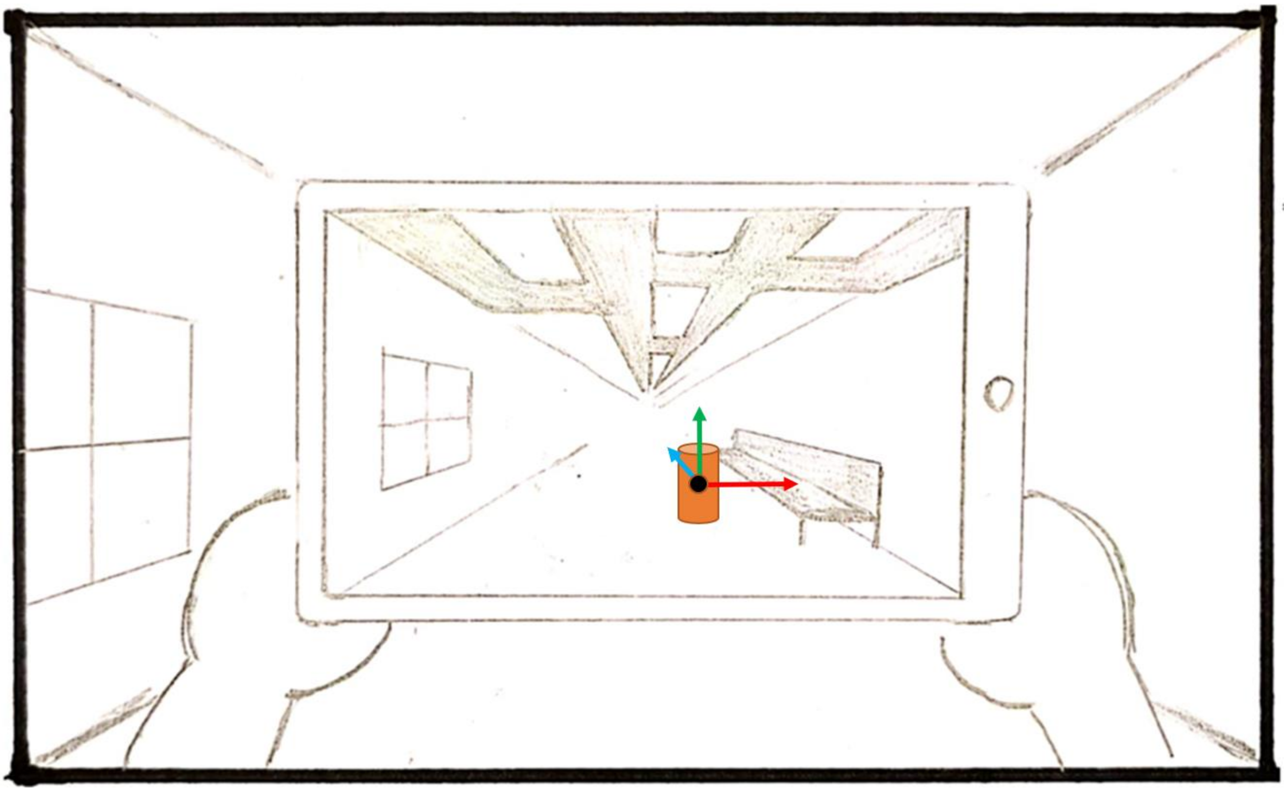

Figure 20. Design concept 2.

5.2.4 Design Concept 3: Retrieving facility information

Summary		
Who is it for?	Facility maintenance engineers	
To do what?	Viewing facility information	
One sentence summary	During the maintenance work, the engineers can view the static and dynamic information of the selected object	
Details	**User Interaction (user-friendly words)**	**Technical Translation (technical words)**
User Scenario	1. When they arrive the point-of-interest, they hold the device and move around viewing the area with the camera, until the area is recognized and virtual facilities are visualized.	1. When they arrive the point-of-interest, they hold the device and move around to make sure enough feature points are collected by the camera, so that it can be recognized, and then the anchor object can be visualized. The HVAC model can be also visualized and the pose is aligned according to the anchor object.
	2. They tap an object of the facility, static informaiton such as its properties from BIM model or facility management system can be visible.	2. They tap an component of the HVAC model, the static information of the selected components can be retrieved from its BIM model properties through Autodesk Forge API.
	3. In addition, dynamic informaiton such as instant status and data from sensors can also be retrieved for diagnosis purpose.	3. In addition, dynamic information such as the instant status and data from sensor can be retrieved from Building Automation System. This is beneficial for diagnosis purpose.

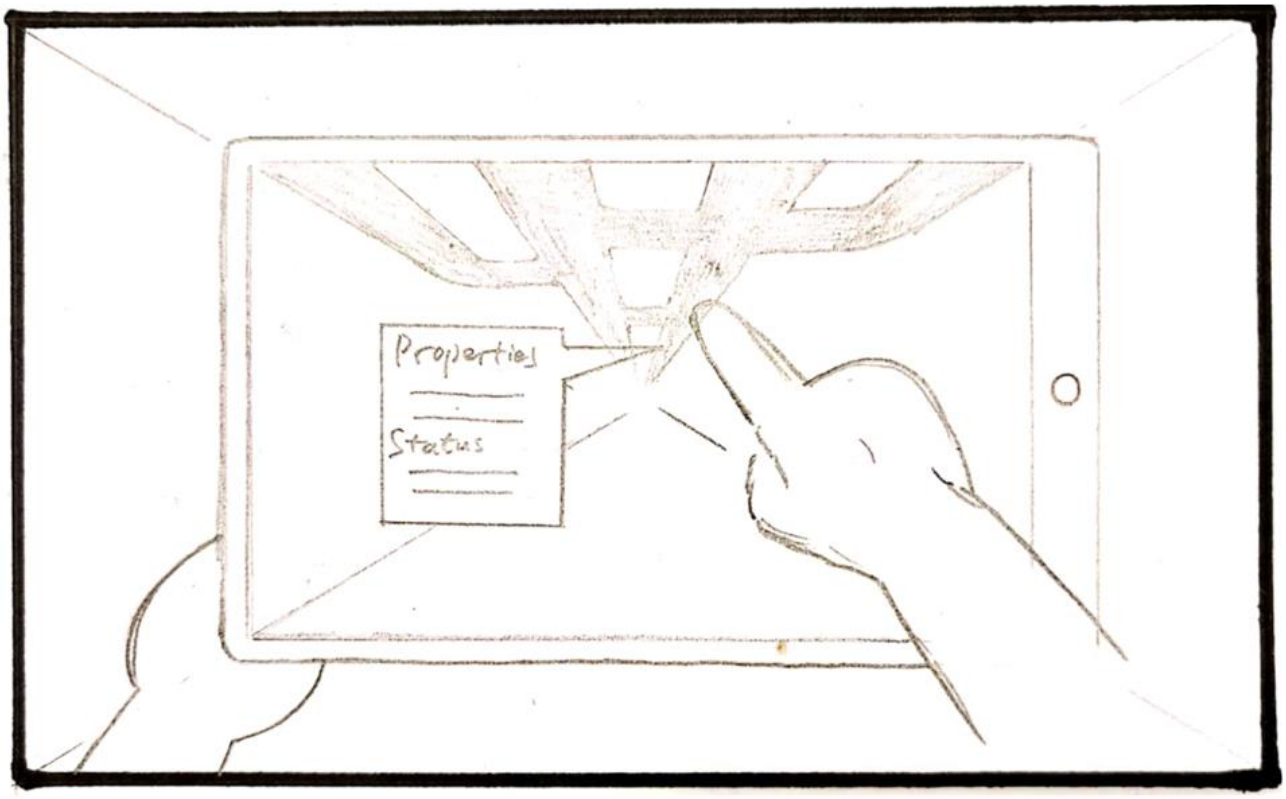

Figure 21. Design concept 3.

5.2.5 Design Concept 4: Indoor navigation

Summary		
Who is it for?	Facility maintenance engineers	
To do what?	Navigate to an point-of-interest	
One sentence summary	During the maintenance work, the engineers can navigate to one of the 'point-of-interest' through indoor navigation	
Details	User Interaction (user-friendly words)	Technical Translation (technical words)
User Scenario	**Scenario 1: Start indoor navigation from any point-of-interest** Maintenance engineers can start indoor navigation from one point-of-interest to another. In practice, the entrance of the building can be one of the point-of-interest and user may start from there.	**Scenario 1: Start indoor navigation from any anchor** Maintenance engineers can start indoor navigation from one anchor to another. In practice, one anchor can be created for the entrance area and user may start from there.
	Scenario 2: Start adhoc indoor navigation based on the concrete maintenance task Maintenance engineers may start adhoc indoor navigation for a concrete maintenace task. For example, if task A is for fixing machine M, user can be navigated to the closest point-of-interest to machine M.	**Scenario 2: Start adhoc indoor navigation based on the concrete maintenance task** Maintenance engineers may start adhoc indoor navigation for a concrete maintenace task. For example, if task A is for fixing machine M, user can be navigated to the closest anchor to machine M.

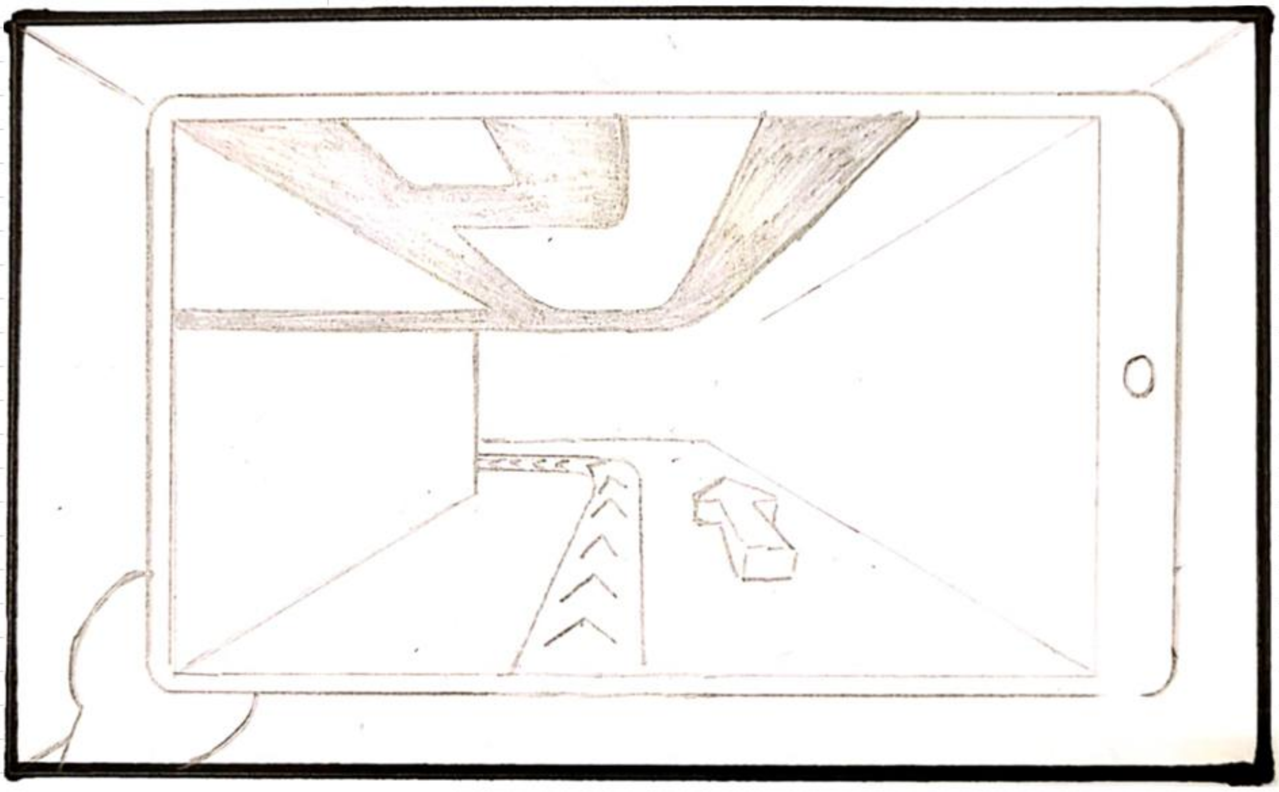

Figure 22. Design concept 4.

5.3 UI Prototype Wireframe

Based on the requirement, the first thing that maintenance engineers need to do is specifying the building and floor that they arrived. The user interface below consists of three parts: the top panel, central panel and bottom panel. Only the central panel is semi-transparent. This design has a metaphor of ‘opening a window’. The semi-transparent central panel shows the camera view, which is like a ‘peeking gap’ of the AR world. When user selects the building and floor and taps the ‘Go’ button, the window, which is the top panel and bottom panel will be open with an animation.

