Project Acronym:	**BIMERR**
Project Full Title:	**BIM-based holistic tools for Energy-driven Renovation of existing Residences**
Grant Agreement:	**820621**
Project Duration:	**42 months**

DELIVERABLE D8.3
Integrated BIMERR ICT System 1

Deliverable Status:	**Final**
File Name:	**D8.3 Integrated BIMERR ICT System 1-v1.0.docx**
Due Date:	**28/02/2021 (M26)**
Submission Date:	**26/02/2021 (M26)**
Task Leader:	**FIT (T8.3)**

Dissemination level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

The BIMERR project consortium is composed of:		
FIT	Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Germany
CERTH	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis	Greece
UPM	Universidad Politecnica De Madrid	Spain
UBITECH	Ubitech Limited	Cyprus
SUITE5	Suite5 Data Intelligence Solutions Limited	Cyprus
HYPERTECH	Hypertech (Chaipertek) Anonymos Viomichaniki Emporiki Etaireia Pliroforikis Kai Neon Technologion	Greece
MERIT	Merit Consulting House Sprl	Belgium
XYLEM	Xylem Science and Technology Management Gmbh	Austria
CONKAT	Anonymos Etaireia Kataskevon Technikon Ergon, Emporikon Viomichanikonkai Nautiliakon Epicheiriseon Kon'kat	Greece
BOC	Boc Asset Management Gmbh	Austria
BX	Budimex Sa	Poland
UOP	University of Peloponnese	Greece
UEDIN	University of Edinburgh	United Kingdom
NT	Novitech As	Slovakia
FER	Ferrovial Agroman S.A.	Spain
UCL	University College London	United Kingdom

Disclaimer

BIMERR project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission (EC). EC is not liable for any use that may be made of the information contained therein.

AUTHORS LIST

Leading Author (Editor)			
Surname	**First Name**	**Beneficiary**	**Contact email**
Devasya	Shreekantha	FIT	shreekantha.devasya@fit.fraunhofer.de

Co-authors (in alphabetic order)				
#	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1	Bosché	Frédéric	UEDIN	f.bosche@ed.ac.uk
2	Bountouni	Nefeli	SUITE5	nefeli@suite5.eu
3	Demeter	Dominik	NT	demeter_dominik@novitech.sk
4	Falcioni	Damiano	BOC	damiano.falcioni@boc-eu.com
5	Fenz	Stefan	XYLEM	fenz@xylem-technologies.com
6	Giannakis	Giorgos	HYPERTECH	g.giannakis@hypertech.gr
7	Katsos	Marios	HYPERTECH	m.katsos@hypertech.gr
8	Katsigarakis	Kyriakos	UCL	k.katsigarakis@ucl.ac.uk
9	Krauß	Veronika	FIT	veronika.krauss@fit.fraunhofer.de
10	Lampathaki	Fenareti	SUITE5	fenareti@suite5.eu
11	Lilis	Georgios	UCL	gnl2@cornell.edu
12	Straka	Martin	NT	straka@novitechgroup.sk
13	Tavakolizadeh	Farshid	FIT	farshid.tavakolizadeh@fit.fraunhofer.de
14	Tsakiris	Athanasios	CERTH	atsakir@iti.gr
15	Vafeiadis	George	UBITECH	gvafeiadis@ubitech.eu
16	Valero	Enrique	UEDIN	e.valero@ed.ac.uk
17	Varga	Ján	NT	varga@novitechgroup.sk
18	Zarbouti	Dimitra	UOP	dzarb@uop.gr

REVIEWERS LIST

List of Reviewers (in alphabetic order)				
#	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1	Fenz	Stefan	XYLEM	fenz@xylem-technologies.com
2	Vafeiadis	Georgios	UBITECH	gvafeiadis@ubitech.eu

REVISION CONTROL

Version	Author	Date	Status
0.1	Shreekantha Devasya, Farshid Tavakolizadeh	03/12/2020	Initial TOC
0.2	Veronika Krauß	10/12/2020	Added Unified UI designs and UX alignments chapter
0.3	Shreekantha Devasya, Farshid Tavakolizadeh and other partners	02/02/2021	Initial contribution and aggregation of the contents to all the chapters from internal wikis and issue tracking tools. Draft for internal review
0.4	Stefan Fenz, Georgios Vafeiadis	24/02/2021	Final Draft reviewed
1.0	Shreekantha Devasya, Giorgos Giannakis	25/02/2021	Submission to the EC

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The current deliverable is the first version of the three deliverables related to the Task 8.3 (end-to-end ICT system integration testing and refinement). These deliverables document the BIMERR integration testing and refinement activities that are performed during the pre-validation and demonstration activities. The deliverables also describe the access control and unification of the user experience across BIMERR applications.

As a preliminary step of the integration testing process, individual software components developed and being used as part of WP4-WP8 are identified and details about their version management, delivery and deployment process are gathered. BIMERR involves different services and applications intended for different target platforms and having different business models. Therefore, the responsibilities of packaging and delivery are restricted to individual component development tasks. The current deliverable lists the overall picture of version control, delivery, and deployment of individual components.

As part of the system integration testing, the functional operation of the end-to-end system is verified with the right information flowing across different services and applications in a secured manner. Validation of the information flow and end to end integration use BIMERR use cases as the foundation. These use cases are defined in the early stages of the BIMERR project and went through further refinements. The interactions between different components towards the achievement of objectives for each use case are being validated, either manually or programmatically or with the help of demonstrations.

Securing the BIMERR system, its components and data from unauthorized access is ensured by integrating these components with the BIMERR identity provider. BIMERR uses the OpenID Connect protocol for authentication and secure exchange of profiles among users and applications. The deliverable describes how different roles and groups are mapped to define the access rules to ensure authorization.

To ensure that various software components provide a coherent experience for end-users, a simple style guide has been created. The guides include recommendations on the color schema, icons and typography to be used in the applications.

1. INTRODUCTION

1.1 SCOPE OF THE DELIVERABLE

The deliverable documents the initial activities covered in Task 8.3:

- Packaging and delivery methods of the individual BIMERR components being integrated as part of the BIMERR system. In addition, this document addresses the requirements each component should fulfil to be integrated and deployed in the production environments.
- Specification and planning of the system integration testing and driving the activity. The flow of information between the components is validated by keeping different use cases (UC) of the BIMERR system as the baseline.
- Ensuring the secure information exchange between the integrated components by providing a uniform access control mechanism.
- Specification of the user interface designs so that a consistent experience is provided to the users across the different applications of the BIMERR system.
- Refining of the individual components to accommodate important changes that may arise on short notice during the demonstration activities.

Being the initial edition of the deliverable series (D8.3, D8.4 and D8.5), this deliverable covers all the aforementioned activities except for the refinement activities which will start after M30.

1.2 STRUCTURE OF THE DELIVERABLE

The deliverable starts with the introduction of the overall BIMERR architecture. The integration methodology followed in BIMERR is described in Chapter 2. In Chapter 3, we list the packaging and delivery strategies followed by the different BIMERR components and the requirements each component should follow to be deployed in production environments. This is followed by a description of validation tasks based on the BIMERR use cases (Chapter 4).

Chapter 5 describes and specifies how uniform access control is achieved in the integrated BIMERR system. Finally, Chapter 6 provides guidelines for uniform UI designs and consistent user experience across BIMERR applications.

Sequence diagrams which were modified after M20 are provided in the annex of this deliverable.

1.3 Relation to the Other Activities and Deliverables

D3.1 (Stakeholder requirements for the BIMERR system) introduces 16 use cases describing the utilization of the BIMERR system. This deliverable uses these use cases as the baseline for validating the integrated system.

The current document uses the deliverable D3.6 (BIMERR system architecture 2nd version) as the basis for the BIMERR software and the information flow between the components. The deliverable provides the overall BIMERR architecture and gives a brief introduction to each component that is developed, enhanced and used as part of the BIMERR project.

The tasks responsible for the development of components as part of WP4-WP8 and their corresponding deliverables are also connected to the current deliverable and they are listed in Table 3-1 of this document.

Task 8.4 related to the BIMERR pre-validation activities help in the validation of the integrated BIMERR system and provides a constant feedback to Task 8.3. Hence the deliverable related to Task 8.4, D8.6 (Report on BIMERR pre-validation activities) which is due in M30 will give an overview of the pre-validation activities of the integrated BIMERR system.

Task 9.3 and Task 9.4 related to the rollout, deployment for demonstration and validation activities provide an overview of the performance of the BIMERR software in the project pilot sites. The feedback from these activities helps in validating the integrated BIMERR system. Hence their corresponding deliverables D9.3 (Report on BIMERR demonstration activities) and D9.4 (Holistic evaluation of BIMERR system performance and impact assessment) which are due in M40 and M42.

1.4 Overview of the BIMERR System

Before delving deep into the end-to-end BIMERR system integration, it is important to provide a high-level overview of the BIMERR system and overall flow of information across the components (see Figure 1-1). The BIMERR system is composed of the BIMERR Middleware, the BIMERR Interoperability Framework (BIF), the BIM Management Platform, the BIMERR Renovation Decision Support System (RenoDSS), the Digital Building Model Creation Tools and the Process & Workflow Modelling and Automation (PWMA) toolkit.

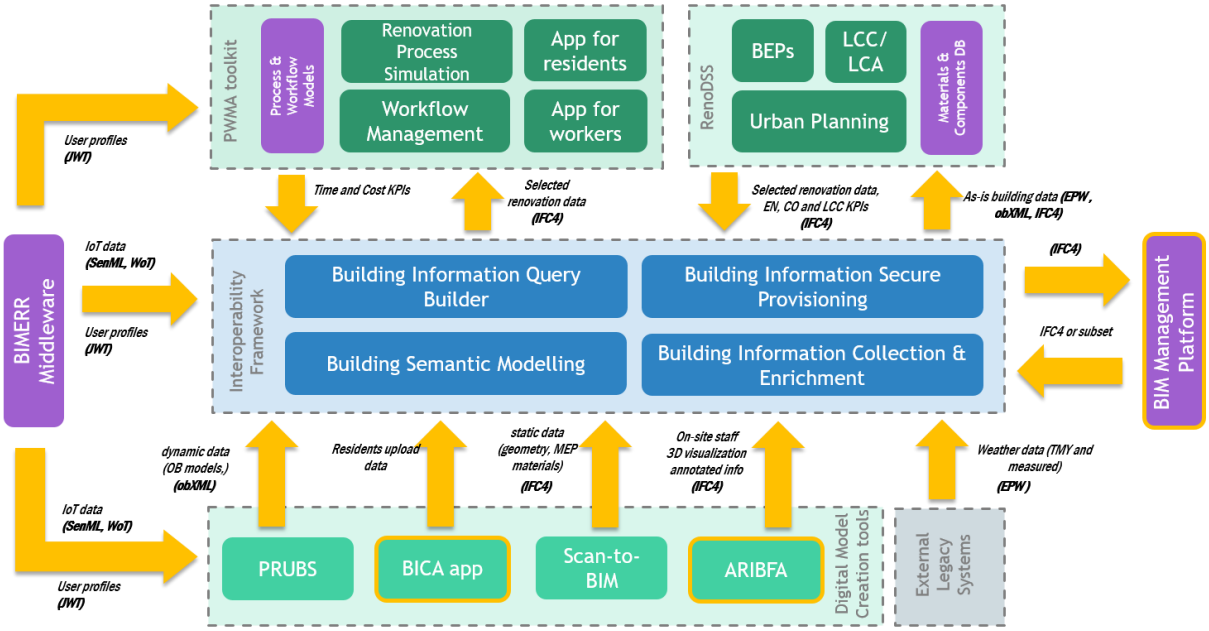

Figure 1-1 High level view of components and the data flow (taken from D3.6)

The BIMERR middleware handles the IoT data, processes, aggregates and stores them in a secure data store. The data is used directly by the Digital Model Creation tools whenever needed. It also acts as the central Identity Provider, providing information about the user profiles. The IoT data are processed by the PRUBS component to generate systems usage profiles which mimic the occupants' behavior, also known as dynamic data in terms of the Building Energy Performance simulation. Static data refers to the geometry, HVAC components and building materials etc. Scan-to-BIM algorithms are applied to the static data to generate the as-is IFC4[1] file that is uploaded to the BIF to be forwarded to the BIM Management Platform. There, data completeness, correctness/consistency checks and cleansing are performed, while the resulted IFC4 file, or subsets of it are sent back to the BIF. Subsets of the IFC4 file are used by the BICA application to allow the occupants to upload information that cannot be captured by any other means and by ARIBFA that supports on-site staff to annotate further information that the Scan-to-BIM did not manage to generate. Additionally, external legacy systems send weather data to the BIF. When dynamic, static and weather data are available, the RenoDSS queries them and initiates the generation and evaluation of renovation scenarios, which are combinations of renovation measures. Afterwards, the RenoDSS user

[1] https://standards.buildingsmart.org/IFC/RELEASE/IFC4/FINAL/HTML/

selects the scenario that meets his/her requirements and KPIs. The IFC4 file of the selected renovation scenario captures the renovation actions and is sent back to the BIF to be queried by the PWMA toolkit. PWMA toolkit is in charge of reporting back time and cost KPIs and the progress of the renovation tasks. Apparently, the BIF is the central data hub of the system, receiving data from the main components of BIMERR (Middleware, Digital Model Creation Tools, PWMA and RenoDSS) as well as external legacy systems. These data are semantically linked and stored in appropriate data models, enriched by the BIF sub-components and then propagated to the relevant recipient components and applications as needed. The main data model employed to describe a BIM model is IFC4. Handling, validating, and managing the internal structure of the BIM model is handled by the BIM Management Platform.