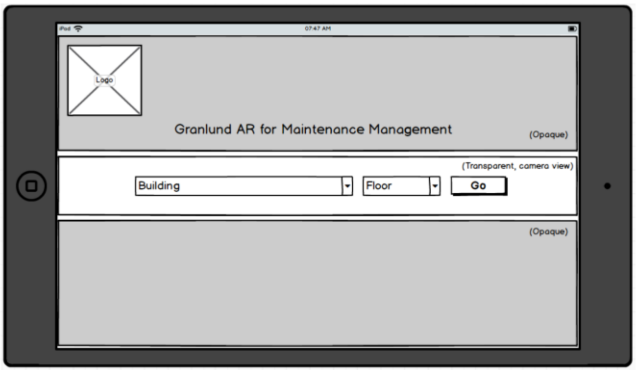

Figure 23. Login UI

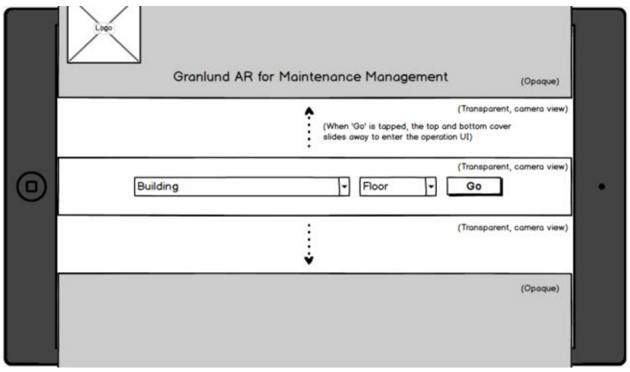

Figure 24. Login animation

After login, user will be at ‘operation UI’. By default, since there are no AR related activities, it is not a good idea to directly open the camera, otherwise then AR activities starts, it would be hard for users to realize it’s started. Therefore, the default operation UI has a camera cover, which will be slide away when AR activities starts, such as when user start creating an anchor, or start locating anchors.

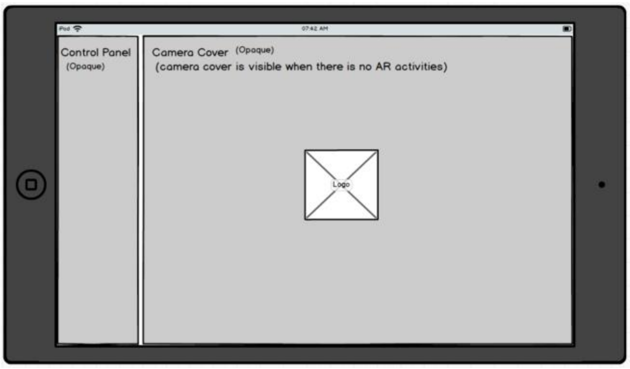

Figure 25. Camera cover

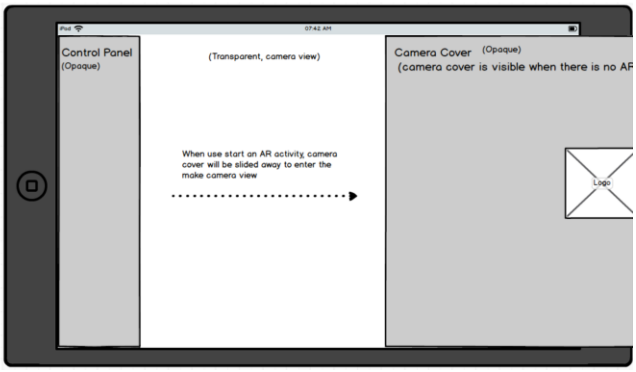

Figure 26. Animation of opening the camera cover

When an AR activity is started and the camera cover slide away, the layout of the operation UI will be divided into three parts. The control panel in the left, the camera view area in the center and the properties panel in the right. In the camera area, we reserved the position for some necessary component, including the drop-down list for selecting indoor navigation destination on the top, a progress-bar component in the middle and a message area at the bottom. The control panel in the left is the place for the buttons for AR operation, such as the buttons for creating, locating, saving and deleting an anchor.

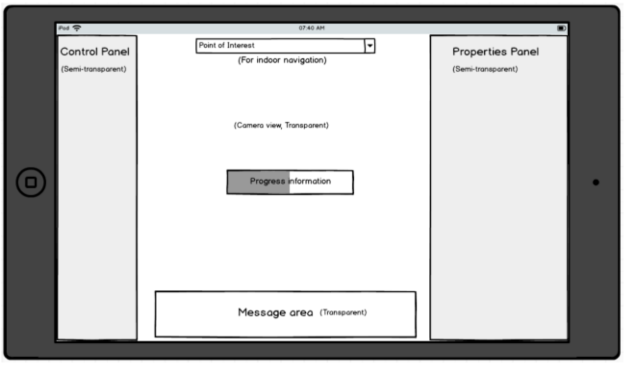

Figure 27. General layout

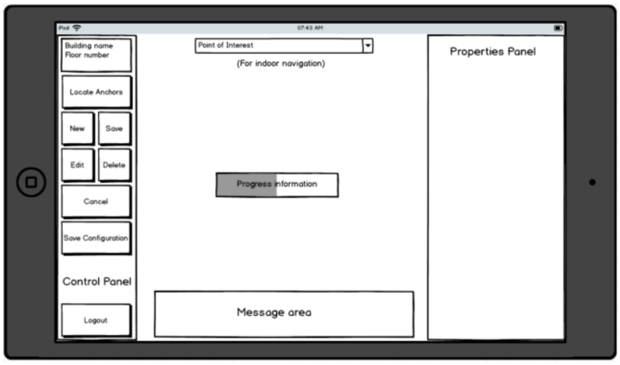

Figure 28. Operation UI

The properties panel in the right is divided into three sub panels, a panel for showing BIM properties on the top, a panel for showing dynamic information in the middle and a panel for showing maintenance tasks at the bottom. Each panel have the same interaction behavior that they can be close by tapping the close button, and new panels can be open when items inside is tapped. For example, if user tap on one property on the dynamic information panel, the detail statistics chart will be shown in a new panel popped-up. And since the size of the panels is limited, they all have a scroll-bar if the size is not enough for showing the full content.

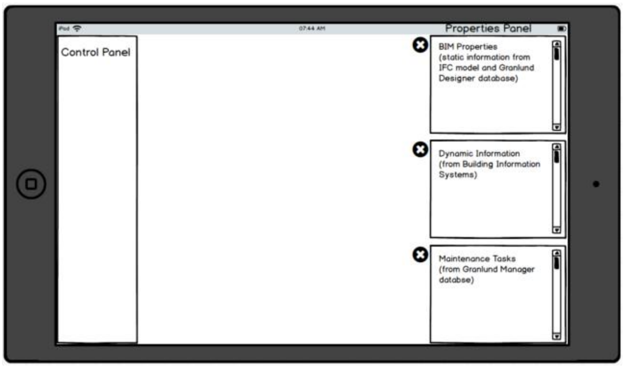

Figure 29. Properties panels

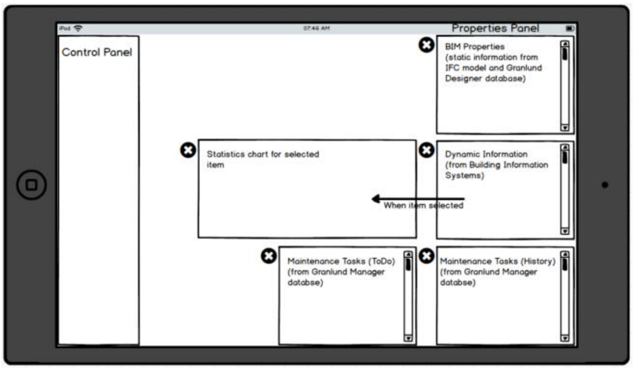

Figure 30. More properties panel

6 Solution Design and Implementation

6.1 The Solution Architecture

The whole system is designed in an architecture with six components, the architecture overview diagram below shows how these six components works with each other.

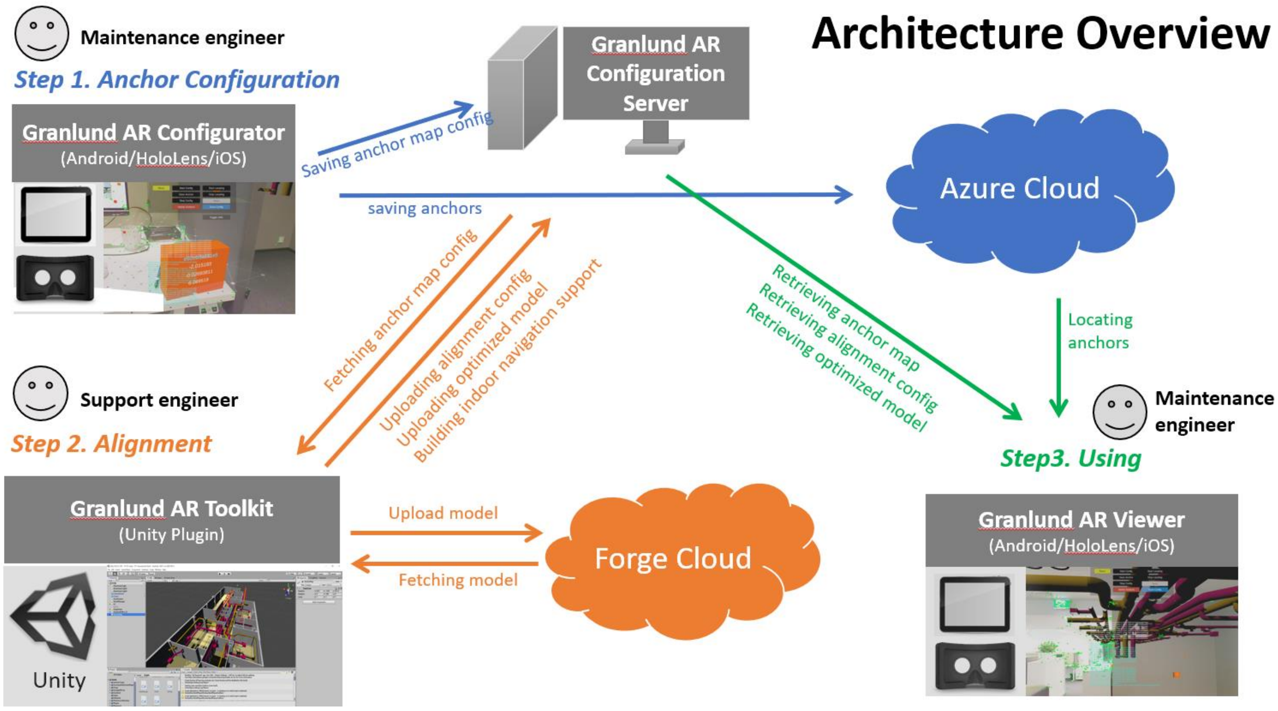

Figure 31. The architecture overviews

- ***Granlund* AR Configurator**: It is the tool for maintenance engineers to make the physical environment 'AR-ready'. Maintenance engineers may use it to create multiple anchors for one floor in the physical environment and save them to the cloud.
- ***Granlund* AR Viewer**: It is the final AR application for maintenance engineers to view facilities in AR. In this project, we integrated *Granlund* AR Configurator and AR Viewer into one Android application, because according to user interview, normally there are only 1-2 maintenance engineers responsible for one building, in practice. The end users may be the configurator, too.
- ***Granlund* AR Toolkit**: This component is a set of Unity plugin, which enables support engineers to do three activities:
    - **Virtual content alignment**: The support engineers aligns virtual content with anchors. They can download and import anchor configuration from AR configuration server, which is collected by maintenance engineers through *Granlund* AR Configurator.
    - **Model optimization**: As Unity is not only a powerful 3D engine but also common platform with a huge number of plugins for various type of purpose. It is easy to integrate 'Level of Details' or LoD plugins for model optimization. This is crucial for AEC projects as the models of AEC context are usually quite large.

  - **Indoor Navigation**: Based on the interview, indoor navigation is considered very beneficial for maintenance engineers. Although indoor navigation is not a mandatory feature in this project. But for designing a comprehensive solution architecture, it is such a important feature that we must consider to support. The original *Microsoft* Azure spatial anchor support a feature so called 'way-finding', which provides a general direction guidance for users. However, it can only provide user a compass-like experience: users can only get a general idea of where to go, because it does not understand the walls thus it cannot provide accurate navigation like 'turning left' or 'turning right'. But in our solution, thanks to the diversity of available plugins in Unity, it is possible to utilize a built-in navigation plugin named *'NavMesh'*, which make it possible to build navigation path intelligently for a whole floor.