2. THE INTEGRATION AND TESTING METHODOLOGY

The BIMERR components developed as part of development activities related to WP4-WP8 are integrated in Task 8.3. The purpose and the design of individual components and their overall role in BIMERR is described in D3.6. The individual BIMERR components have varying packaging, delivery and deployment strategies. Considering the versatility of these components and considering the intellectual property rights of the partners on these components, it is practical to let the partners define the delivery strategies of these components. The packaging and delivery of individual BIMERR components is discussed in Chapter 3. As part of Task 8.3, we define the validity of the integrated BIMERR system.

The plan for the BIMERR software testing methodology is described in Figure 2-1. We validate different levels of abstraction of the BIMERR software at different testing stages. We identify the stages as: component development, system integration, pre-validation and final validation where we perform component testing, integration testing, system testing and user acceptance testing. These phases are derived from the specifications defined by Baresi and Pezze, 2006. This is an iterative process, and the process shall be repeated multiple times based on the changing requirements with time.

Component Testing

As a first step, each component is tested individually before integration. The component is validated against both functional and non-functional requirements defined in Section 4 of D3.6. Some of the additional criterions for each BIMERR components are well described in Section 5.2 of D3.6. These include and are not limited to well defined interfaces, documentation covering the different aspects of the component, definition of component delivery process and component testing. Result of the component development and component testing will be a complete functional component ready to be integrated with the BIMERR system.

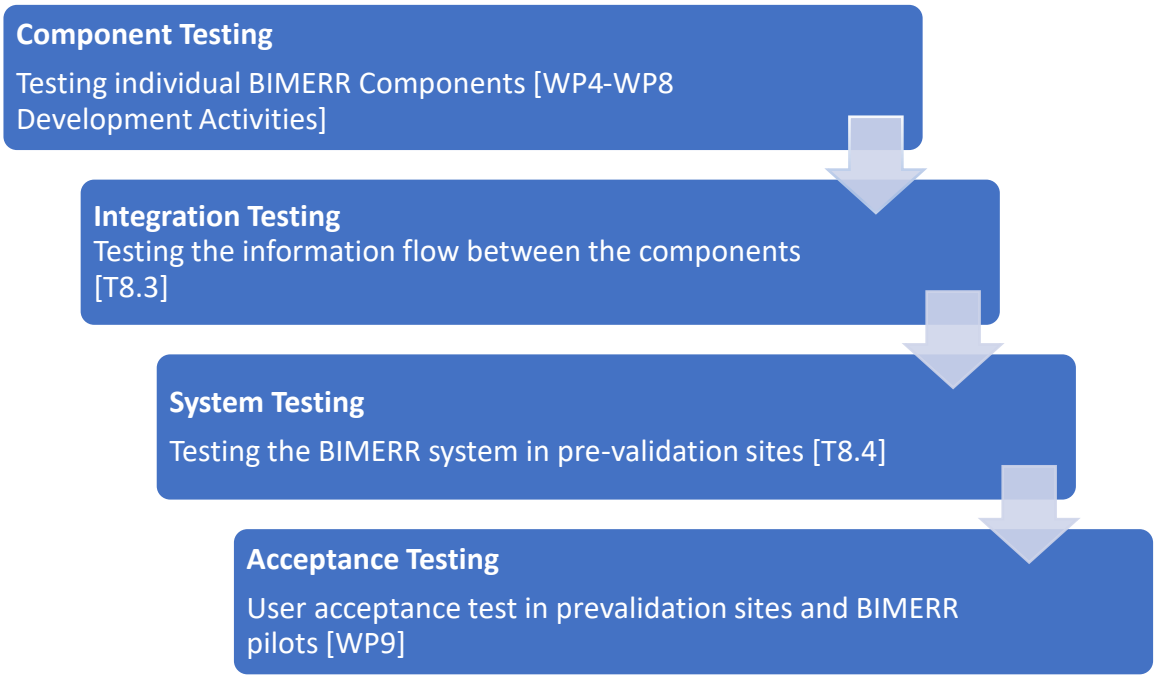

Figure 2-1 Different stages of testing and corresponding tasks

Integration Testing

Integration testing validates the interaction between different components. This phase identifies the failures caused by unexpected interactions between individual components. In the BIMERR project, the interactions are based on the interaction sequence diagrams designed for each use case. Each interaction is validated manually or using automated scripts. In the current deliverable and its upcoming iterations, we shall focus on the integration testing. The result of the integration and integration testing is a BIMERR system ready for system testing.

System Testing

Integration testing gives the confidence over the interaction between individual components. This is not enough to assess the behavior of the overall system, which is performed as part of system testing. In case of BIMERR, pre-validation sites act as the testbeds for system testing of the overall BIMERR system. Two pre-validation sites are identified for the project, namely, CONKAT and KRIPIS. The issues raised during the pre-validation activates help the BIMERR team to refine the components and overall BIMERR system. The result of the pre-validation activities shall be recorded in D8.6. The pre-validation activities help in creating a refined BIMERR system ready for acceptance testing.

Acceptance Testing

System testing validates the system against its specifications whereas acceptance testing helps us assessing how the system is capable of fulfilling the user expectations. The stakeholders related to the Pre-validation sites and the pilot sites provide continuous feedback about the overall BIMERR system with respect to various performance indices and these feedbacks shall be used to improve the system in different levels. The result of these activities shall be covered in D9.3 and D9.4.

2.1 INTEGRATION TESTING

Integration testing is performed when all the individual components of a system are combined to form a working system. Testing is performed between the components emphasizing their interfaces rather than the functionality of each modules (Leung and White 1990).

As mentioned earlier, the sequence diagrams of the use cases are used as the baselines for integration testing. These use cases are taken from D3.1 and D3.6. Each component, once ready, is validated against the use cases for their usability and interaction with other related components.

Considering "A receives X from B" as the interaction that has been implemented and needs to be verified through one the methods described in Table 2-1, it is possible to have a single procedure to verify multiple interactions. It is also possible to combine few of the methods to test interactions that are high level and involve more than one interaction internally.

Table 2-1 Verification means for interactions.

Method	Description
Automated testing	A scripted test: 1. Takes a reference R (identical or similar to X) as input. 2. Triggers the interaction programmatically such that A receives X from B. 3. Checks whether R is identical or similar to X. The test script should be executable by other partners and produce predictable and successful results.
Manual testing	Similar to automated testing except that it will be performed by hand. The procedure for manual testing should be documented. It should: 1. Describe the expected result R of this interaction. 2. Describe how to trigger the interaction such that A receives X from B. 3. Describe how the actual result X should be compared with R. The steps should be reproducible by partners with the necessary technical background.

Demonstration	Similar to manual testing, but instead of describing the testing procedure, there will be a recorded visual walkthrough to a live audience or an undoctored pre-recorded screen capture. The video recording should correspond to an existing version of the software.

2.2 TRACKING THE INTEGRATION TESTING ACTIVITIES

Integration testing activities are tracked by using the Jira[2] software. Each use case is considered as a Jira Task. The partners leading the use case realization create different subtasks for each integration activity associated with the use case and assign them to the developer responsible for the involved BIMERR components. The developers verify the interactions between the integrated BIMERR components using the methods described in Table 2-1 and record the steps of reproduction in Jira. The subtask is considered to be done once the verifications are performed and documented in Jira.

Once all the subtasks related to the use case are done, the main task related to the use case is also considered done. The verification of interactions among BIMERR components involved in different use cases marks the completion of the first iteration. Further iterations are possible based on the feedback from the system testing and user acceptance testing.

[2] https://www.atlassian.com/software/jira

3. PACKAGING, DELIVERY AND DEPLOYMENT OF BIMERR SYSTEM

The BIMERR system artefacts that are developed as part of WP4-WP8 follow different version control, licensing, and software delivery approaches. The current chapter lists the BIMERR components and their packaging strategies. The requirements for these components before the integration are described in Section 5.2 of D3.6. Deployment requirements for these components are described in Section 3.3 of this chapter.

3.1 OVERVIEW

Before going through the delivery strategies of each component, the overall strategies these components follow are described.

Source Code Availability

The source codes of the BIMERR components are maintained by the partners responsible for the components. The majority of BIMERR components are closed source and only a few of them are open source. Some of them plan to make their code public in the future.

Software Delivery

BIMERR components are delivered in one of the following forms:

- Docker image that can be downloaded from a docker repository
- Binary distribution compatible with the platform where the component is meant to be deployed
- Software as a Service (SaaS)
- Mobile application (as a downloadable app package or available in the marketplace)
- HoloLens[3] App

Documentation

Documentation of the BIMERR components is found in one of:

- Wiki page of a public repositories, if applicable

[3] https://www.microsoft.com/en-us/hololens

- BIMERR Deliverables corresponding to the components
- Project's internal wiki pages accessible to the project partners

3.2 VERSION CONTROL AND DELIVERY OF BIMERR COMPONENTS

Table 3-1 lists all components developed and used in BIMERR.

Table 3-1 Packaging, delivery, and documentation of BIMERR components

Component	Responsible partner	Delivery (distribution) form	Source code repository	Documentation	Software License
BIF: Building Semantic Modelling	SUITE5	Software as a service (SaaS)	Closed source on private GitHub repositories Note: UPM component (OMF) is available as open source	Deliverables D4.4 and D4.5	Closed source
BIF: Building Information Collection & Enrichment	SUITE5	Software as a service (SaaS)	Closed source on private GitHub repositories Note: UPM component (KGG) is available as open source	Deliverables D4.6 and D4.7	Closed source
BIF: Building Information Query Builder	UBITECH	Software as a service (SaaS)	Closed source on private Gitlab repository	Deliverables D4.8 and D4.9	Closed source
BIF: Building Information Secure Provisioning	UBITECH	Software as a service (SaaS)	Closed source on private Gitlab repository	Deliverable D4.8 and D4.9	Closed source

Component	Responsible partner	Delivery (distribution) form	Source code repository	Documentation	Software License
Scan-to-BIM	UEDIN	Binaries + source code	Closed GitHub repos at the moment. Will be open after D5.2 is published	• Deliverable D5.3 and D5.4 • Markdown files in the source repository	GNU GPLv3
BIM Management Platform (UCL BIM library)	UCL	Software as a service (SaaS)	Closed repository, and APIs to be exposed when core functionalities have been implemented.	Deliverable D5.1 and D5.2	Closed source, Cloud repository online accessible using APIs and a web interface (for openBIM data transfer)
ARIBFA	CERTH	HoloLens UWP App Binary	Closed source on private Gitlab repository	Deliverables D5.9 and D5.10	Closed source
BICA	SUITE5	Android Mobile App in Play Store	Closed source on private GitHub repository	Deliverables D5.5 and D5.6	Closed source
PRUBS	HYPERTECH	Software as a service (SaaS)	Closed source on private Gitlab repositories	Deliverables D5.7, D5.8	Closed source
RenoDSS: Building Material and Components Database	XYLEM	Software as a service (SaaS)	Closed source on private Gitlab repositories	Deliverables D7.1 and D7.2	Closed source
RenoDSS: Renovation LCA/LCC Module	XYLEM	Software as a service (SaaS)	Closed source on private Gitlab repositories	Deliverable D7.3 and D7.4	Closed source