- ***Granlund* AR Configuration Server**: There are several reasons why we need our own configuration server, even besides the Azure cloud and Forge cloud. The major reason is the we need a server for storing support information, such as the list of *Granlund* buildings/floors their related anchors. In this architecture, Azure cloud is mainly the storage of anchors and point-clouds, and that of all the buildings will be stored there. It is quite common that sometimes point-clouds looks similar, when two places have similar layout or decoration. To avoid the situation that one place is incorrectly recognized as another one, we need filter the anchor query with the information of building and floor. In detail, the anchor query starts from the *Granlund* AR configuration server, it gets the list of anchors of specified building and floor and query from *Azure* cloud service. Technically, the AR configuration server is a simple HTTP server, implemented with Node.js. In this experimental project, it is a standalone server. But it can also be integrated with other *Granlund* servers, or instead of running on our own server, it can be migrated to be running in Azure application servers.
- ***Microsoft Azure* Cloud**: This component includes the storage of *Azure* spatial anchor and Azure application service. The former part is technically not only the storage of the anchors, but also the point-clouds. The latter part is for sharing the device pose related to anchors for multiple user collaboration.
- ***Autodesk Forge* Cloud**: In this solution, *Autodesk Forge* cloud is mainly for model storage and management. The models of all sorts of facilities, including the architecture models, HVAC models, electricity system models..., etc. are stored in Forge cloud. They can be easily previewed and managed through web browser. Mostly importantly, in this way, the same version of model can be accessed through different type of clients, including the web application (e.g. *Granlund Manager*, *Granlund Designer*), mobile application (*Android* / *iOS*), desktop application or AR/VR application (based on *Unity*).

6.2 The Process

In this section we will talk about the whole process from the user's point-of-view, in another word, how user use this system in practice. There are two roles among the end user, like we defined in 'Personas'. The maintenance engineers who works in the target building and support engineers who works in remote. The whole process consists three phases: anchor configuration, anchor – model alignment and AR viewing.

6.2.1 Anchor Configuration

Since the models in AECO project are usually separated by floors. For every floor, there is an architecture model, a model for HVAC system, a model for electricity…, etc. Therefore, for the AR solution, the AR configuration should be done for one floor. In detail, the maintenance engineers create anchors for one floor. This can also make the anchors more reliable, because the physical environment could look the same, especially inside office buildings, which can lead to fault that incorrectly recognize one place as another. If we manage anchors as a group for one floor, that means for every AR recognition, the query happens only for the current floor.

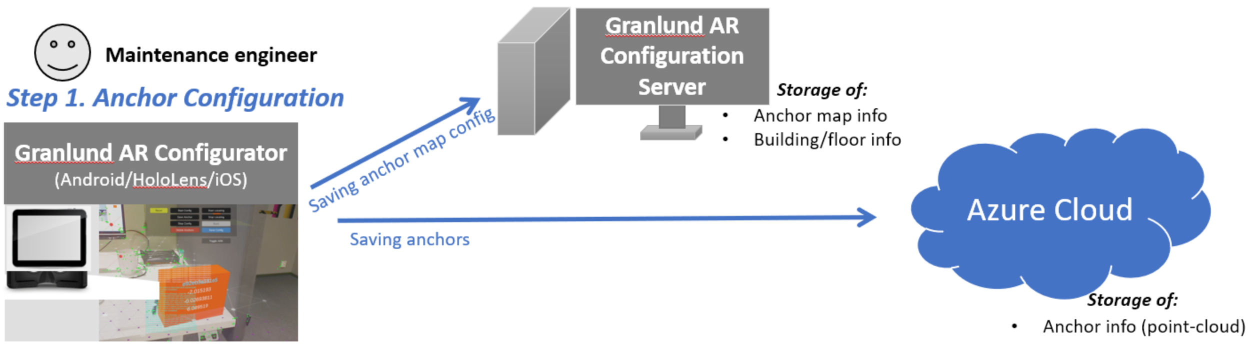

Figure 32. Anchor configuration process

When the maintenance engineer arrives the target building, from the login user interface, they may choose the building and floor. Then they can enter the operation UI.

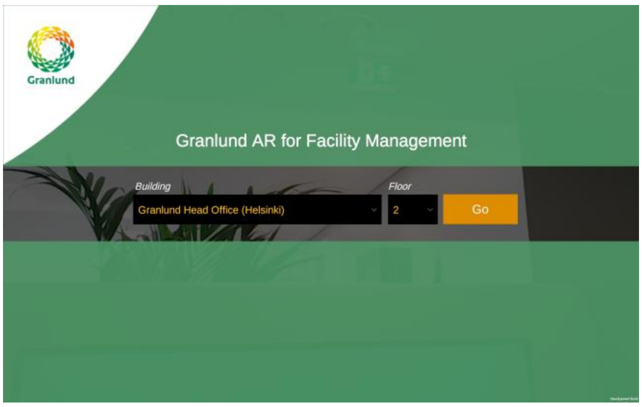

Figure 33. The Login UI

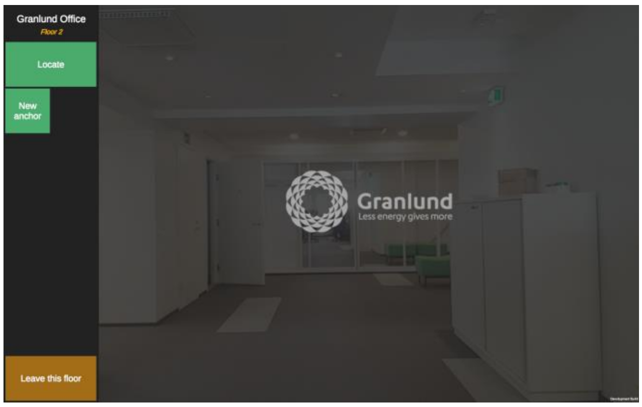

Figure 34. The Operation UI

By default, the camera view is available as the semi-transparent background, but the AR features are not activated. From the available buttons in the control panel in the left, users can see that they can create new anchors or locate anchors of this floor. They may also tap 'Logout' button to go back to the login UI to switch to another floor or building. When they do that, the whole AR session will be reset.

User may start the anchor creating session by tapping 'New Anchor' button. Then *ARCore* plane detection and visualizer are activated, which can help users to place anchors on the ground. According to the design, anchors are only allowed to be placed on detected ground, that's why there is a hint message showing at the bottom of the screen, asking users to 'Move around to get plane detected, choose a position to create an anchor by tapping on the plane'.

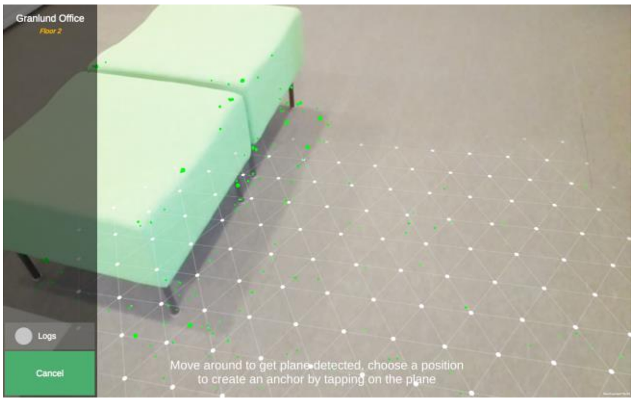

Figure 35. When user starts the anchor creation session.

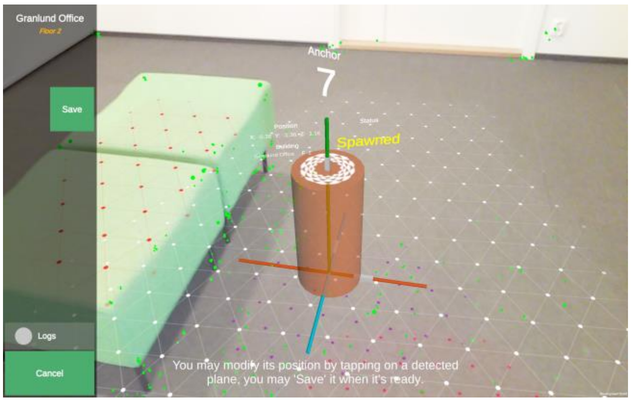

Figure 36. When an anchor object is spawned.

By tapping a position on the plane, an anchor object can be spawned. The anchor object is a cylinder body with three bars which represents x, y, z axis. Users may change its position by tapping on a different place. When it's ready, users may tap 'Save' button to save the anchor to the cloud. Technically speaking, what is saved in detail is the point-cloud of this area and the pose of the anchor related to the point-cloud. Therefore, it is critical to make sure to have enough features, otherwise the anchor cannot be easily recognized. We showed a message asking users to move the device to capture enough environment data and implemented a progress indicator in percentage. When 100% is reached, the anchor can be saved. It may take a few seconds to save the anchor, according to the network situation.

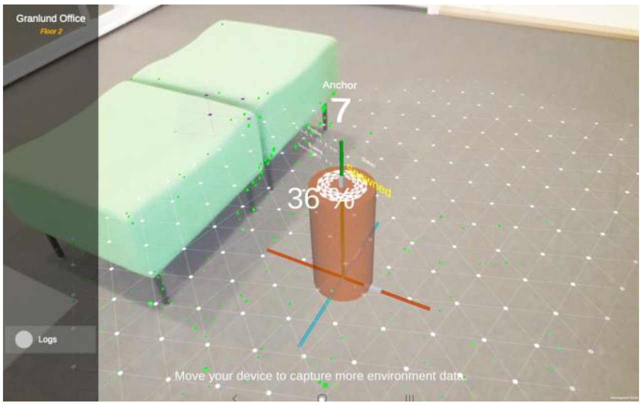

Figure 37. Moving device to collect enough feature

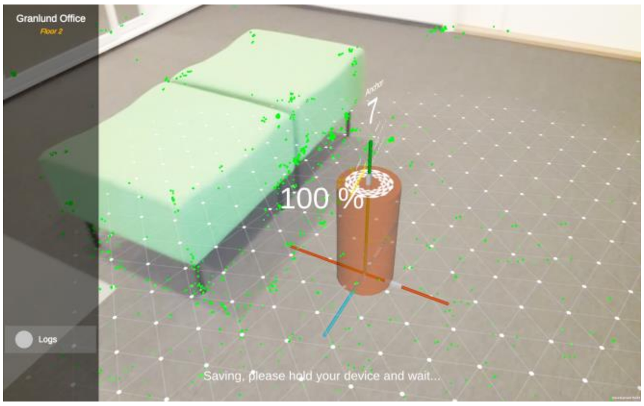

Figure 38. Saving an anchor to Azure cloud

If the anchor is saved successfully, it got an unique ID assigned by Azure spatial anchor service. This ID will be used for query later for locating. After saving, the anchor is located automatically, and then user can continue creating more anchors for the floor.

In addition, they need to mark the exact position of the anchors on a 2D floor plan. This is for making the content alignment easier, otherwise user may forget the anchor position when doing the alignment. This is mandatory at least for the current version. However, as it takes extra effort for maintenance engineers, we consider it as a potential usability breakdown, and we will discuss the possible solution in the last chapter.

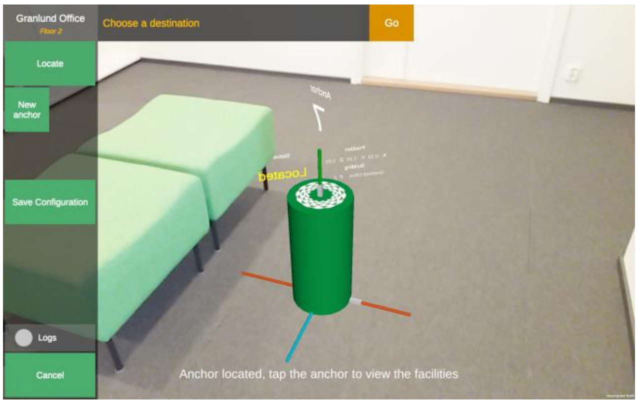

Figure 39. Anchor is located automatically afte saving

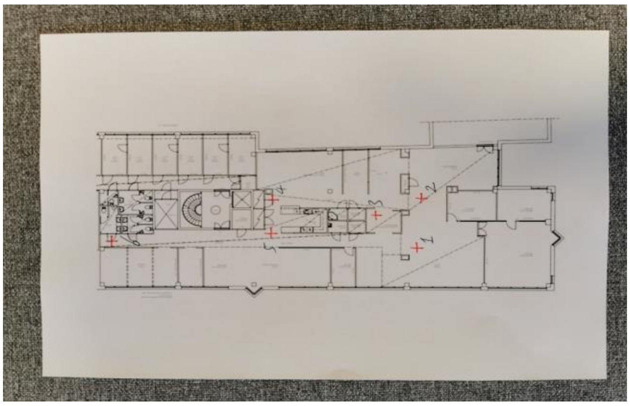

Figure 40. Marking anchors position on a floor plan

When all the needed anchors are created. User may tap button ‘Save Configuration’ to save an anchor map. Anchor map is a configuration file with the list of anchor and their pose information. An anchor map can be imported into Unity with ‘*Granlund* AR Toolkit’ for alignment with virtual content.

6.2.2 Anchor-Model Alignment

In this solution, the alignment between the anchor and the virtual content is done offline within Unity editor. For maintenance companies in practice, the AR configuration can be done by the maintenance engineer onsite, while the alignment work can be done by support engineer in remote. This is reasonable because the placing anchors is relatively as easy as a game, but the alignment work requires more computer skills.