Component	Responsible partner	Delivery (distribution) form	Source code repository	Documentation	Software License
RenoDSS: Building Energy Performance Modelling Module	HYPERTECH	Software as a service (SaaS)	Closed source on private Gitlab repositories	Deliverables D7.5 and D7.6	Closed source
RenoDSS: Urban Planning Module	XYLEM	Software as a service (SaaS)	Closed source on private GitLab repositories	Deliverable D7.7 and D7.8	Closed source
RenoDSS: Decision Support System Engine and UI & Module Integration	XYLEM	Software as a service (SaaS)	Closed source on private GitLab repositories	Deliverable D7.9 and D7.10	Closed source
Middleware: Storage (LinkSmart Historical Datastore)	FIT	Public Docker image	GitHub (public)[4]	• GitHub Wiki[5] • Deliverable D8.2	Apache 2.0
Middleware: Registry (LinkSmart Thing Directory)	FIT	Public Docker image	GitHub (public)[6]	• GitHub Wiki[7] • Deliverable D8.2	Apache 2.0
Middleware: Data Processor (Node-RED)	FIT	Public Docker image	GitHub (public)[8]	• Node-RED Docs[9] • Deliverable D8.2	Apache 2.0

[4] https://github.com/linksmart/historical-datastore

[5] https://github.com/linksmart/historical-datastore/wiki

[6] https://github.com/linksmart/thing-directory

[7] https://github.com/linksmart/thing-directory/wiki

[8] https://github.com/node-red/node-red

[9] https://nodered.org/docs/

Component	Responsible partner	Delivery (distribution) form	Source code repository	Documentation	Software License
Identity Provider (Keycloak)	FIT	Public Docker image	GitHub (public)[10]	• Keycloak Docs[11] • Deliverable D8.2	Apache 2.0
PWMA: Adaptive Renovation Process & Workflow Models	BOC	Software as a service (SaaS) + binary distribution	GitLab (public)[12]	Deliverables D6.4 and D6.5	Currently without license
PWMA: Renovation Process Simulation tool	BOC	Binary distribution	GitLab (public)[13]	Deliverables D6.4 and D6.5	Apache 2.0
PWMA: Adaptive Workflow Management and Automation tool	NT	Software as a Service (SaaS)	Closed source on private GitLab repositories	Deliverables D6.6 and D6.7	Closed Source
PWMA: Smart glass application for on-site renovation worker support	NT	Android Mobile App delivered by us	Closed source on private GitLab repositories	Deliverables D6.8 and D6.9	Closed Source
PWMA: Renovation progress monitoring & alerting application for residents	CERTH	Android Mobile App in Play Store	Closed source on private Gitlab repositories	Deliverables D6.10 and D6.11	Closed Source

[10] https://github.com/keycloak/keycloak

[11] https://www.keycloak.org/documentation

[12] https://git.boc-group.eu/bimerr/fast-deploy-package

[13] https://git.boc-group.eu/adoxx/knowledge-based-model-simulation

3.3 DEPLOYMENT REQUIREMENTS

In this subsection we introduce the general guidelines to be followed during the deployment of the BIMERR components and the integrated system in production environments. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this section are to be interpreted as described in RFC 2119 (Bradner 1997).

The requirements of the BIMERR component are specified as follows:

- The component MUST be protected from unauthorized access using state-of-practice authentication and authorization techniques.
- The component SHOULD NOT manipulate the host system without clear consent from the administrator.
- The component MUST NOT prevent the runtime of the host system by exhaustive use of resources.
- The component MUST be easily removable from the host system.
- The component MUST NOT include any functionality apart from what is advertised.
- The component MUST NOT collect and distribute data without end user consent.

Requirements of the Integrated BIMERR system are specified as follows:

- The system SHOULD be validated against different use cases described in Section 3.2 of D3.6.
- The integrated system SHOULD integrate compatible versions of the components.
- The system SHOULD be documented considering the technical knowledge of all stakeholders.
- The system MAY expose non-personal metrics that can be used for monitoring of the overall system.

4. INTEGRATION AND TESTING TASKS

This chapter discusses different use cases and how they are integrated as part of T8.3. First, we list out different use cases and the components involved in it. This is followed by listing out the status of verification of the interactions involved in the use cases.

4.1 USE CASES AND THE COMPONENTS

All the use cases and the components involved are listed in Table 4-1. The complete description of these use cases can be found in D3.1 and D3.6. The realization of the complete use cases is distributed across the BIMERR partners. As part of the integration testing activities, we shall validate the realization of these use cases. Since realization of few of the use cases are highly coupled with the realization of other use cases, the integration testing activities of such use cases are merged together.

Table 4-1 The use cases, components and the responsible partners for realization

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-01	Rapid scanning of the geometry of the building, semantic modeling and accurate representation in a BIM	• UEDIN* • SUITE5 • UCL	• Scan-to-BIM • BIF • BIM Management Platform	
UC-02	Accelerate the collection of data about the building systems through BIM-based internal audit support tools and interaction with building managers and occupants.	• CERTH* • SUITE5 • FIT • UCL	• ARIBFA, • Middleware • BIM Management Platform • BIF • BICA	

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-03	Adapt design to the actual building use, including accurate information about occupancy and schedules, comfort requirements/ preferences and energy uses.	• HYPERTECH* • SUITE5 • XYLEM • FIT	• PRUBS • RenoDSS • Middleware • BIF	
UC-04	Consider new materials and technologies in any design and simulation activity through appropriately configured BIM-compliant models residing in relevant open repositories (with accurate specification of their impact in energy performance of buildings).	XYLEM*	• RenoDSS • Material and component DB (Task 7.1)	
UC-05	Accurate scheduling of activities and assessment of their efficiency through simulation and verification.	• NT* • BOC • SUITE5	• PWMA • BIF	

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-06	Process automation and execution on a workflow-based approach (exchange of information and documentation on a BIM-based approach) with a sequential initiation of sub-processes once specific activities have been completed.	• NT* • BOC • SUITE5	• BIF • Renovation Process Simulation and Formal Verification • Adaptive Workflow Management and Automation tool	
UC-07	Stakeholders' systems exchange appropriate and "understandable" data between each other.	• SUITE5* • UPM • UBITECH • FIT	• Middleware • BIF	
UC-08	Daily renovation activity schedules are automatically generated (based on accurate project scheduling) and individual guidelines are provided to the workforce responsible through ambient interfaces and apps.	• NT* • BOC • SUITE5 • CERTH	• Adaptive Workflow Management and Automation tool • PWMA for workers app • BIF • ARIBFA	The integration of UC-08 and UC-09 is performed together as almost similar interactions are followed to achieve them.

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-09	Continuous monitoring and updates of renovation activity schedules (based on reporting from the workforce and monitoring of process execution) towards effective devising and avoidance of delays (bi-directional communication through ambient interfaces)	• NT* • BOC	• Adaptive Workflow Management and Automation tool • PWMA for workers app	
UC-10	Continuous reporting from workforce and occupants for changes performed over the initial renovation design (location-based on a BIM representation) and automated update of the BIM model (as-built documentation).	• NT* • SUITE5 • CERTH • BOC • UCL	• Adaptive Workflow Management and Automation tool • PWMA for residents app • PWMA for workers app • BIF • BIM Management platform	
UC-11	Identification of threats and dangers and provision of alerts to workforce and occupants through BIM-based apps and UIs	• CERTH* • SUITE5 • NT	• Adaptive Workflow Management and Automation tool • PWMA for residents app • PWMA for workers app	The integration of UC-11 and UC-12 is performed together as the components are same and follow a similar sequence of interactions.

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-12	Continuous reporting from workforce and occupants for dangers and threats (location-based on a BIM representation) and automated update of the BIM model.		• BIF	
UC-13	Perform back-to-back simulations of alternative renovation scenarios to evaluate and select the best energy-performing renovation scenario	• XYLEM* • HYPERTECH • SUITE5	• RenoDSS • BIF	The integration of UC-13 to UC-16 is performed together as the components are same and follow a similar sequence of interactions.
UC-14	Energy performance assessment to be elevated at a life-cycle perspective including relevant LCA-LCC metrics			
UC-15	Energy performance simulations to assess not only energy metrics, but also accurately evaluate occupants' comfort and indoor air quality			

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components	Additional Notes
UC-16	Assessment of energy performance to also address the district aspect and enable the consideration of interactions between buildings, but also between buildings and district systems in a holistic assessment framework incorporated in urban planning applications			

4.2 USE CASE INTERACTION VERIFICATION STATUS

We use the BIMERR use case sequence diagrams described in D3.6 as the base for the integration verification. The updated sequence diagrams of the use cases after the submission of D3.6 (after M20) are provided in the appendix of this deliverable.

Table 4-2 shows the status of different interactions derived from the sequence diagrams of different use cases. Although some of the interactions have been implemented already, their validation activities are not documented. Interactions whose validations are not currently documented are also considered incomplete in the current deliverable. This table shall be updated in the upcoming deliverables with the changes in the corresponding status.