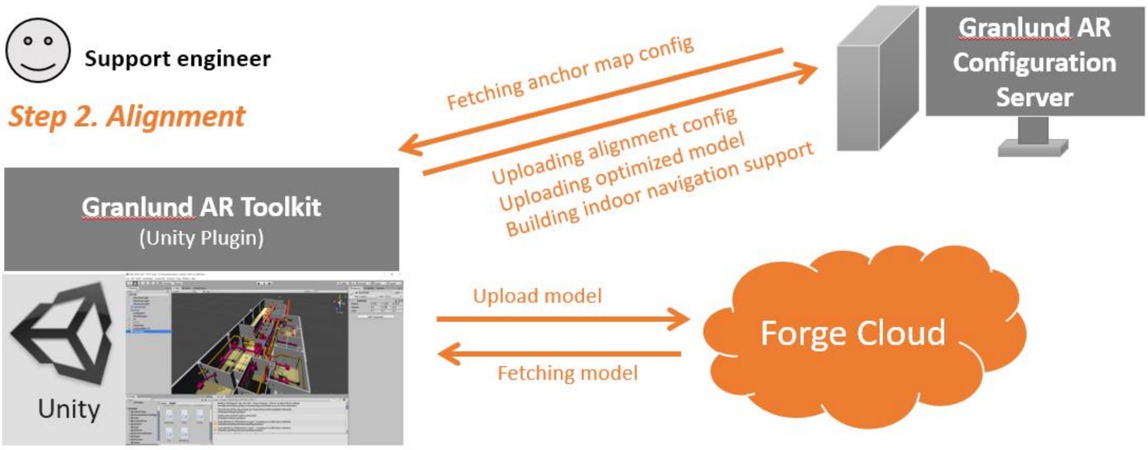

Figure 41. Anchor-model alignment process

To import an anchor map, users can create a floor with ‘*Granlund* AR Toolkit’, which is a Unity plugin. This way of creating AR target is intuitive and following the same workflow as many other AR authoring tools such as *Vuforia* SDK. When a ‘GranlundFloor’ is created, users may use the tools in Inspector view to do the required action step-by-step. By taking a close look at the ‘GranlundFloor’ object, users may see that it consists an anchor map and models, which are the two elements to be aligned. Now users need to follow and complete the steps.

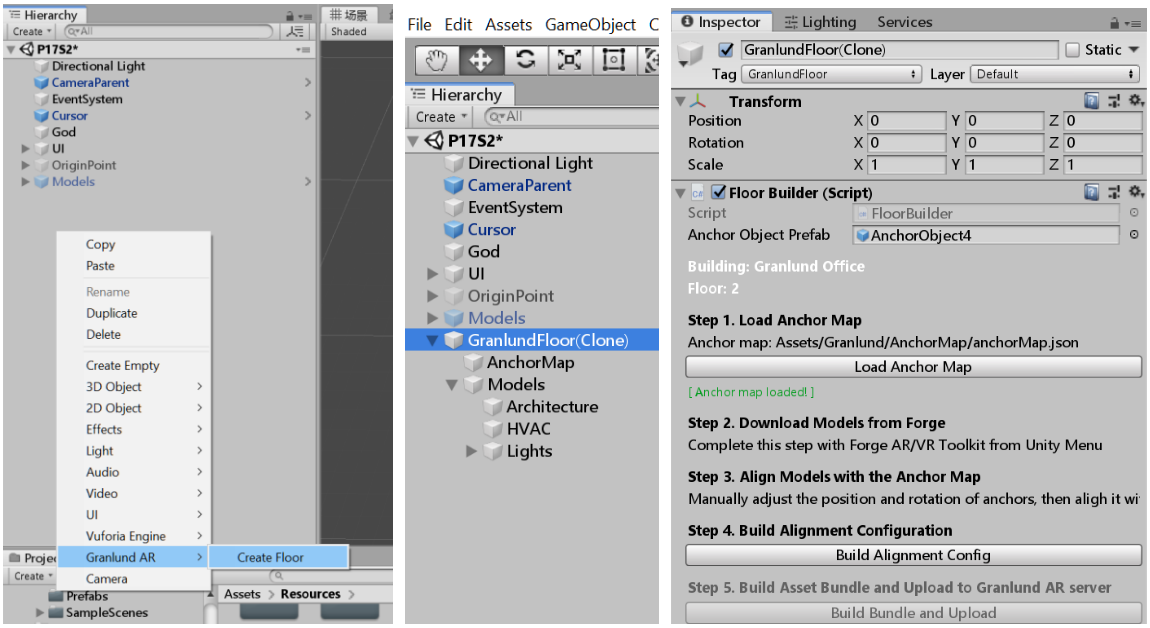

Figure 42. Creating a floor with *Granlund* AR Toolkit.

The first step is the load an anchor map. This can be done simply by clicking button 'Load Anchor Map', once it's loaded, use can see the anchors not only listed in the object Hierarchy view, but also visualized in the scene. The anchors are in the same relative position as when they are created in the physical environment. That is ensured by *ARCore* motion tracking during the AR configuration phase. However, as it is mentioned in previous chapters, as the motion tracking of phones and tablets are completely relay on the embedded sensors including gyroscope and accelerometer, sensor drift is a noticeable limitation when user walks too far away. Therefore, it could be noticeable that in the loaded anchor map, the position of some anchors which are far away from the starting position might be not in the exact position as it should be. However, users can adjust the position of every anchor as needed, according to the paper floor plan with anchor position marks.

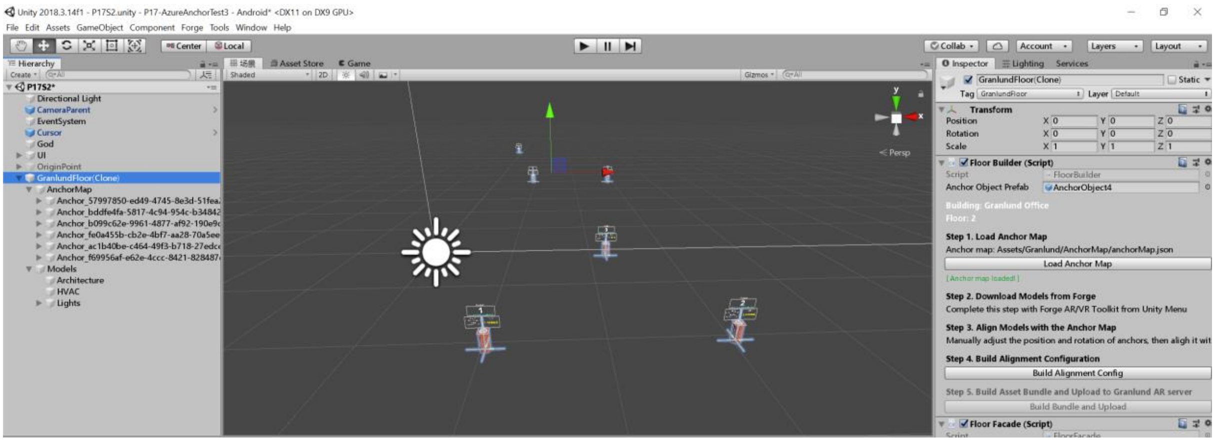

Figure 43. The loaded anchor map in Unity editor

The seconds step is to download the model from *Forge* cloud with *Autodesk Forge* Unity plugin. In future version, we can consider to fully integrate the downloading function from Forge plugin into *Granlund* AR Toolkit, to make the process easier. Users may download multiple models. But the architecture model is useful only for alignment, and the HVAC model or other type of model is the final content to be visualized in AR view. There is a prerequisite that the models are already aligned to each other before uploading to *Forge* cloud. This can be ensured by model designers in the model authoring time.

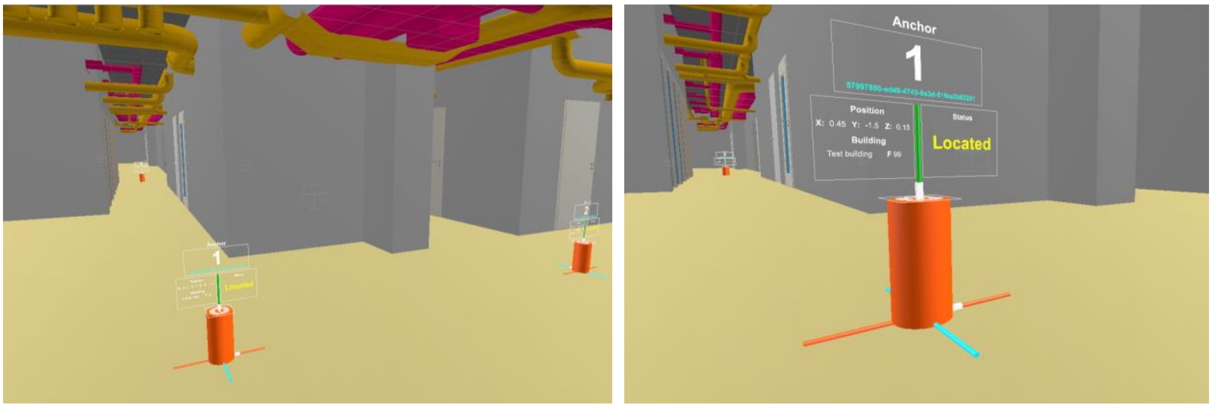

Figure 44. Users may adjust the position of the anchors or models to achieve the accurate alignment in Unity

Users may adjust the pose of anchors or models to achieve the perfect alignment. After it's ready, users may generate 'Alignment config', which is a configuration file containing the list of anchors and relative pose information for the model. The alignment configuration file is the magic of how a large virtual content such as the HVAC model can be visualized in the correct pose (position and rotation), no matter from which point-of-interest users view inside the floor. With Granlund AR Tookit, the alignment configuration file can be saved in *Granlund* AR Server, and the information will be queried by *Granlund* AR Viewer in viewing phase.

As the last step, by clicking button 'Build Bundle and Upload', users can build a Unity asset bundle and upload to Granlund AR Server. At this moment, the Granlund AR Viewer retrieves model from Granlund AR Server, instead of Autodesk Forge cloud. This is mainly because of the fact that Forge AR/VR toolkit is still in preview phase and according to the evaluation, it is not yet reliable enough to be integrated into the product. On the other side, In this solution, as the alignment is done within Unity editor, it makes it possible for users to use various kind of LoD (Level of Detail) plugins to optimize the model, and then they can upload the optimized model to Granlund AR Server, which is another reason why Granlund AR Server is considered as a mandatory component in the solution.

6.2.3 AR Viewing

For viewing the virtual content, user can open the application and tap 'Locate' button, when they reach one of the places where anchor was placed.

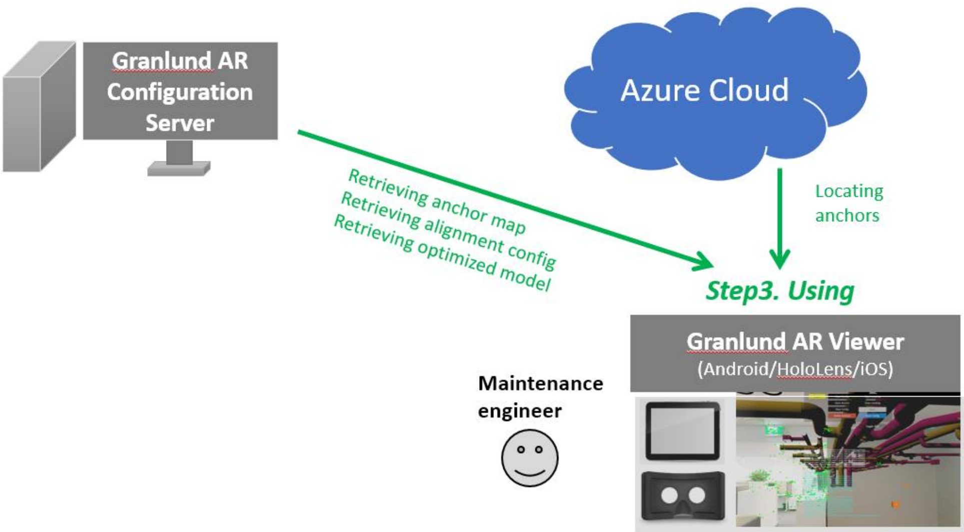

Figure 45. AR viewing process

Figure 46. Locating an anchor

Figure 47. Tap the anchor to launch visualization

When an anchor is located, user may notice the message at the bottom, saying 'Anchor located, tap the anchor to view the facilities. They may do so to launch the visualization of the model. Then users may tap any component on the model to view its information, including static information from BIM properties, dynamic information from building information system (BIS), which in the screenshot is the sensor statistics data from Siemens BIS, and the related maintenance tasks of the selected component.