Table 4-2 Interactions and the status of their verifications

Use Case/s	Summary	Description	Components	Means of Verification	Status
UC-01	Building surveyor verifies BIM delivered by scan-to-BIM		Scan-to-BIM	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
UC-01	BIM Management Platform checks IFC geometry and sends file back to building surveyor		BIM Management Platform	Automated-Testing	In Progress
UC-01	Building surveyor adds PSets information through Scan-to-BIM		Scan-to-BIM	Manual-Testing	In Progress
UC-01	BIM Management Platform receives IFC from the BIF		BIF, BIM Management Platform	Manual-Testing	In Progress
UC-01	Building surveyor creates a new project and uploads the BIM model (i.e., IFC file) to the BIF		BIF	Manual-Testing	In Progress
UC-02	User (building occupant) logs in to her BICA app account.	User (building occupant) logs in to her BICA app account.	BICA	Manual-Testing	In Progress
UC-02	BICA receives from BIF the related parts of the apartment's IFC, components, comfort data.	BICA receives from BIF the related parts of the apartment's IFC, components, comfort data.	BICA, BIF	Manual-Testing	In Progress
UC-02	BIM-MP Receives BIM IFC from ARIBFA	Steps: * ARIBFA exports updated BIM IFC file. * Exported file is available to BIF's data collection jobs through a REST service's endpoints implemented inside ARIBFA. * BIM-MP receives IFC from BIF.	ARIBFA, BIF, BIM Management Platform	Manual-Testing	In Progress
UC-02	BIM-MP Receives BIM MVD from ARIBFA	Steps: * ARIBFA exports BIM MVD file. * Exported file is available to BIF's data collection jobs through a REST service's endpoints implemented inside ARIBFA. * BIM-MP receives BIM MVD file from BIF.	ARIBFA, BIM Management Platform	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
UC-02	ARIBFA Receives Error report from BIM-MP	Steps: * Other BIMERR tools upload Error Reports on BIM-MP. * ARIBFA connects to BIM-MP. * Error Reports collected are imported and incorporated to ARIBFA's procedures.	ARIBFA, BIM Management Platform	Manual-Testing	In Progress
UC-02	ARIBFA Receives BIM Model from BIF	Î'RIBFA connects to BIF's provided server endpoints and downloads the BIM Model. At this stage of development, there are no direct API connection capabilities for BIF file downloading. So testing is being done manually.	ARIBFA, BIF	Manual-Testing	In Progress
UC-02	BICA receives from Middleware the sensor measurements from the apartment.	BICA receives from Middleware the sensor measurements from the apartment.	BICA, Middleware	Manual-Testing	In Progress
UC-02	BIF gets the user's data from BICA.	BIF gets the user's data from BICA.	BICA, BIF	Manual-Testing	In Progress
UC-02	The other BIMERR tools get the submitted information in BICA from BIF	The other BIMERR tools get the submitted information in BICA from BIF	BICA, BIF	Manual-Testing	In Progress
UC-02	BIF Receives BIM IFC from ARIBFA	Steps: * BIF's data collection jobs perform requests to ARIBFA's available REST service endpoints. * ARIBFA's REST service consumes the requests and responds with the required file.	ARIBFA, BIF	Manual-Testing	In Progress
UC-02	BICA displays the apartment information to the user through the UI.	BICA displays the apartment information to the user through the UI.	BICA	Manual-Testing	In Progress
UC-02	BICA receives notifications from other BIMERR components for additional input from the user. BICA	BICA receives notifications from other BIMERR components for additional input from the user. BICA displays the notification to the user.	BICA	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
	displays the notification to the user.				
UC-02	User submits additional information through the BICA UI	User submits additional information regarding the apartment, the components, issues and her comfort through the BICA UI.	BICA	Manual-Testing	In Progress
UC-03	BIF-RenoDSS interaction: RenoDSS receives obXML from BIF	* RenoDSS data management module requests obXML for specific project from BIF * Manual verification that correct obXML file was received from BIF	BIF, RenoDSS	Manual-Testing	In Progress
UC-03	PRUBS - BIF interaction: BIF receives obXML from PRUBS		BIF, PRUBS	Demonstration	In Progress
UC-03	BICA - BIF interaction: BICA receives and sends OB data to BIF		BICA, BIF	Manual-Testing	In Progress
UC-03	PRUBS - Middleware interaction: PRUBS receives WoT and SenML data from Middleware		Middleware, PRUBS	Automated-Testing	In Progress
UC-04	RenoDSS UI retrieves materials and their properties from the database	* User opens a RenoDSS view with material properties (material admin interface or "check and complete missing data" view) * Manual verification that the material properties shown in the UI are identical to the ones stored in the database	Material and Component Databases, RenoDSS	Manual-Testing	In Progress
UC-04	Database receives new/updated material properties from RenoDSS UI	* User opens material in RenoDSS UI * User updates all material properties fields * User presses Save Button to trigger update in Database form RenoDSS UI * Check if DB content for the given material is identical to the data provided by the UI to the DB	Material and Component Databases, RenoDSS	Manual-Testing	In Progress
UC-04	RenoDSS extends IFC file with correct material property	* Original IFC file is loaded and materials are extracted from the IFC file	Material and Component	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
	sets as retrieved from database	* IFC materials are mapped by the user to materials in the database OR user provides material properties manually to the RenoDSS UI * IFC file is extended with correct material properties and written to disk by the RenoDSS data management module * Updated IFC file is manually inspected to verify that correct material properties are included in the IFC file OR updated IFC file is loaded by RenoDSS UI to verify the correctness of stored material properties	Databases, RenoDSS		
UC-05	Project manager selects optimal renovation process and defines KPIs using Workflow modelling and simulation tool		PWMA	Manual-Testing	In Progress
UC-05	Project manager runs simulation		PWMA	Manual-Testing	In Progress
UC-05	Project manager redesigns reference model, sets required parameters and formally verifies model using Workflow modelling and simulation tool		PWMA	Automated-Testing	In Progress
UC-05	Project manager selects renovation project in Workflow modelling and simulation tool		PWMA	Manual-Testing	In Progress
UC-06	PM updates Schedule in the PWMA Execution engine		PWMA	Manual-Testing	In Progress
UC-06	Notification system collects information from the PWMA Execution engine and issues corresponding notifications		PWMA	Manual-Testing	In Progress
UC-06	PWMA Workflow and simulation modelling		PWMA	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
	tool sends updated schedule to the PWMA Execution engine				
UC-06	PWMA Workflow and simulation modelling tool gets schedule from the PWMA Execution engine		PWMA	Manual-Testing	In Progress
UC-06	PM creates Initial Schedule in the PWMA Execution engine	PM creates Initial Schedule in the PWMA Execution engine from scratch or from imported BPMN file created in UC-05.	PWMA	Manual-Testing	In Progress
UC-06	PWMA Execution engine populates initial schedule into BIF		BIF, PWMA	Manual-Testing	In Progress
UC-07	App/middleware sends raw data to BIF [under a configured data collection process] through an API according to a predefined schedule.	The app/middleware sends raw data to BIF [under a configured data collection process] through an API according to a predefined schedule. The app/middleware developer can find the data under her account with the data structure, metadata and policies she has defined.	BIF, Middleware	Manual-Testing	In Progress
UC-07	User sees a list of available data assets in the BIF UI, based on the applied access policies.	The user sees a list of available data assets in the BIF UI, based on the applied access policies. The user applies filters, searches using keywords and concepts and can see the preview for each data available on the list.	BIF	Manual-Testing	In Progress
UC-07	The uploaded data are available for search from BIF users	The data are available for search from BIF users, through the BIF UI, according to the access policies that apply.	BIF	Manual-Testing	In Progress
UC-07	The BIMERR applications (or any application) send Other files (e.g. IFC) to BIF [under a configured data collection process] through an API on demand or according to a predefined schedule	The BIMERR applications (or any application) send Other files (e.g. IFC) to BIF [under a configured data collection process] through an API on demand or according to a predefined schedule. The app developer can find the data under her account with the data structure, metadata and policies she has defined.	BIF	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
UC-07	Add metadata and access policies to data	The user adds through the UI the metadata and access policies that will apply the data that will be retrieved. BIF updates the configuration file accordingly. BIF stores the final configuration.	BIF	Manual-Testing	In Progress
UC-07	Mapping verification and finalisation	The user verifies the mapping configuration. BIF creates the final mapping configuration based on the concept mapping and units/transformations configuration. BIF stores the final configuration.	BIF	Manual-Testing	In Progress
UC-07	Changes to mapping configuration (before it is finalised)	(optional) The user goes back to the mapping configuration and makes any changes.	BIF	Manual-Testing	In Progress
UC-07	User create new concept request	The user creates a concept request through the BIF UI. BIF receives the request and forwards it to the BIMERR Data Model Admin through the Model Lifecycle Manager.	BIF	Manual-Testing	In Progress
UC-07	BIF apply data model changes to previously stored files and communicate changes to Ontology Manager Framework	BIF updates the previously stored files to the new version of the model (backwards compatibility) and communicates the changes made to the Ontology Manager Framework.	BIF	Manual-Testing	In Progress
UC-07	Make changes to the BIMERR data model through the Model Lifecycle Manager	The BIMERR Admin/Modeler makes changes to the data model in the Model Lifecycle Manager.	BIF	Manual-Testing	In Progress
UC-07	The user receives a notification in BIF about the status/progress of her concept request.	The user receives a notification in BIF about the status/progress of her concept request.	BIF	Manual-Testing	In Progress
UC-07	Mapping corrections through the BIF UI	User makes mapping corrections through the BIF UI, sets the units for numeric/datetime fields, runs the validations check. BIF takes the concept mapping and units/transformations configuration, creates the mapping and displays it to the user	BIF	Manual-Testing	In Progress
UC-07	User previews the results of the query in the integrated console in the BIF UI	The user previews the results of the query in the integrated console in the BIF UI.	BIF	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
UC-07	User creates query from the available data assets. BIF provides the user with an API endpoint to retrieve data	The user selects one or more of the available data assets from the list, selects the concepts she wants to retrieve and saves the query. BIF provides an API endpoint to the user to retrieve the data.	BIF	Manual-Testing	In Progress
UC-07	User upload data as file to BIF	User uploads data as file to BIF [under a configured data collection process]. BIF receives the data, applies the necessary pre-processing steps (according to the configuration) and stores the file. The user can find the data under her account with the data structure, metadata and policies she has defined.	BIF	Manual-Testing	In Progress
UC-07	User/app uses the API and the authorisation token she has been provided from BIF to retrieve the data	The user/app uses the API and the authorisation token she has been provided from BIF to retrieve the data. The user/app gets a response with the data.	BIF	Manual-Testing	In Progress
UC-07	BIF creates a semi-automated mapping of the sample data to the BIMERR model and displays it to the user through the BIF UI	BIF creates a semi-automated mapping of the sample data to the BIMERR model and displays it to the user through the BIF UI	BIF	Manual-Testing	In Progress
UC-07	BIF receives from the user the sample data file.	BIF receives from the user the sample data file.	BIF	Manual-Testing	In Progress
UC-07	User creates a new data collection process in BIF	User (app developer/middleware developer/stakeholder) creates a new data collection process in BIF. User can find the unconfigured data collection under her account in BIF.	BIF	Manual-Testing	In Progress
UC-08, UC-09	As worker moves through the construction site, his/her location is continuously identified	After the 3D BIM model is effectively loaded, visualized, and aligned to the real world (registration), the user's position and orientation within the building should be continuously identified. To achieve indoor localization in ARIBFA, the aligned 3D BIM model is set as a persistent spatial anchor. Therefore, its position and orientation is maintained and tracked	ARIBFA	Demonstration	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
		across different initializations of the application via the Hololens SDK.			
UC-08, UC-09	Site manager selects particular reconstruction task and develops corresponding work orders and work instructions in the PWMA Execution engine UI		PWMA	Manual-Testing	In Progress
UC-08, UC-09	Site manager assigns Work orders to particular Worker		PWMA	Manual-Testing	In Progress
UC-08, UC-09	Worker reports results of Work order to Site manager		PWMA	Manual-Testing	In Progress
UC-08, UC-09	Actual worker location and pose are send to PWMA On-site support app. Worker can see appropriate work orders according to his/her actual location.	Work orders are sent from the PWMA On-site support app to ARIBFA. ARIBFA visualizes the appropriate work tasks depending on worker's location and pose.	PWMA	Demonstration	In Progress
UC-08, UC-09	Worker asks Site manager for additional information		PWMA	Manual-Testing	In Progress
UC-08, UC-09	Worker displays mapped BIM or information about components	The worker visualizes the mapped BIM model and receives information regarding the components. In ARIBFA, the worker can select a 3D building component of interest and visualize its IFC properties via a user interface.	ARIBFA, PWMA	Manual-Testing	In Progress
UC-10	Site manager verifies the change and decides to adjust respective workorder or send a notification to Foreman.		PWMA	Manual-Testing	In Progress
UC-10	Site manager verifies the change and decides it should be ignored.		PWMA	Manual-Testing	In Progress
UC-10	Foreman/worker sends change		PWMA	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
	notification to the Site manager.				
UC-10	Resident sends change notification to the PWMA back end.		PWMA	Manual-Testing	In Progress
UC-11	Foreman/worker sends H&S issue notification to the Site manager.		PWMA	Manual-Testing	In Progress
UC-11, UC-12	PWMA for Residents receives H&S Notification from PWMA	PWMA for Residents receives H&S Notification from PWMA backend through an API in a JSON format. * User opens PWMA for Residents * Navigates to Notifications Screen by tapping the Notification button * App downloads JSON from PWMA Notification Server * Selects H&S Notification and displays it in the Notification Screen	PWMA	Manual-Testing	Done
UC-12	PWMA for Residents receives BIM Model from BIF	PWMA for Residents connects to BIF's provided server endpoints and downloads the BIM Model.	BIF	Manual-Testing	In Progress
UC-12	PWMA for Residents sends BCF with H&S annotations to BIF	PWMA for Residents sends BCF with H&S annotations to BIF as JSON according to BCF format.	BIF, PWMA	Manual-Testing	In Progress
UC-12	PWMA for Residents sends new H&S Issue to PWMA		PWMA	Manual-Testing	In Progress
UC-12	Site manager/H&S manager sends H&S report, notifications and instructions to foreman/worker.		PWMA	Manual-Testing	In Progress
UC-12	PWMA for Residents sends new H&S Issue to PWMA	PWMA for Residents sends new H&S Issue to BIF via a REST API service, which is accessible to the data collection jobs in BIF. Then PWMA backend gets the new H&S Issue from BIF.	PWMA	Manual-Testing	In Progress
UC-13, UC-14, UC-15, UC-16	Data Management retrieves IFC, obxml, and epw files from BIF	* User logs into RenoDSS * RenoDSS data management module requests projects to which	BIF, RenoDSS	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
		the user is authorized from BIF * User selects project * RenoDSS data management module requests IFC, obxml, and epw files for this specific project from BIF * Manual verification if correct files were retrieved from BIF			
UC-13, UC-14, UC-15, UC-16	RenoDSS UI receives Projects from Data Management	* User logs into RenoDSS * RenoDSS data management module requests projects to which the user is authorized from identity provider * RenoDSS data management module forwards the correct projects to RenoDSS UI and displays them to the user	RenoDSS	Manual-Testing	In Progress
UC-13, UC-14, UC-15, UC-16	BIF retrieves selected renovation scenarios as IFC files and KPIs as JSON file	* User logs into RenoDSS, selects projects, sets renovation measures, generates renovation scenarios and selects specific renovation scenario for implementation * RenoDSS data management module pushes IFC files and KPIs (JSON) of the selected renovation scenario to BIF * Manual verification that the correct IFC files and KPIs have been pushed to BIF	BIF, RenoDSS	Manual-Testing	In Progress
UC-13, UC-14, UC-15, UC-16	RenoDSS UI retrieves Energy, LCA/LCC, and Urban Planning KPIs for each renovation scenario from Data Management	* User logs into RenoDSS, selects project and sets potential renovation measures * RenoDSS scenario generator generates renovation scenarios and RenoDSS modules calculate KPIs * Manual verification that the scenario KPIs have been calculated correctly	RenoDSS	Manual-Testing	In Progress

Use Case/s	Summary	Description	Components	Means of Verification	Status
		* RenoDSS UI requests from RenoDSS data management module the scenario KPIs and displays them in the UI * Verification that the correct scenario KPIs are displayed in the RenoDSS UI			
UC-13, UC-14, UC-15, UC-16	Renovation scenarios are generated by Data Management based on set renovation measures	* User logs into RenoDSS, selects project and sets potential renovation measures * RenoDSS scenario generator generates renovation scenarios * Manual verification that renovation scenarios contain only renovation measures according as set by the user in the "renovation measure" view	RenoDSS	Manual-Testing	In Progress
UC-13, UC-14, UC-15, UC-16	RenoDSS UI retrieves baseline scenario Energy, LCA/LCC, and Urban Planning KPIs from Data Management	* User logs into RenoDSS and selects project * RenoDSS modules calculate the baseline KPIs * Manual verification that the baseline KPIs have been calculated correctly * RenoDSS UI requests from RenoDSS data management module the baseline KPIs and displays them in the UI * Verification that the correct KPIs are displayed in the RenoDSS UI	RenoDSS	Manual-Testing	In Progress

5. SECURE INFORMATION EXCHANGE

The BIMERR system consists of various software components deployed in different locations and on a variety of platforms. These range from server applications deployed and exposed in the cloud to software running on isolated gateways on-premises and applications on end-user desktop, mobile or head-mounted computers. Regardless of the type and the deployment of the applications, they always have to communicate over a secure, TCP/IP network. The project had to ensure authenticated and authorized access to protect the integrity of the system as well as to protect sensitive occupancy and construction data.