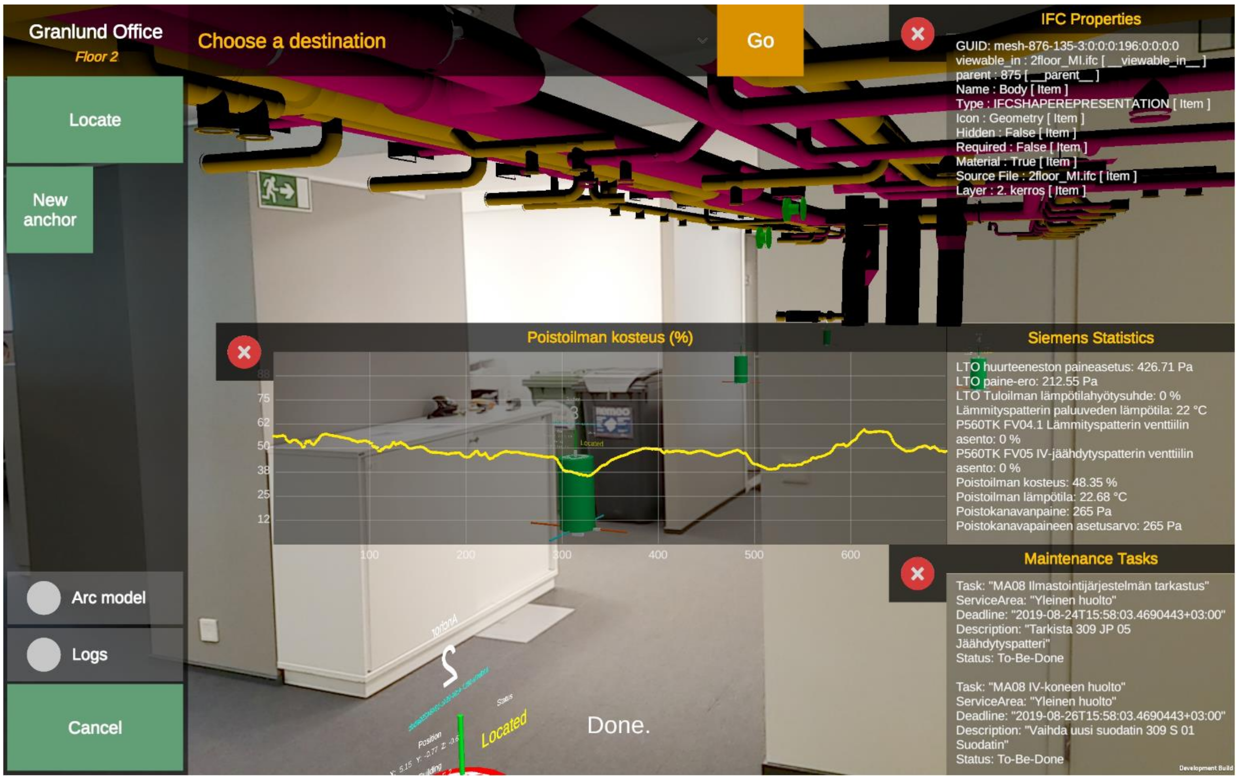

Figure 48. Viewing component static and dynamic information

Thanks to *ARCore* motion tracking, usually users can feel free to walk around without losing the tracking. But if the user moves too fast, or even the camera view is covered for a while, the tracking ca n be lost. When tracking is lost, for a place which has just been recognized in the same session, tracking can be restored automatically by *ARCore*, but anyway, user can always tap 'Locate' button to re-launch an AR recognition to fix the tracking.

Indoor navigation is also implemented with *NavMesh*. *NavMesh* is built-in Unity plugin for generating navigation path from a given model. The process of creating a *NavMesh* from the level geometry is called *NavMesh* Baking. The process collects the Render Meshes and in the scene. In this solution, this process can be done during the anchor-model alignment phase. The baked navigation path can be included into asset bundle of the whole model and gets loaded dynamically in AR viewing phase. When users recognize an anchor in AR viewing phase, since all the anchors will be located, it is possible to let the users to choose one of the anchors to be the destination. In this project, a navigation robot is implemented as a navigator and users may simply follow the robot navigator to reach the destination.

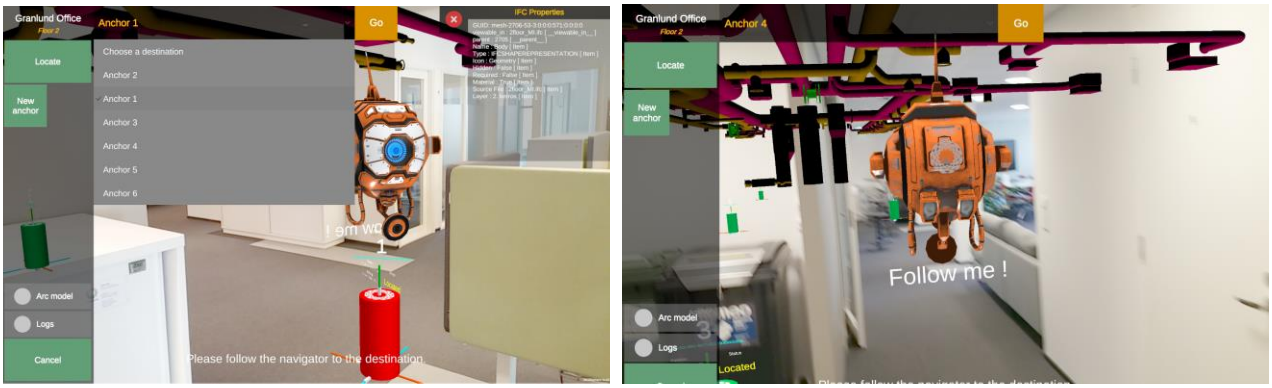

Figure 49. Users may choose one of the anchors as the destination follow the navigator robot to reach it.

6.3 Highlights

6.3.1 Highlights about the Solution Architecture

The whole system is created based on maintainable and extendable architecture, below are some of the advantages:

- **It can easily adapt to model changes**: Although it is rare that BIM models are changed after it's designed, but it is possible, thus we must handle it. When there is a new version of model, all that needs to be done is to re-build the asset bundle with *Granlund* AR Toolkit and upload it to Granlund AR Configuration Server. This can be done without involving maintenance engineers. When new model is available, it can be retrieved automatically when maintenance engineer login the application.

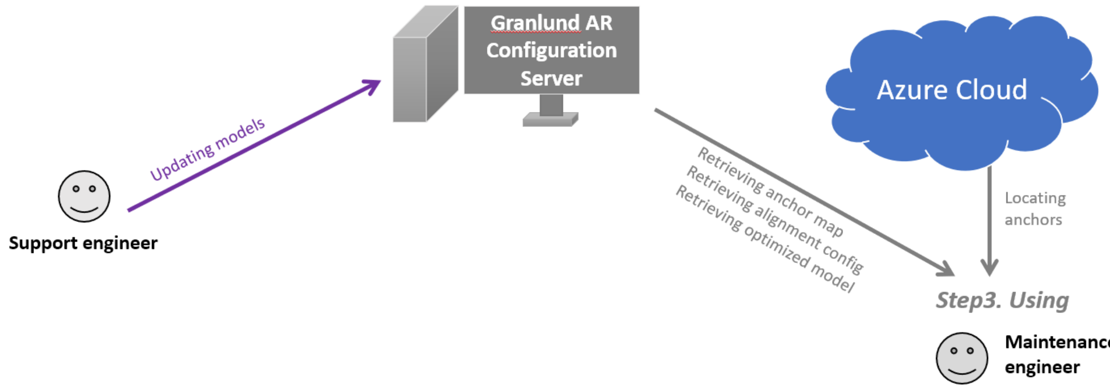

Figure 50. Adapting to model changes

- **It can easily adapt to physical environment changes**: Comparing to model changes, the physical environment has a greater chance to be changed, due to layout change or new decoration inside the building. When it happens, it only requires small effort from maintenance engineers that they need to replace the old anchor with a new one, then following the normal process, support engineers generate update the alignment configuration to Granlund AR configuration server and that's all.

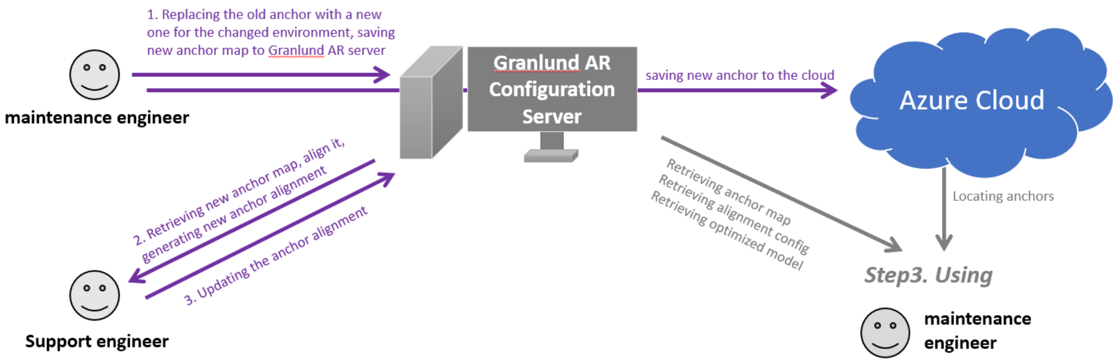

Figure 51. Adapting to physical environment changes

- **It is extendable for model optimization**: Model optimization is a critical issue for AECO projects. Especially for AR project, because the AR devices, including phones, tablets or smart glasses are all mobile devices, which has limited computing resources comparing to VR hardware, which can be supported with powerful graphic card. That is why most of the cloud-based model management service, such as *Autodesk Forge* and *Tridify* are trying to convert the traditional model format such as IFC, OBJ to an abstract and optimizable format in the cloud. However, besides the cloud-side optimization, it is always cheaper and more flexible to optimize models locally with various kind of LoD (Level of Details) tools in Unity.
- **It is extendable for indoor navigation**: Indoor navigation can be supported by integrating Unity's built-in plugin '*NavMesh'*, which can easily build intelligent path for a given model, which in our case, the architecture model of the building. *NavMesh* can be integrated with *Granlund* AR Toolkit to make the process even more easy for support engineers.

6.3.2 Highlights about the Technical Implementation

The diagram below shows the core classes and their relationship. The unity scene is mainly managed by *WorldManager* and its parent classes *ManagerBase* and *InteractionBase*. Those three class is responsible not only the initialization of the whole scene but also handing user input and dispatching events to the corresponding utilities classes. There are many different utilities for different tasks, such as that *CloudManager* focus on handling Azure cloud interaction, *AnchorManager* handles all the anchor related operation such as anchor creation, anchor moving, anchor saving and deleting, *UIManager* can be called for all kinds of user interface updates, such as the updates of messages and button status. There are two state machines implemented for managing the UI status and anchor status. Additionally, most of the utility classes are implemented in 'Singleton' design pattern to make sure that there is only one instance in the whole scene, and this also make it easy for any class to call the utility classes. However, as a rich communication application, frequent UI updates and Network request/response cloud conflict and block the whole process, if there is no proper communication rule. *QueueManager* is the common communication channel, which has an embedded queue, updated by Unity frames. All sorts of actions can be executed through *QueueManager*, especially the UI, I/O and Network related actions.

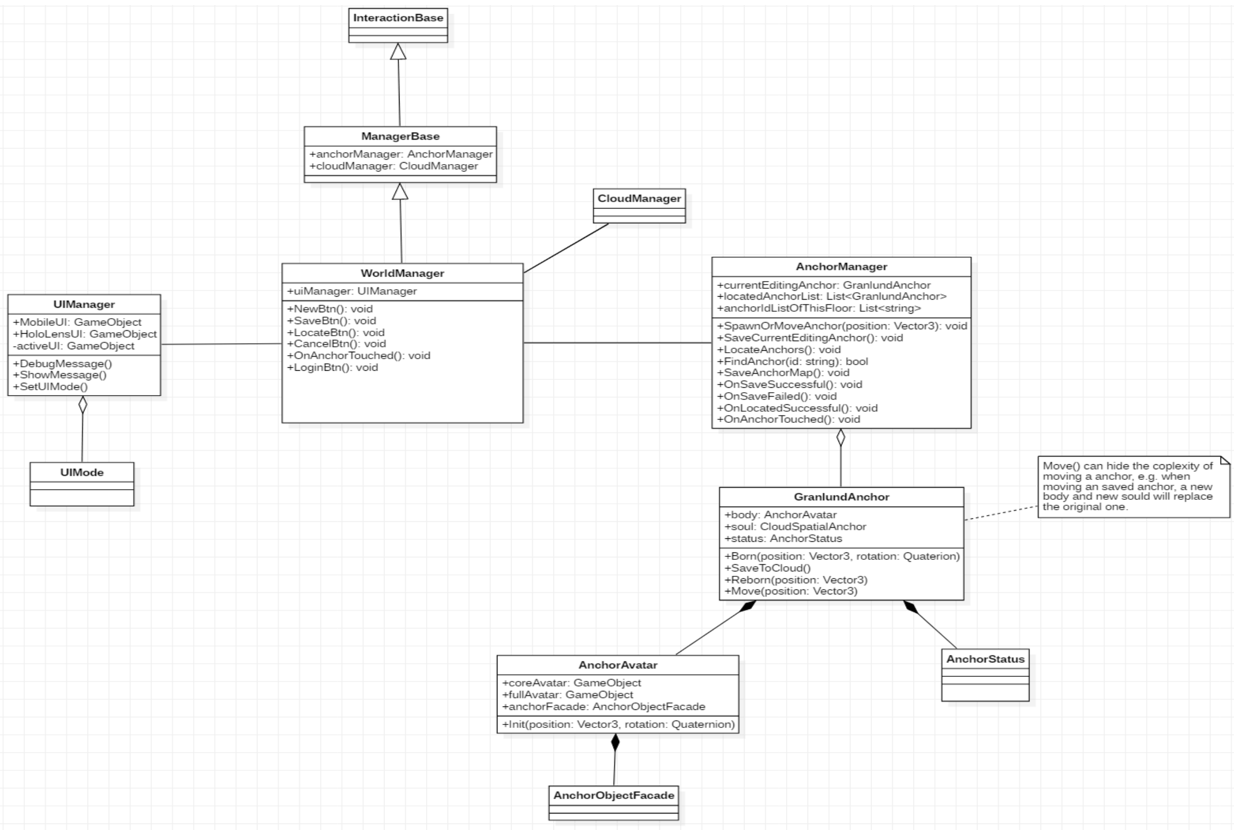

Figure 52. UML diagram: The core classes and their relationship with each other.