5.1 AUTHENTICATION

The information exchange in the BIMERR system is either between users and applications or among the applications (machine to machine). We use the term "user" to refer to human operators and term "client" to refer to applications. When a user or client needs to access resources available in another application, they need to be authenticated. The authentication process is to verify if a user or client is who they claim to be.

We selected the OpenID Connect protocol for authentication and secure exchange of profiles among users and applications. This protocol uses OAuth 2.0[14] internally to authenticate users and clients based on specific flows or grant types. The flows perform authentication based on provided credentials such as username, password, secret, and certificate. Two-factor authentication (2FA) can be added for increased security. The flows typically involve exchanging these credentials with an Identity Provider (Authentication Server) and getting back a verifiable security token. Users and application should follow OAuth 2.0 flows which are designed for them. The common flows are listed below:

- Authorization Code: Suitable for traditional web apps, single-page apps, mobile apps, modern desktop apps. This flow redirects the user from applications or websites to a web page to provide credentials and give necessary consent before being redirected back. This enables single sign-on (SSO) capabilities such that the authentication process does not have to be repeated when a user tries to access other applications in a specified period.

[14] https://oauth.net/2/

- Resource Owner Password Credentials: For trusted applications; user credentials are shared with the application and must be handled with care.
- Client Credentials: For machine-to-machine authorization when there is no user involvement.

The full list of flows and the specifications are available in OpenID Connect Core 1.0[15] and the OAuth 2.0 Authorization Framework[16] specifications.

Once the authentication flow is complete, the application will receive a token which contains the profile of user or client (given as a JWT Token). The token is signed by the issuer (Identity Provider) and can be cryptographically verified.

5.2 AUTHORIZATION

When an application receives a request for resources, it has to check whether the requester is allowed to perform the operation on the underlying resource. The Identity Provider performs the authentication but that is often not sufficient for fine-grained access control. We predefined a set of attributes that are useful to identify a user and make authorization decisions. The Table 5-1. BIMERR to Identity Provider attribute mapping lists these attributes and provides a mapping to technical terms available in the OpenID Connect (used for authentication) domain:

Table 5-1. BIMERR to Identity Provider attribute mapping

BIMERR Attribute	Identity Provider Attribute	Example
Data provider/consumer	User	Jane Doe
Renovation project	Group	KRIPIS
User role	Role	Renovation Planner
Application	Client	BIF

[15] https://openid.net/specs/openid-connect-core-1_0.html

[16] https://tools.ietf.org/html/rfc6749

Applications that host and serve sensitive information such as user behaviors, construction and renovation data, and project execution secrets are responsible for enforcing access control on every request.

At the time of writing, authorization is performed at two data providing points: BIMMER Interoperability Framework (BIF) and Middleware. The BIF utilizes Building Information Secure Provisioning (BISP) for access control on requests submitted to it. It does so by cross checking the security token data with its existing internal policy database populated using profile data queried from Identity Provider and configured with a UI. The Middleware Registry and Storage perform access control against security tokens based on simple policies maintained internally in configuration files.

5.3 BIMERR Identity Groups and Roles

A set of groups and roles have been defined to address various access control needs beyond what is already possible with username and client IDs.

The list of current groups and roles are available in Table 5-2 and Table 5-3 respectively.

Table 5-2. List of BIMERR identity groups

Group Name	Description
FITDEV	Testing site at Fraunhofer FIT
KRIPIS	KRIPIS pre-validation site by CERTH
CONKAT	Pre-validation site by CONKAT
Budimex Pilot	Budimex pilot site
Ferrovial Pilot	Ferrovial pilot site

Table 5-3. List of BIMERR identity roles

Role Name	Description
ARIBFA Developer	Application developer
BICA Developer	Application developer
BIMERR Developer	Software developer in BIMERR Project. Members can view all users, groups, roles.

BIMERR Identity Manager	Users who can create group and assign group/roles.
BIMMP Developer	BIM Management Platform Developer
Building Manager	BIMERR Building Manager
Construction Manager	BIMERR Construction Manager
Demo Role	This role is for documentations only
Device Maintainer	People who are responsible for maintenance of WSN
Middleware Developer	Application developer
Project Manager	Renovation/Construction project manager
PRUBS Developer	Application developer
PWMA Developer	Application developer with additional access to registration events

6. UNIFIED UI DESIGNS AND UX ALIGNMENTS

To ensure that the various software components provide a coherent experience for end-users, a simple style guide was created. This style guide was based on both the material.io [17] guidelines, as well as the color schema and font types defined for the communication materials. The simple style guide was documented in the BIMERR project's wiki and consists of rules for the usage of colors, font sizes, font families, the BIMERR project logo's usage, and the application of icons.

While some of the developed end-user applications are based on frameworks that are difficult to adapt to the style guides, other applications for head-mounted displays (HMDs) and smart glasses require a different approach to UI design compared to web and mobile-based graphical interfaces. Therefore, the style guide only defines a minimum set of rules to be as flexible as possible and therefore adaptable for cross-device development. Table 6-1 provides an overview over the developed applications and the target medium as well as the user group and the responsible task / partner.

Table 6-1 Overview over applications, targeted platforms, responsible partners and affected tasks

UI Name	Partner	Platform/Device	Target Group	Resource	Task
Scan-To-BIM	UEDIN	Desktop	surveyor / BIM modeler	Open Infra Platform[18]	T5.2
ARIBFA	CERTH	HoloLens (HMD)	construction workers	-	T5.5, T5.6
BICA	Suite5	Smartphone (iOS, Android)	building residents	-	T5.3
PWMA	CERTH	Smartphone (Android)	building residents/owners	-	T6.5
PWMA Process Design	BOC	Web browser	Renovation Process Modeler	-	T6.3

[17] https://material.io/

[18] https://www.cms.bgu.tum.de/en/17-research-projects/46-open-infra-platform

PWMA Process KPI Dashboard	BOC	Web browser	Renovation Process Modeler	-	T6.3
PWMA Process KB Simulation and Verification	BOC	Web browser	Renovation Process Modeler	-	T6.3
I3D	NT	Web browser	Reconstruction Project manager Technologist	-	T6.4
I3D	NT	HMT-1 (HMD)	construction workers	-	T6.5
RenoDSS	Xylem	Desktop	renovation designer	-	T7.5

The following subsections provide an overview of the style guide and details about the rules to be applied in order to create an aligned user experience.

6.1 USAGE OF THE LOGO

The usage of the colorful logo is to be preferred. The logo should be used on transparent background.

Applications on all platforms should show the logo on the login-screen and / or the loading-screen. Desktop and web applications should additionally add the logo to the header section (if available) and as a favicon (if possible). Additionally, the logo should be shown on the privacy, terms and conditions, and contact pages (if available). The logo versions are presented in Table 6-2. Adaptions of the logo are not allowed.

Table 6-2 Various versions of the logo

Logo Version	Usage
Color version	To be applied on mono color backgrounds with decent contrast
	To be applied on bright colorful backgrounds

Black	
 White	To be applied on dark colorful backgrounds

6.2 COLORS AND COLOR SCHEMA

The color palette for applications is based on the colors used in the BIMERR project logo and further detailed in Table 6-3. In case a framework used for designing an application for the BIMERR project does not support the definition of personalized colors, the creators should apply a color schema as close as possible to the colors defined in Table 6-3.

Table 6-3 Defined colors and their usage

	Primary	**P Light**	**P Dark**
Hex	#299cd5	#6ccdff	#006ea3
RGB	41, 156, 213	108, 205, 255	0, 110, 163
Text color	black	black	black & white
	Secondary	**S Light**	**S dark**
Hex	#55bd81	#88f0b0	#1a8c53
RGB	85, 189, 129	136, 240, 176	26, 140, 83
Text color	black	black	black & white

	Attention	Error	
Hex	#ffb300	#f4511e	
RGB	255, 179, 0	244, 81, 30	
Text color	black	black and white	
	Background1	**Background2**	**Background3**
Hex	#f5f5f5	#e1e1e1	#333333
RGB	245, 245, 245	225, 225, 225	51, 51, 51
Text color	black	black	white

The application of the colors **Attention** and **Error** is reserved for the respective message types. Besides from that, background colors should be used for page and paragraph backgrounds, primary and secondary colors for main screen elements such as headers etc. If an area needs to be highlighted, it is also legible to use the respective primary and secondary color (basic, light, dark) as a background. Figure 6-1 provides examples for a good application of colors.

The screens depicted in Figure 6-1 in fact show examples for mobile websites or native apps. However, the color combination also holds for webpages and desktop apps, and HMDs.

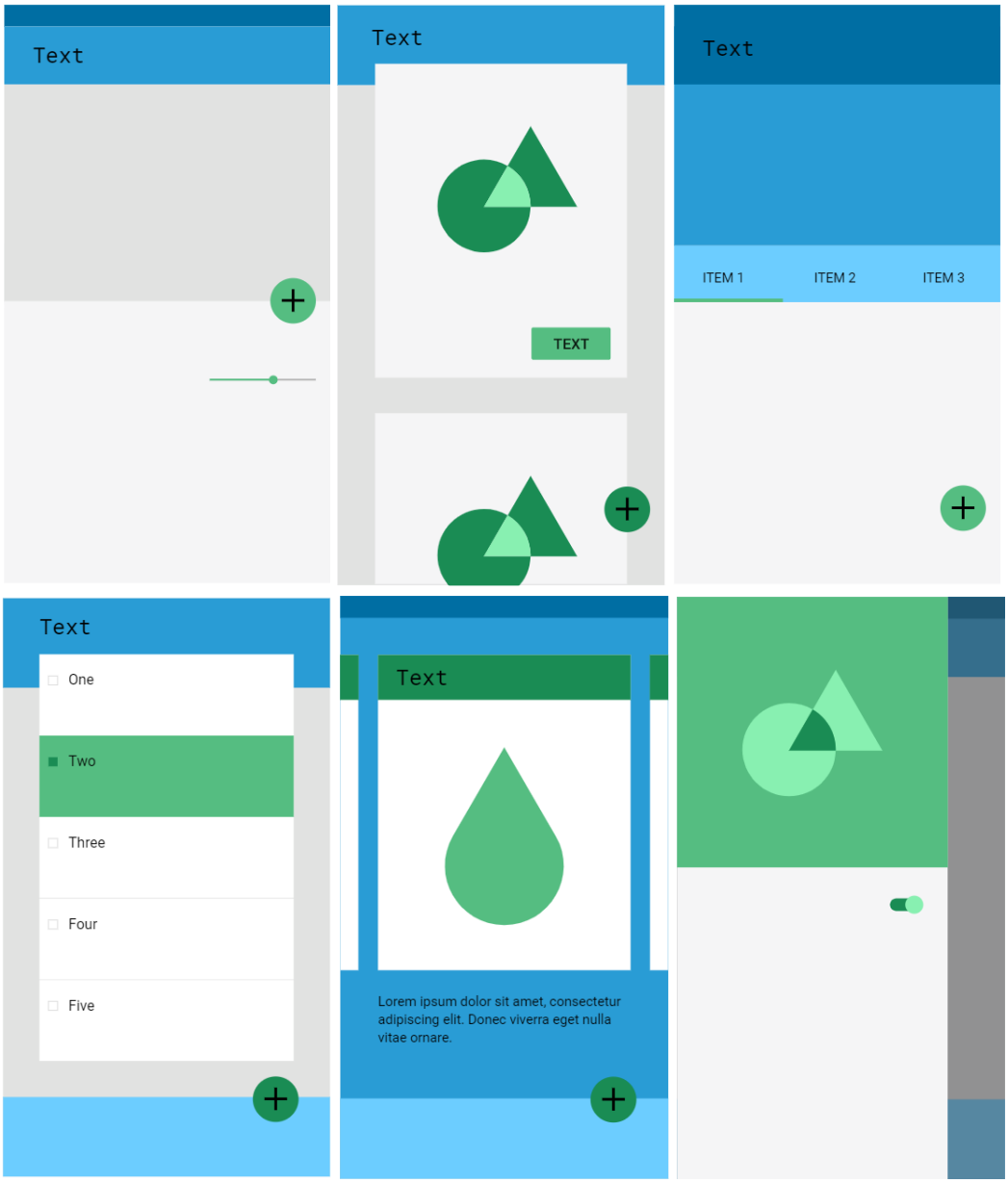

Figure 6-1 Example for the application of the color schema

6.3 ICONS

Icons, if needed, should be taken from the material.io library (svg or png format). Applications should use the two-tone library [19] in either black or white, depending on the applied background colors. Table 6-4 provides some examples.