The AR anchors are the most complicated object to manage in the whole application. Although in this project, we are using *Microsoft* Azure spatial anchors, to make it easy to be maintained and extended, we created an abstract layer on top of the actual anchor technology. Class *GranlundAnchor* is an abstract anchor which defines the properties and behavior that an anchor should have, it is designed metaphorically that it has a 'body', which is a visible object, and it also has a 'soul', which an invisible Azure spatial anchor object. This kind of 'body' and 'soul' design metaphor is based on the behavior of Azure spatial anchor. As mentioned in earlier chapters, Azure spatial anchor is only the anchor located in the cloud side. The 'local' anchor is always an anchor of the platform, such as *ARCore* anchor, *ARKit* anchor and *HoloLens* spatial anchor. Therefore, we designed the soul-body metaphor that when a *GranlundAnchor* is spawned on the ground, it only has a 'body' without a 'soul', the 'soul', which is the Azure spatial anchor can be created when user save the anchor. When an anchor is located, it is 'born' on the world with a visible avatar, which is the anchor object.

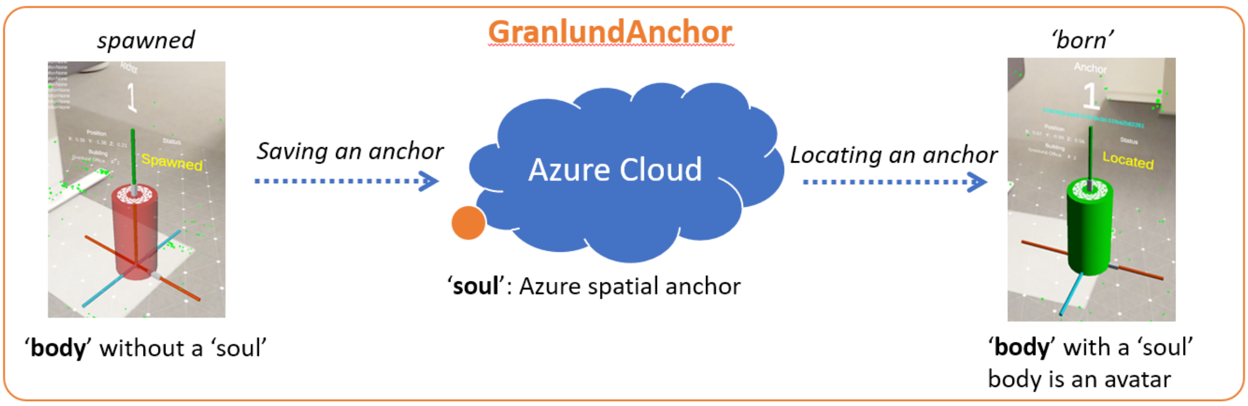

Figure 53. The 'body' – 'soul' metaphor of *GranlundAnchor*

This kind of metaphor is not only funny but also make it easy for us to manage the anchor operation. In detail, *GranlundAnchor* hides the detail of the complexity of the actual anchor operation. For example, for moving the position of an anchor, developers can just call method '*move*()' of *GranlundAnchor*. What actually happens in the background can be more complicated, if the anchor has both a body(*AnchorAvatar*) and a 'soul'(Azure spatial anchor), it has to be done in a sequence of deleting the 'soul', deleting the local anchor, moving body to a new position, adding a new local anchor, creating a new 'soul' and saving it to Azure cloud. This kind of complexity is hidden for developers so that the maintainability of the application can be improved.

In terms of the extendibility of the user interface, as the UI and logic are de-coupled, it would be easy to create another set of user interface for a different type of device, such as *iOS* devices or *HoloLens*.

6.3.3 Highlights about the Design or User Experience

6.3.3.1 The Design of Anchor Object

The anchor object is designed as a cylinder body with three bars which represents x, y, z axis, x in red, y in green and z in blue, which is identical to the color convention in 3D authoring software like Unity. The axis bars can be helpful for users to understand its pose while the cylinder object makes it easier for users to tap in the AR view. The anchor object has different look when it's in different status. When it's newly created on the ground, its cylinder body is in orange and it's rotating, which represents that it's in editing mode and not yet settled. When the anchor is saved and located, it turns green and the animation would be stopped. From the information on its panels, users can see its sequence number, ID (if it's saved), position, rotation and status.

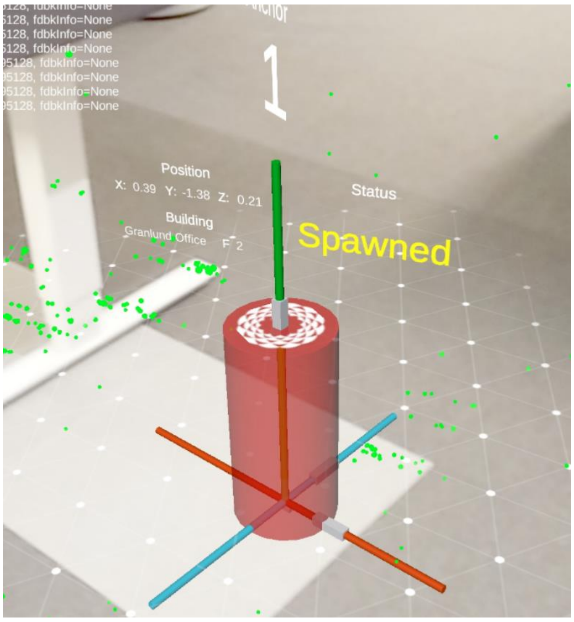

Figure 54. An anchor object when it's spawned

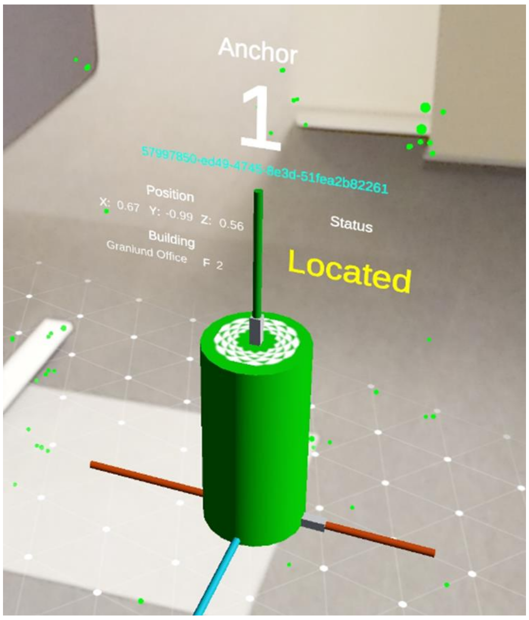

Figure 55. An anchor object when it's located

The size of the anchor object is designed in the consideration of forcing users to stay closer to the target area. It cannot be too small that it would be hard for users to tap on it, in contrast, it cannot be too big because then users can tap it from distance, which is not an expected behavior, because an anchor can only make sure that the virtual content is in the correct pose in the near area. Tapping on an anchor in distance may lead to wrong pose of the virtual content.

People may also ask why the anchors must be put on the ground, considering that the HVAC pipes usually locate behind the ceiling. Technically it is possible to place an anchor on any surface detected by *ARCore*, it can be on a horizontal surface like the ground or ceiling, or a vertical surface like a wall. However, there are two reasons why it's designed to be on the ground:

1. Usually the ceilings or walls do not have enough feature for an Azure spatial anchor to work well. The ceilings could like the same everywhere and the white walls are usually considered bad place for AR recognition.
2. Putting anchors on the ground can make the anchor-model alignment easier, because in this way we can make sure that all the anchors are having a same Y-position: zero. Combined with the anchor marks on the 2D floor plan, which only has the X and Z position information, it is enough to locate an anchor accurately. Otherwise, if the anchors' Y position is not fixed, it would be hard to determine its accurate position in the space when doing the alignment.

6.3.3.2 General UX Consideration

The user interface components such as labels, panels and buttons are designed to have an identical design language of *Granlund* application. All the buttons, including the buttons on the login view, the buttons on the control panel, and even the close buttons of properties panels, all of them are large enough for users to tap easily, this is an important usability consideration not only because it's common design practice for mobile applications, but also because the users are maintenance engineers, who usually wears cut resident gloves during the maintenance work, according to the interview records. Some of the gloves are touch screen friendly that they don't have to take it off for using the tablets, but it would make it harder to tap comparing to naked fingers directly.

7 Evaluation and Discussion

In this project, the implementation was done in three iterations. Each iteration focuses on different features. But it was also ensured that the outcome of every interaction is a complete mini product so that it can be evaluated.

- **Prototype 1**: A basic system with a simple framework was implemented. This prototype was mainly for technical feasibility study on *Azure* spatial anchors.

- **Prototype 2**: A simple but complete system was established including the AR configurator, *Granlund* AR Toolkit (Unity plugin) and AR viewer. This prototype can be considered as a minimal viable product. The goal was to evaluate whether the whole solution process works fine.

- **Prototype 3**: A new framework with better extendibility and maintainability was introduced, which enables us adding more features easily without breaking existing functions. On the other side, in this prototype, we evaluated the feasibility of integrating with building information system for dynamic information.

Ideally, the usability evaluation should be performed for each interaction. However, due to the schedule conflict with the summer holiday period, it was not possible to arrange comprehensive usability testing session until August. Therefore, for the first two prototypes, we only demonstrated to a limited scope of people for collecting feedback, and the formal evaluation was executed for the third prototype in August/2019.

7.1 Evaluation Process

Usability evaluation session was executed for collecting feedback for future improvement. As the whole process in this solution is divided into three phases, including AR configuration, anchor-model alignment and AR viewing. In practice, those three phases could be different user scenarios targeting to different user group. For example, the AR configuration phase and AR viewing phase are for maintenance engineers, while the anchor-model alignment is mainly for support engineers. Therefore, the evaluation should also be separated for those three phases, with different set of tasks and different questionnaire, involving different type of participants with different background.

Type	Participants	Tasks	Questions
Scenario A: **AR configuration**	Participants are expected to have basic understanding on Facility maintenance. No special requirement for AR.	Creating an anchor Saving an anchor Saving anchor map of the floor	The questions are mainly about how easy it is to create, modify and save anchors. (rating from 1-5)
Scenario B: **Anchor-model alignment**	Participants are expected to have experience on Unity	Importing anchor map into Unity Loading models	The questions are mainly about how easy or how complex the alignment process is. (rating from 1-

		Anchor-model alignment & generating alignment configuration Generating model asset bundles.	5)
Scenario C: **AR viewing**	Participants are expected to have basic understanding on Facility maintenance. No special requirement for AR.	Locating anchors Navigate to a different point-of-interest Select components to view its static and dynamic information	The questions are mainly about how easy and how reliable it is, when viewing virtual content and exploring the properties information (rating from 1-5)

The evaluation session was executed individually with seven people with different background knowledge. One of them have Unity experience and attended evaluation #2 while all people attended evaluation #1 and #3. Every session contains an introduction session and testing session, followed by a short interview and questionnaire. The questionnaires are designed with different set of questions for those three evaluation types, and some common questions are shared such as asking about the general feeling or impression for the user interface. The last question in the questionnaire is an open question which allowed participants to fill comments or suggestion freely.

7.2 Findings

Based on the result, the general feedback is that completing the tasks are not difficult. But for some users, the concept of anchors and point-of-interest are a bit confusing in the beginning.

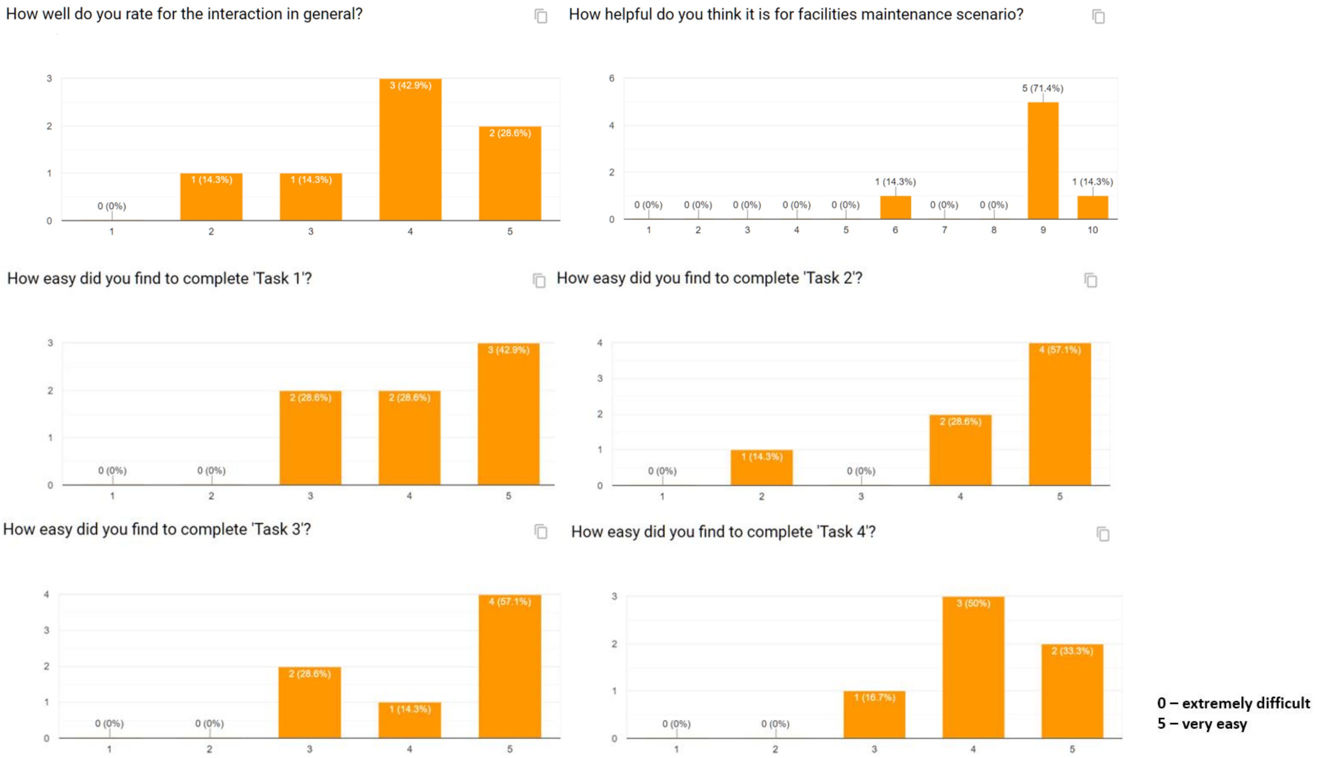

Figure 56. The usability evaluation results

There are many valuable comments are collected from the evaluation sessions. We summarized the main problems found in different categories with severity information in the table below. The severity values are defined as below:

- **A - Critical**: The problem can be considered as a usability breakdown, which blocks user to complete the task as expected. It must be fixed with high priority.
- **B - Major**: The problem is considered as a functional bug or design mistake, which is not a blocker issue but preferred to be fixed when time allowed.
- **C - Minor:** The problem only causes inconvenience or bad feeling when using the software.

ID	Category	Severity	Description	Comments & Ideas
1	AR configuration	C	When placing anchors on the ground, some users naturally want to move the anchor object by dragging, which is not possible.	It's easy to add dragging support as a small feature later.
2	Anchor-model alignment	A	Sometimes the maintenance engineers could forget marking anchors on the 2D floor plan in AR configuration phase, then for the missing anchor, it would be hard for support engineers to adjust and align in Unity (in anchor-model alignment phase)	This is the major limitation of the current version. The workaround in short-term is marking anchors pose on 2D floor plan, and a longer-term solution will be discussed in the next chapter.
3	Anchor-model alignment	B+	It is not easy to accurately align the anchor and the models in Unity. A small distance in Unity can lead to significant wrong alignment in the AR view.	This is the limitation of off-line alignment in authoring tools. Discussion about online alignment in runtime can be implemented, which is discussed in the next chapter.
4	Anchor-model alignment	B	The process is a bit complicated because user must use different tools for different steps. For example, the model downloading is done through Forge plugin. Integrating all needed software together into *Granlund* AR Toolkit would make the process much easier.	It is just a matter of time to integrate all needed tools into the *Granlund* AR Toolkit unity plugin.
5	AR Viewing	C	When switching floors, it takes a while (about 10 seconds) to load the model. It's better to make it faster or provide a process bar.	The root-cause is about the model size and the solution must be from model optimization, either from the cloud service (Forge) or from LoD tools in Unity. For the Forge size we cannot predict when it's going to provide satisfying model optimization, but at least in short-term we can integrate LoD tools into *Granlund* AR Toolkit.
6	AR Viewing	B-	AR tracking sometimes gets lost easily.	This is a limitation from the hardware capability of phones or tablets. AR glasses like HoloLens can hardly lost tracking, thanks to its 'environment understanding camera set', which includes fisheye camera and depth camera.
7	AR Viewing	A-	Users can easily get confused about what is anchor and why it's needed. Because users expect to view content (such as the	This is the major usability issue. Anchor is a technical word in AR world. We 'd better hide the concept of anchor with a

			model), not anchors.	user-friendly term such as 'maintenance point', or 'point-of-interest', 'viewing point', and somehow design a metaphor the make it reasonable to interact with anchor objects.
8	AR Viewing	C	Maybe it's not needed to show the whole model in AR, only the components near the user can be visualized.	Currently there is no optimization for showing the HVAC model, but it is possible to make it flexible to show part of the components according to user's position.
9	AR Viewing	C	Sometimes it's hard for users to realized that they need to move the tablet around to collect enough environment information either when creating an anchor or locating an anchor	An animation can be added for guiding users to move the devices.
10	AR Viewing	B	Sometimes users might not realize that anchor is drifted. How do we know if it's drifted or not without referring to the anchor pose marked on the 2D floor plan?	Currently there is no elegant solution than referring to the 2D floor plan. Later if we introduce a digital floor plan or digital 3D model on the tablet, that can be also referred.
11	General	C	Sometimes the message is updated too fast, users cannot complete reading the message.	The message showing feature can be optimized easily.
12	General	C	The AR virtual content looks unrealistic because it never gets occluded by physical objects.	This is since occlusion cannot be well handled by the current AR engine (e.g. *ARCore*). It is also due to the lack of depth camera on phones and tablets.

In the usability evaluation, all the anchors are configured before the evaluation session. However, there is a major limitation for anchor configuration, which is about the accuracy of anchor map. In a small area, the anchor map can be as accurate as it should be, but when it is larger than user needs to walk more than 30 meters away, the anchor map can be distorted because the position of the anchors far away is not accurate because of ARCore drifting, which is a noticeable problem on mobile devices when user keep walking for a distance. And this is considered as a limitation that prevents users from spreading anchors freely. To get an idea on how bad the drifting is, we measured the drifting number during anchor configuration, based on this we can have an idea of how large one area can be supported without losing the accuracy of anchor map.

To measure the drifting, we put three measurement points at distance of 0 meters, 15 meters and 30 meters. We record the original position and retrieve the located position from the anchor map. In figure 57, the original position is marked in green while the located ones are in red. From the figure we can see that the drifting is noticeable, and it is not liner, the drifting per meter increased when the distance increased.

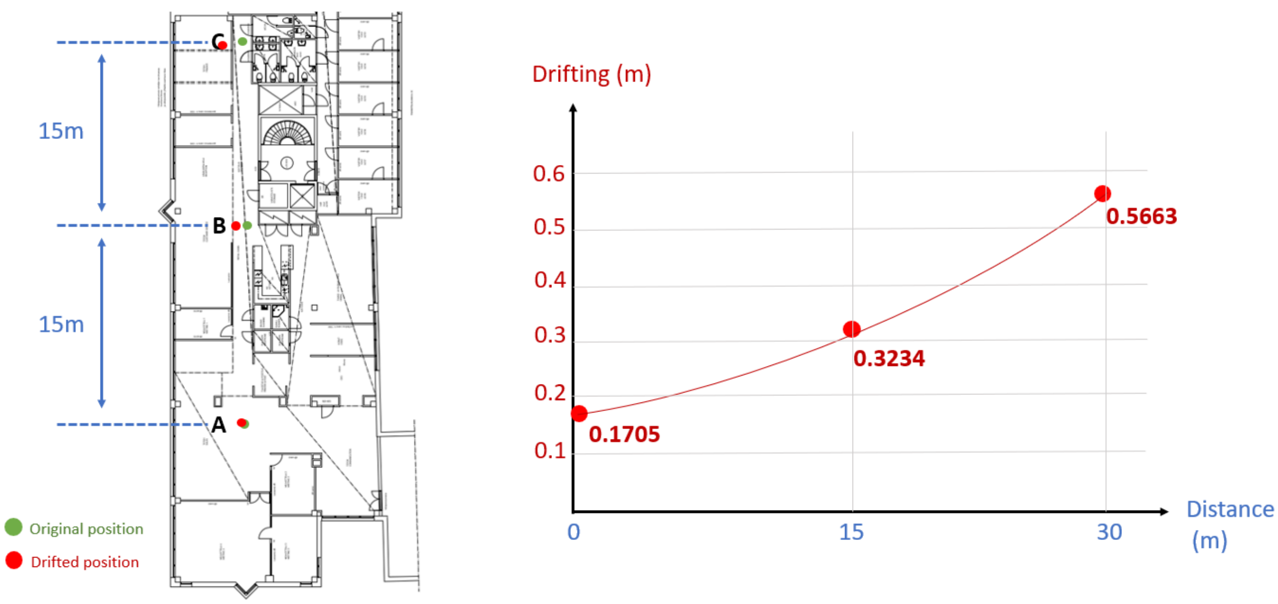

Figure 57. The ARCore drifting measurement

7.3 Discussion

The two most critical issue are finding #2 and finding #7.

- **Finding #7** is a design issue. Users find it difficult to understand the concept of 'Anchors', and they wonder 'why anchor is needed', and 'how is it related to the virtual content'. This is understandable, because users' expectation is simply opening the camera and viewing the content directly. The lesson learned from this finding is that 'Anchor' is an AR technical word, when designing a product which targets to normal audience, we need to hide the technical details. This problem can be easily solved by re-designing the metaphor and human-computer interaction.
    1. The concept of 'Anchors' should be hidden. Instead, the concept of 'Point-of-interest' should be introduced and highlighted to the end-user, and for maintenance context, a 'point-of-interest' can be a 'maintenance point', or 'service point', where users may find tools that can be used in AR view. One practical metaphor for the anchor object can be the lights for lighting up the model of the area. In the current prototype, lights are hard-coded and built with the model in Unity. Ideally, they should be provided dynamically, and only for the focus area according to user's position.

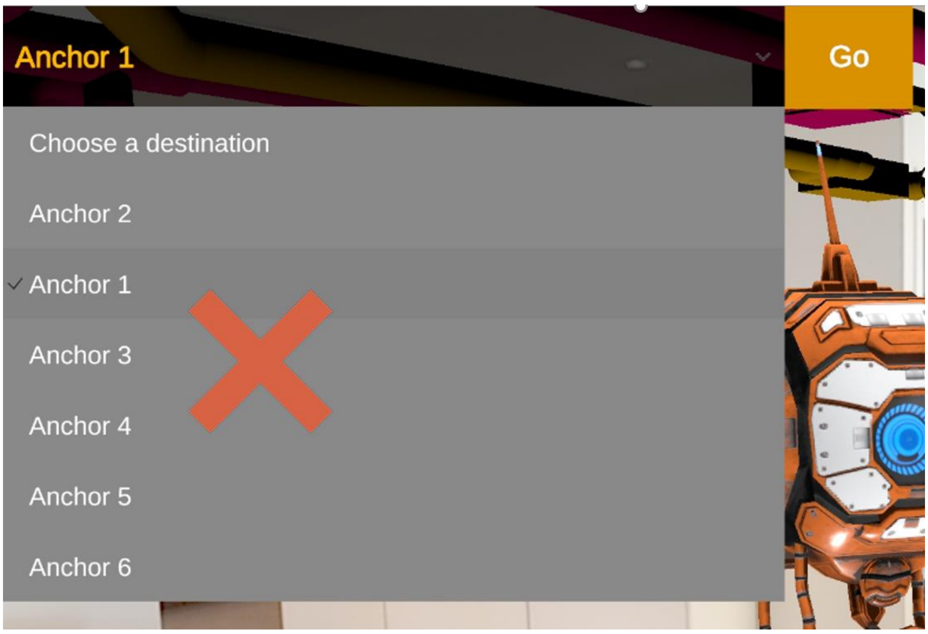

Figure 58. The concept of 'anchor' should be hidden, and replaced by the name of the 'point-of-interest'

2. In the current prototype, is requires users to manually tap the anchor to align the virtual content (HVAC model). And sometimes user do not know if the anchor is drifted or not. The whole process can be automated, and some detail should be hidden in the background. When user arrive a 'point-of-interest', the anchor locating can be executed automatically and the model can be aligned to the anchor automatically. This would greatly reduce the mental effort for the end user.