Table 6-4 Examples for two-tone library in black and white

Bug Report	Build	Assignment Late	Download	Accessibility
				
				

6.4 TYPOGRAPHY

The typography selected for BIMERR applications is based on Google fonts[20] to ensure the availability for web and mobile applications, as well as to offer the option to download and use the proposed typography as a local font.

Medium 500

BIMERR RENOVATION 4.0

Figure 6-2 Type face for titles, headlines, and subtitles - Josefin Sans Medium 500

[19] https://material.io/resources/icons/?style=twotone

[20] https://fonts.google.com/

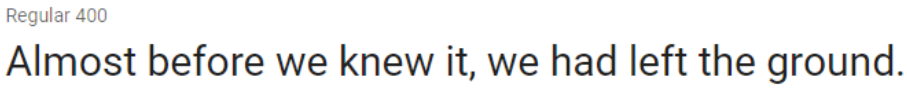

Figure 6-3 Type face for paragraphs – Roboto Regular 400

For headlines, titles, and subtitles the font Josefin Sans Medium 500[21] (Figure 6-2) was selected, whereas paragraphs should be styled in Roboto Regular 400[22] (Figure 6-3). Table 6-5 lists the available scale categories and how to apply them. In general, the adaption of font styles is not allowed to ensure a unified user experience across the BIMERR UIs.

Table 6-5 Available scale categories

Scale Category	Typeface	Weight	Size	Letter spacing	Case
Headlin	Josefin Sans	Light	123px	-1.5px	Sentence
Headline 2	Josefin Sans	Light	77px	-0.5px	Sentence
Headline 3	Josefin Sans	Normal	61px	0px	Sentence
Headline 4	Josefin Sans	Normal	43px	0.25px	Sentence
Headline 5	Josefin Sans	Normal	31px	0px	Sentence

[21] https://fonts.google.com/specimen/Josefin+Sans

[22] https://fonts.google.com/specimen/Roboto

Headline 6	Josefin Sans	Medium	26px	0.15px	Sentence
Subtitle 1	Josefin Sans	Normal	20	0.15px	Sentence
Subtitle 2	Josefin Sans	Medium	18px	0.1px	Sentence
Body 1	Roboto	Normal	16px	0.5px	Sentence
Body 2	Roboto	Normal	14px	0.25px	Sentence
BUTTON	Roboto	Medium	14px	1.25px	All caps
Caption	Roboto	Normal	12px	0.4px	Sentence
OVERLINE	Roboto	Normal	10px	1.5px	All caps

CONCLUSIONS

This document describes the work carried out within T8.3 during the first 12 months after the kickoff of the task (M15 to M26). The achievements so far include the following:

- Identified the version control, delivery and distribution of individual BIMERR components.
- Identified the methodology to be followed for the verification activities.
- Established a tracking methodology and tooling for the integration testing activities.
- Designed and partially executed the integration testing activities to ensure information flow across different components involved in BIMERR use cases.
- Identified authentication and authorization mechanisms.
- Recommendations for the UI designs and user experience alignments are concluded as part of the task and described in the current deliverable.

Initial integration testing activities shall be continued until M30 and the results shall be submitted in the second version of this deliverable. Further refinement of the BIMERR system shall be carried out until M40 as part of task T8.3.

BIBLIOGRAPHY

[Baresi and Pezze, 2006] Baresi, L. and Pezze, M., 2006. An introduction to software testing. *Electronic Notes in Theoretical Computer Science*, *148*(1), pp.89-111.

[Bradner 1997] Bradner, S., 1997. RFC2119: Key words for use in RFCs to Indicate Requirement Levels.

[Leung and White 1990] Leung, H. K., & White, L. (1990, November). A study of integration testing and software regression at the integration level. In Proceedings. Conference on Software Maintenance 1990 (pp. 290-301). IEEE.

A . APPENDIX - UPDATED USE CASE SEQUENCE DIAGRAMS

This appendix provides the sequence diagrams for the use cases updated after M20 of the BIMERR project. The original sequence diagrams and their descriptions have already been presented in the Section 3.2 of D3.6. Therefore, D3.6 acts as the baseline for the current section where only updated sequences are provided.

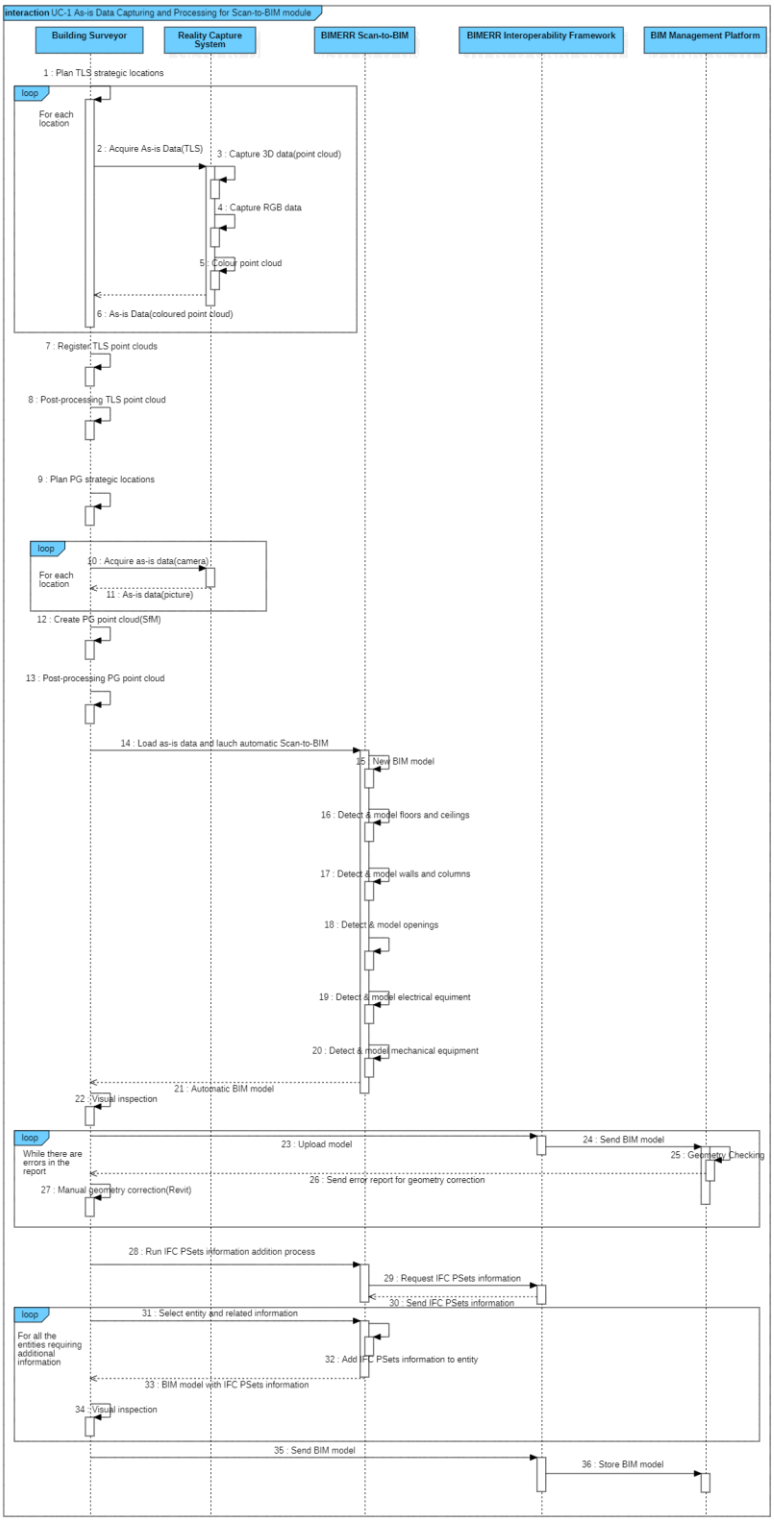

Figure A-0-1 UC–01: As-is Data Capturing and Processing for Scan-to-BIM

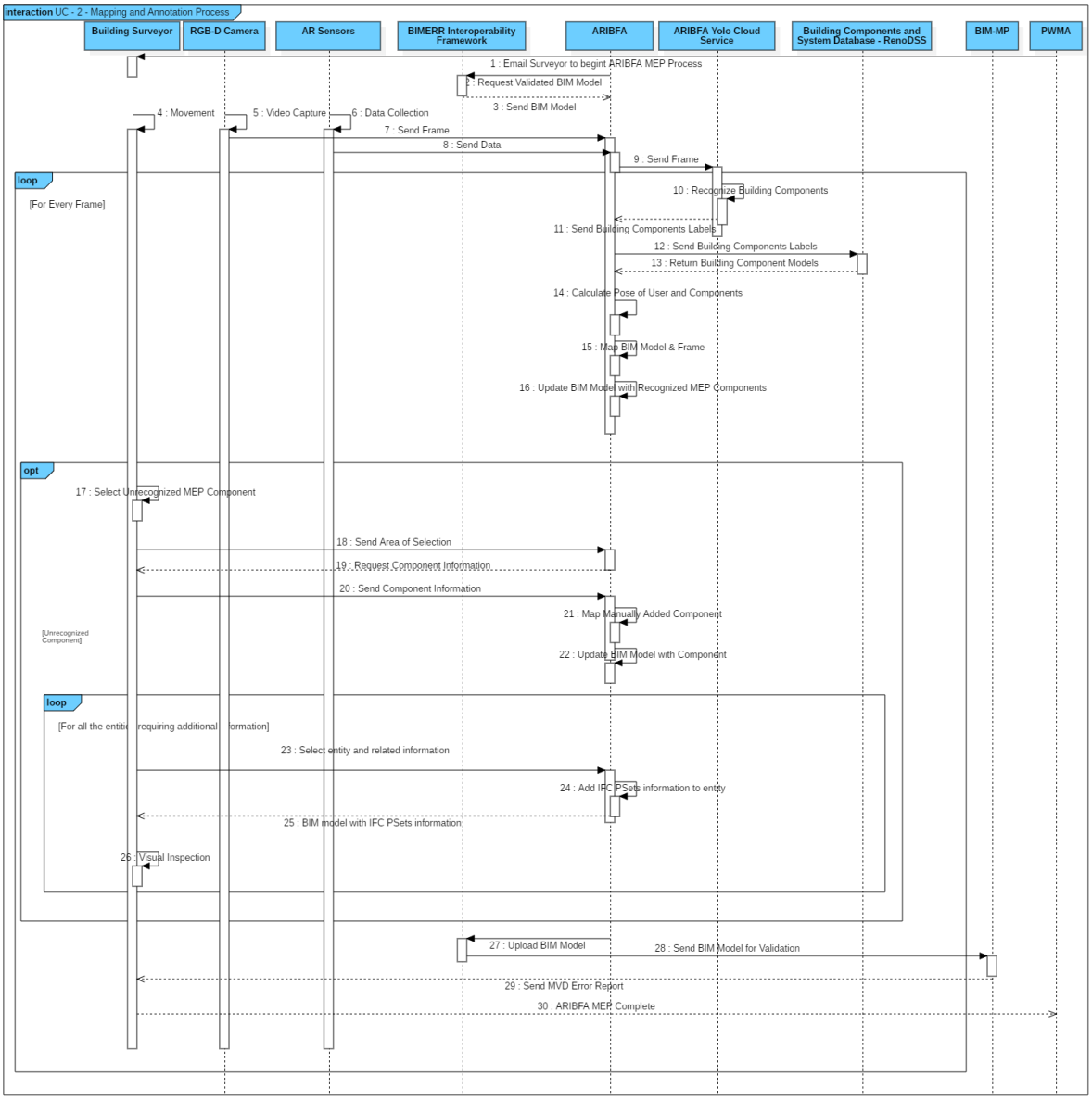

Figure A-0-2 UC-02: Mapping and annotation process

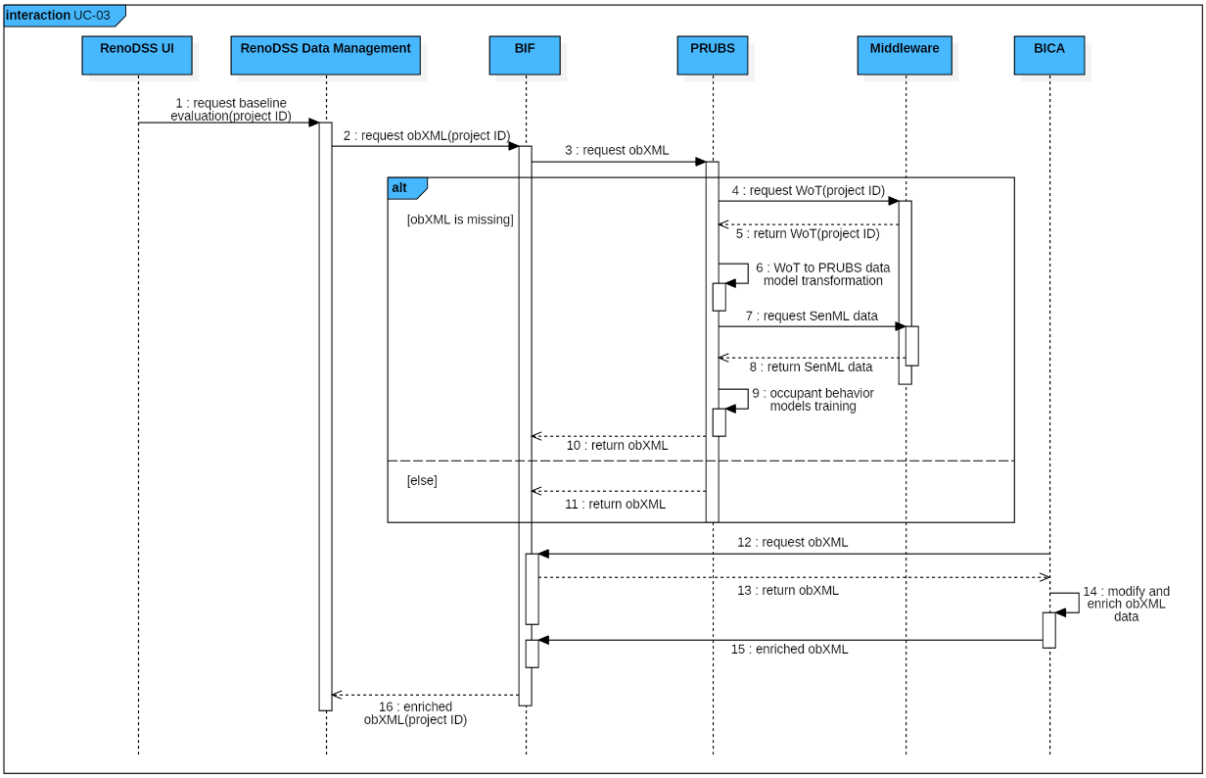

Figure A-0-3 UC-03 Adapt design to the actual building use

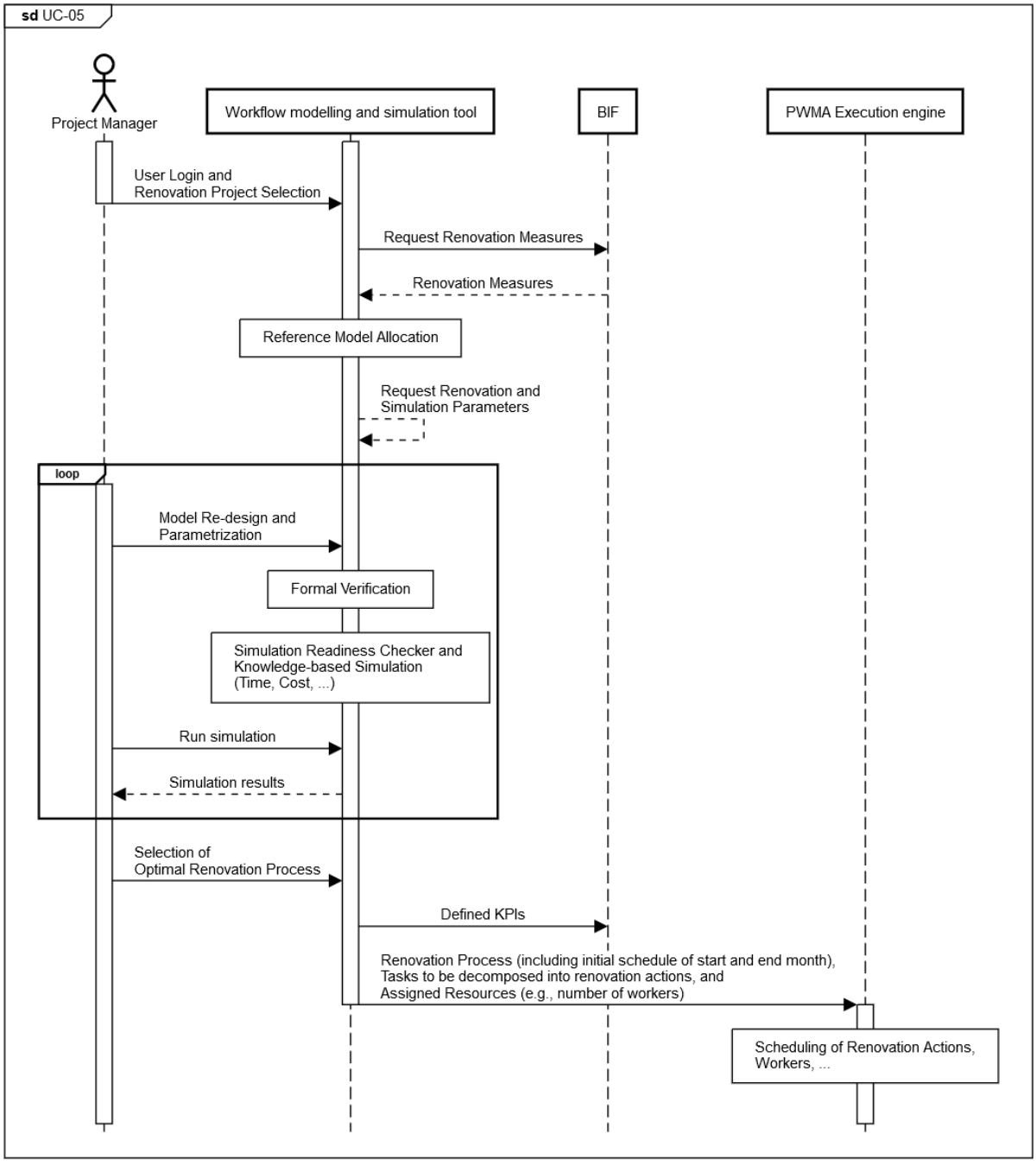

Figure A-0-4 UC-05 Simulation of renovation process and optimal scheduling