- **Finding #2** is about the difficulty on remembering the anchor position. A quick workaround for problem #2 can be automatically take a picture when saving an anchor, so that the picture can be referred during the alignment. However, in longer-term we need an elegant solution. The solution for problem #2 can be considered together with finding #3, which is about the difficulty on accurately align anchors and models. A quick workaround for finding #3 can be providing grids in both the AR view and Unity editor, grids can be used as rulers, and it would make the alignment easier and more accurate. For a long-term solution, those two findings made us re-consider about the offline alignment approach (anchor-model alignment in authoring time). Offline alignment within Unity sounds reasonable and we naturally borrow this approach from *ALVAR* process and *Immersal* process. But now there are two new ideas to be explored with online alignment approach:

  1. When anchors are placed in the physical environment, what if we enable maintenance engineers to mark the anchor on a digital model within the application right away? That can save the time of doing alignment offline in Unity.
  2. There is an even easier solution for maintenance engineers. During our investigation, we learned from *Trimble* solution that it provides a feature to align 1:1 model to the physical environment in AR view, even though it does not have AR recognition at all. What if we enable the same before anchor configuration? And then model can be hidden and when anchors are placed, its relative pose can be recorded automicticially, i.e. the alignment can be completed without the awareness of the user. Nevertheless, the feasibility of this idea still needs to be explored because the *Trimble* alignment feature is designed for *HoloLens*, it is still a question on how easy it is to do the same with tablets.

For other findings in severity B, most of them can be complemented and taken as normal feature development tasks. For example, for finding #4, it is just a matter of time to integrate all necessary tools together; for finding #6, the performance of AR tracking depends on the performance of Azure spatial anchor, or the AR hardware capability. e.g. the same code running on *HoloLens* should be more accurate and reliable than mobile devices.

For all the C-level issues, they can be easily solved except #5 and #12. Finding #5 is about model optimization. What we can do in future is to explore and integrate LoD plugins into *Granlund* AR Toolkit, or expect the Forge cloud provides better optimization, as it’s still be preview stage. Finding #12 about AR occlusion is a hot research topic, which can be expected to be solved by new AR engines.

8 Conclusion and Perspectives

In this project, we explored the feasibility of a practical Augmented Reality solution for facility maintenance scenario, and we made a small working prototype. Due to the schedule limitation, it is still far from perfect, but it can be considered as a good start, if we put it into a long-term product roadmap.

This project was executed following both a design approach and a technical engineering approach. However, the result clearly proves that the most critical blocker for building a practical Augmented Reality solution is still about the limitation of AR technology itself, which is reasonable, as AR is still an emerging technology. From the perspective of planning a long-term product roadmap, the below improvements can be expected for future works:

- **More reliable AR hardware**: The prototype in this project is built with a *Samsung* S4 tablet (Android), the option with *HoloLens*1 was dropped mainly because of its expensive price-tag and its outdated computing capability, which is troublesome especially when loading large AEC models. However, thanks to its extra hardware such the fish-eye camera and depth camera, its accuracy and reliability on AR is still far beyond the mobile devices. *HoloLens*2 sounds a powerful platform for doing feasibility research. Regardless of its price tag, at least some new interaction pattern brought by HoloLens2 can be explored for our context.
- **More mature AR engine (software)**: In this project, the core AR engine is from *Azure* spatial anchor and *ARCore*. As the *Azure* spatial anchor technology is still in preview stage, improvements on AR recognition accuracy can be expected in future. In addition, the capability of occlusion handling is a hot item on the 'expected feature list' of AR engines, and once it's enabled, it would make the virtual content in AR more realistic. However, it might require extra depth camera embedded, for retrieving the depth information for occlusion analysis. But the trend is clear that mobile device manufacturers may bring it to the future models and even the capability on Google Tango can be enabled by *ARCore* again.
- **Novel interaction between AR glasses and hand-held devices**: In practice, even when AR glasses are accessible for the user, tablets might be still important, for accessing traditional system for maintenance work, such as checking the floor plan, or generating maintenance report. The interaction pattern between AR glasses and hand-held devices is a new era to explore. An example use case can be that user tap a component on the tablet, the corresponding components visualized in AR can be highlighted.
- **Full integration with indoor navigation**: Indoor navigation is considered as an extremely beneficial feature to be integrated for facility maintenance context. The current prototype proves the feasibility together with AR technology. However, an accurate and reliable vision-based indoor navigation is not easy to achieve and might requires noticeable efforts to prepare. In addition, non-vision-based indoor navigation approach, such as beacon-based can be also considered, and then AR can only focus on the visualization and interaction. But again, such approach requires deployment of physical beacons, which is more costly and complicated to setup.
- **More integration with third-party system & more activities in AR**: In this project, we already prove the feasibility of the integration with third-party system such as the facility management system and building automation system. It can be complemented in future releases. In addition, based on reliable AR tracking, more interesting activities can be enabled, such as adding annotation or notes, and user guide for maintenance tasks.

Bibliography

[1]. NBIMS-US, NBIMS-US Fact Sheet (PDF), https://www.nationalbimstandard.org/files/NBIMS-US_FactSheet_2015.pdf, National BIM Standard - United States, 2015.

[2]. Gheisari M., Irizary J. (2014). "Exploring BIM and Mobile Augmented Reality Use in Facilities Management", Construction Research Congress 2014.

[3]. Wagner, D. & Schmalstieg, D. First steps towards handheld augmented reality. Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on, 18-21 Oct. 2005, 2003. 127-135.

[4]. Henrysson, A. & Ollila, M. UMAR: Ubiquitous mobile augmented reality. 2004. ACM, 41-45.

[5]. Irizarry, J., Gheisari, M., Williams, G. & Walker, B. N. 2012. InfoSPOT: A Mobile Augmented Reality Method for Accessing Building Information through a Situation Awareness Approach. Automation in Construction, 33, 11-23.

[6]. Gheisari, M., 2013. An Ambient Intelligent Environment for Accessing Building Information in Facility Management Operations; A healthcare Facility Scenario, PhD Thesis, School of Building Construction, Georgia Institute of Technology, Atlanta, GA, December 2013

[7]. Woodward C., Kuula T., Honkamaa P., Hakkarainen M., Kemppi P. (2014). "Implementation and Evaluation of a Mobile Augmented Reality System for Building Maintenance", Proceedings of the 14th International Conference on Construction Applications of Virtual Reality, 16-18 November 2014, Sharjah, UAE

[8]. Kelly G., Serginson M., Lockley S., Dawood N., Kassem M. (2013). "BIM for Facility Management: A Review and a case study investigating the value and challanges". Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, October 2013, London, UK

[9]. Lee, S., An, H., and Yu, J. (2012). An Extension of the Technology Acceptance Model for BIM-Based FM. Construction Research Congress, 2012, pp. 602-611.

[10]. BIM Task Group (2012). The Government Soft Landings Policy. http://www.bimtaskgroup.org/gsl-policy-2 (last accessed Apr 5th, 2013).

[11]. Becerik-Gerber, B., Jazizadeh, F., Li, N., and Calis, G. (2012). Application areas and data requirements for BIM-enabled facilities management. Journal of construction engineering and management, vol 138 (3), pp. 431 - 442.

[12]. Masoud Gheisari Javier Irizarry, (2016), "Investigating human and technological requirements for successful implementation of a BIM-based mobile augmented reality environment in facility management practices", Facilities, Vol. 34 Iss 1/2 pp. 69 – 84

[13]. Eastman, C. M., Teicholz, P., Sacks, R. and Liston, K. (2008), BIMHandbook, WileyOnlineLibrary, NewYork, NY.

[14]. Deuter A. and Pethig F. (2019). The Digital Twin Theory - A New View on a Buzzword. 18. March 2019, München.

[15]. Shafto, M.; Conroy, M.; Doyle, R.; Glaessgen, E.; Kemp, C.; LeMoigne, J.; Wang, L.: Draft modeling, simulation, information technology & processing roadmap, Technology Area (11), 2010

[16]. Grieve, M.: Virtually perfect: Driving Innovative and Lean Products through Product Lifecycle Management. Cocoa Beach, USA, 2011

[17]. Rosenberg, L.B. (1992). "The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments". Technical Report AL-TR-0089, USAF Armstrong Laboratory, Wright-Patterson AFB OH, 1992.

[18]. Kato, H., Billinghurst, M. "Marker tracking and HMD calibration for a video-based augmented reality conferencing system.", In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99), October 1999.

[19]. ALVAR: "a library for virtual and augmented reality." VTT Technical Research Centre of Finland Ltd., http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html. Last release Aug 2019.

[20]. Vuforia Engine, PTC Inc https://developer.vuforia.com/. Last release June 20, 2019.

[21]. ARCore, 2019 Google LLC. https://developers.google.com/ar/

[22]. Motion tracking in ARCore, Google LLC https://developers.google.com/ar/discover/concepts, Last updated May 7, 2019.

[23]. Uimonen M., Hakkarainen M. Accessing BIM-related information through AR, IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct) 2018.

[24]. Cowen M., Lemon A., Gill-Hesselgrave D., User-Centered Design (UCD) Process Description. Space and Naval Warfare Systems Center Pacific (SSC Pacific), San Diego, CA. Code 53621, December 2014.

[25]. Gulliksen J., Lantz A., Boivie I., User Centered Design in Practice - Problems and Possibilities (A summary of the 1998 PDC & CSCW workshop), SIGCHI, April 1999

[26]. Mackay W., Breakdown analysis. HCI Bootcamp 2018 Handouts (PDF) https://ex-situ.lri.fr/content/7-workshops/5-hci-bootcamp-2018/1-handouts/doit-book-method-breakdowns-oct.pdf, Last update 2018.

[27]. Mackay W., User Scenario, HCI Bootcamp 2018 Handouts (PDF) https://ex-situ.lri.fr/content/7-workshops/5-hci-bootcamp-2018/1-handouts/doit-book-method-use-scenario-oct.pdf, Last update 2018.

[28]. Common Air Handler Problems & Maintenance Schedules. High Performance HVAC (website) https://highperformancehvac.com/common-problems-air-handlers/. Last update 2017.

[29]. Koch C., Neges H-M., König M., Abramovici M., BIM-based Augmented Reality for Facility Maintenance using Natural Markers, Proc. of the 2012 EG-ICE International Workshop on Intelligent Computing in Engineering, January 2012

[30]. Fire Protection safety policy, Occupational Safety and Health Administration (OSHA). United States Department of Labour (website) https://www.osha.gov/pls/oshaweb/owadisp.show_document?p_id=9811&p_table=standards. Last update 2019.

[31]. Trimble *SiteVision*, Trimble Inc. (website) https://sitevision.trimble.com/

[32]. *Pokemon* Go, 2019 Niantic, Inc. (website) https://pokemongolive.com/en/

[33]. ALVAR Point cloud resource target, VTT http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/doc/alvar-unity/index.html#pointcloud-resource

[34]. Immersal AR SDK, Immersal, Ltd. https://immersal.com/

[35]. Google Project Tango (Augmented reality computing platform), https://en.wikipedia.org/wiki/Tango_(platform). Last update Dec 2017.

[36]. Autodesk Forge., 2019 Autodesk Inc. https://forge.autodesk.com/

[37]. Tridify BIM to XR Cloud Solution, 2019 Tridify Ltd., https://www.tridify.com/

[38]. ANSI Z87.1 Standard – Safety Glasses, American National Standards Institute (ANSI). United States. https://www.creativesafetysupply.com/articles/ansiz871safety-glasses/. Last update 2019.

[39]. CSA Standard Z94.3-02 Eye and Face Protectors, Standard Information Sheet. CSA Group. Last update 2019.

[40]. Trimble XR20 with HoloLens 2, Trimble Inc, https://mixedreality.trimble.com/, 2019.

[41]. Developing Mobile Augmented Reality (AR) Applications with Azure Spatial Anchors, Microsoft Build 2019, https://azure.microsoft.com/en-us/resources/videos/build-2019-developing-mobile-augmented-reality-ar-applications-with-azure-spatial-anchors/, May 2019.