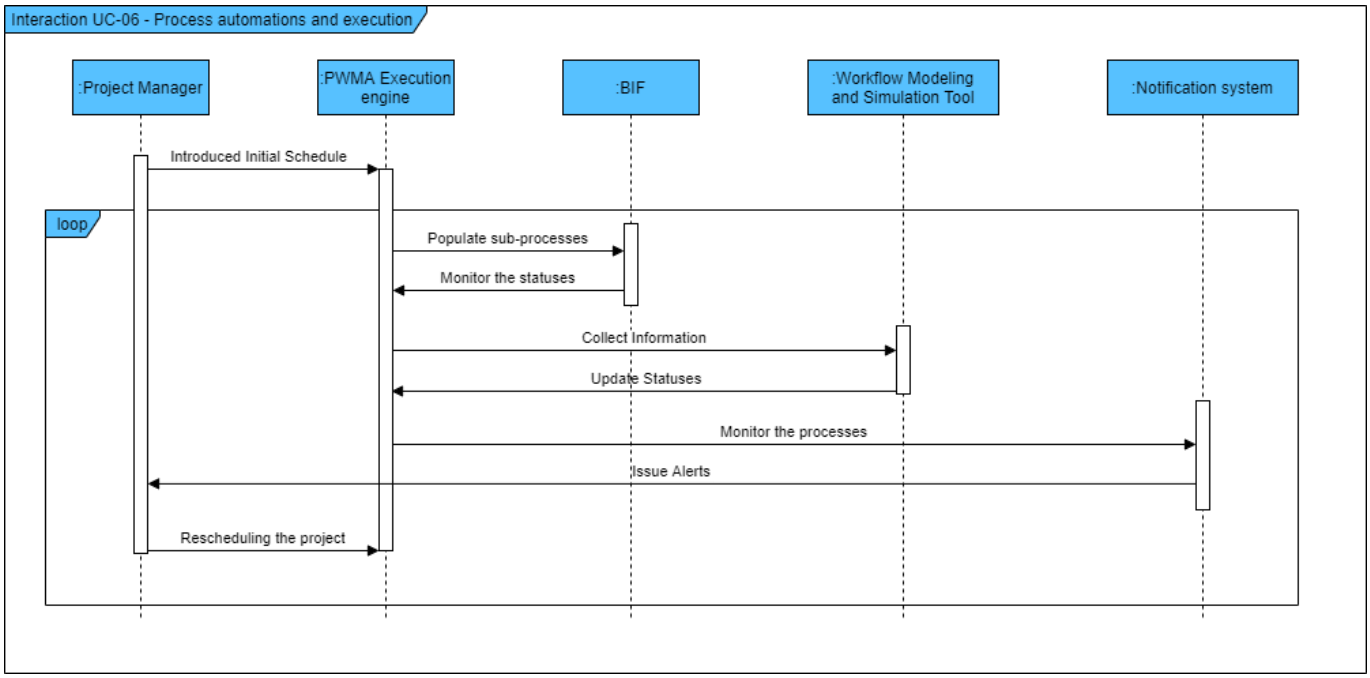

Figure A-0-5 UC-06 Process automation and execution

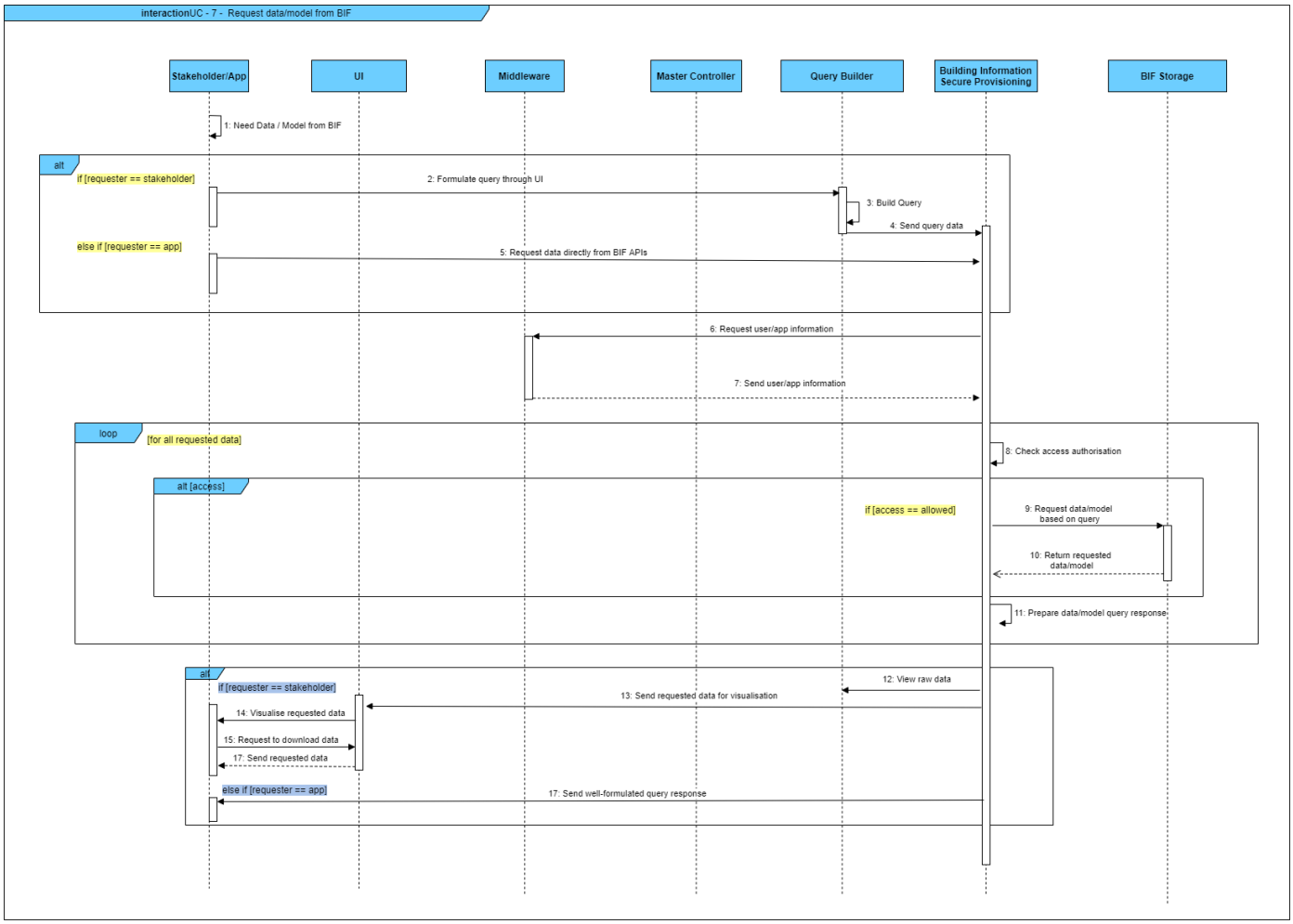

Figure A-0-6 UC-07 Request data model from BIF

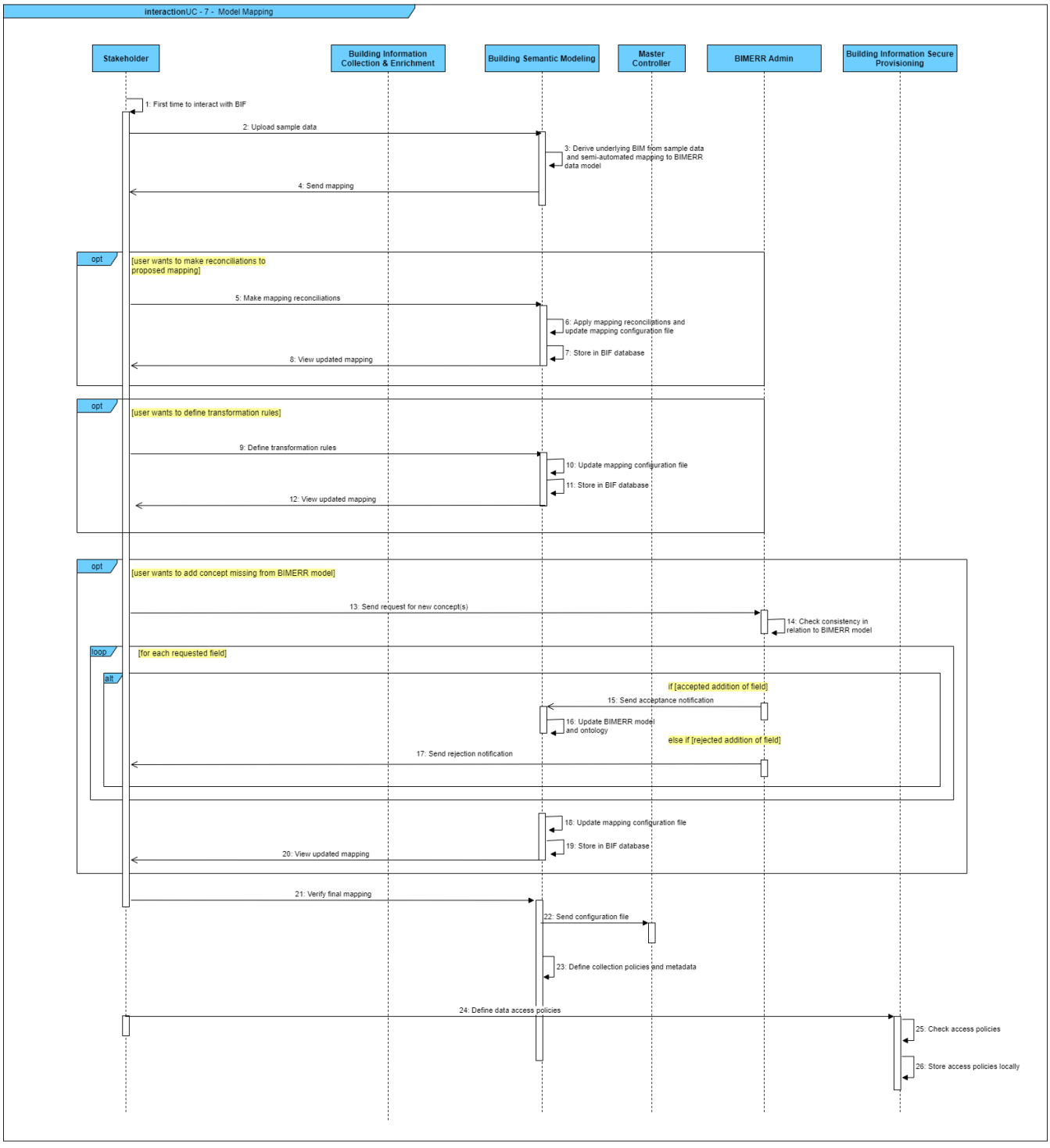

Figure A-0-7 UC-7 Model mapping

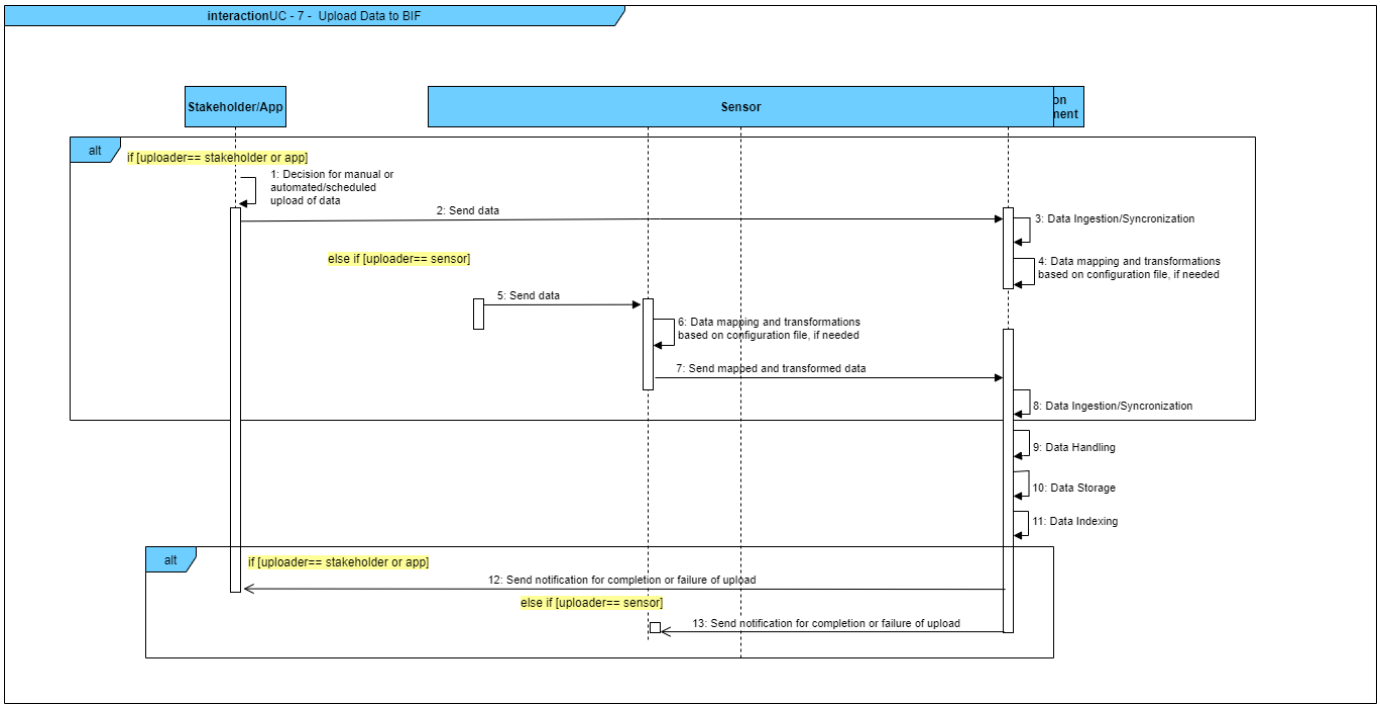

Figure A-0-8 UC-07 Upload data to BIF

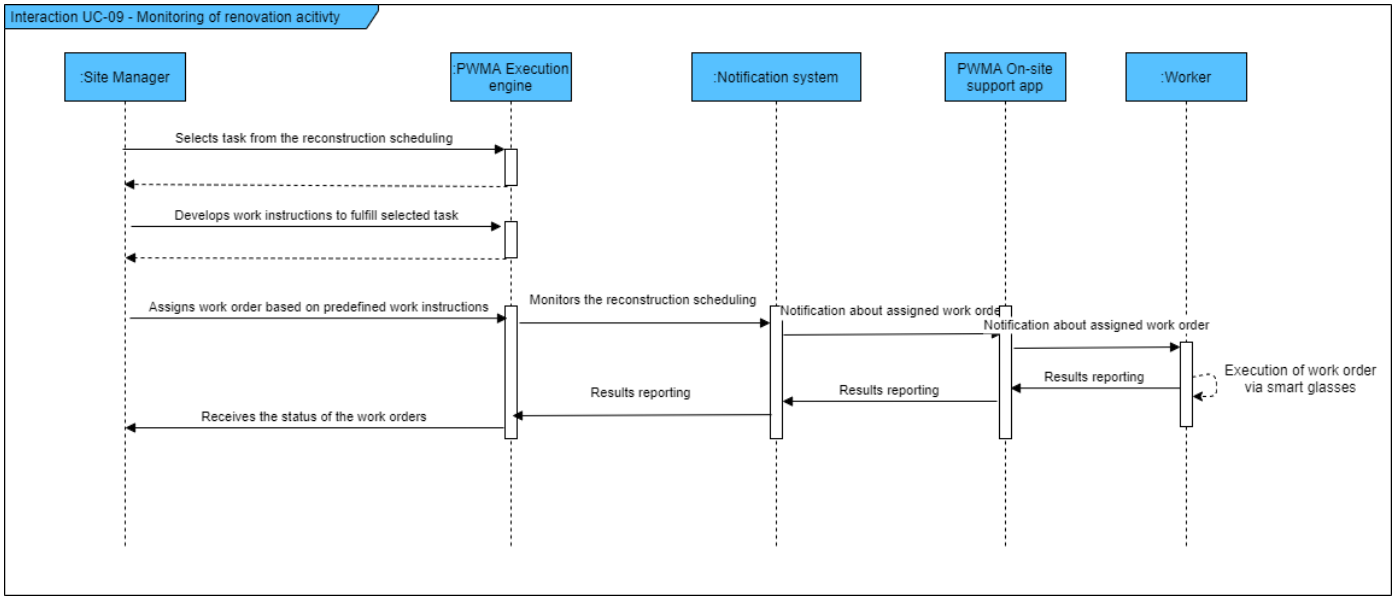

Figure A-0-9 UC-09 Monitoring the renovation activity.