Project Acronym:	**BIMERR**
Project Full Title:	**BIM-based holistic tools for Energy-driven Renovation of existing Residences**
Grant Agreement:	**820621**
Project Duration:	**45 months**

DELIVERABLE D8.4
Integrated BIMERR ICT System 2

Deliverable Status:	**Final**
File Name:	**D8.4 Integrated BIMERR ICT System 2-v1.0.docx**
Due Date:	**30/09/2021 (M33)**
Submission Date:	**30/09/2021 (M33)**
Task Leader:	**FIT (T8.4)**

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**
	Bosché	Frédéric	UEDIN	f.bosche@ed.ac.uk
	Valero	Enrique	UEDIN	e.valero@ed.ac.uk
	Bountouni	Nefeli	SUITE5	nefeli@suite5.eu
	Lampathaki	Fenareti	SUITE5	fenareti@suite5.eu
	Varga	Ján	NT	varga@novitechgroup.sk
	Falcioni	Damiano	BOC	damiano.falcioni@boc-eu.com
	Fenz	Stefan	XYLEM	fenz@xylem-technologies.com
	Giannakis	Giorgos	HYPERTECH	g.giannakis@hypertech.gr
	Katsos	Marios	HYPERTECH	m.katsos@hypertech.gr
	Kompos	Kostas	HYPERTECH	k.kompos@hypertech.gr
	Iousef	Samy	HYPERTECH	s.iousef@hypertech.gr
	Katsigarakis	Kyriakos	UCL	k.katsigarakis@ucl.ac.uk
	Krauß	Veronika	FIT	veronika.krauss@fit.fraunhofer.de
	Lilis	Georgios	UCL	gnl2@cornell.edu
	Straka	Martin	NT	straka@novitechgroup.sk
	Tavakolizadeh	Farshid	FIT	farshid.tavakolizadeh@fit.fraunhofer.de
	Tsakiris	Athanasios	CERTH	atsakir@iti.gr
	Vafeiadis	George	UBITECH	gvafeiadis@ubitech.eu
	Zarbouti	Dimitra	UOP	dzarb@uop.gr

REVIEWERS LIST

List of Reviewers (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Giannakis	Giorgos	HYPERTECH	g.giannakis@hypertech.gr
2	Lampathaki	Fenareti	SUITE5	fenareti@suite5.eu

REVISION CONTROL

Version	Author	Date	Status
0.1	Shreekantha Devasya, Farshid Tavakolizadeh	15/06/2021	Initial contents taken from D8.3, updates to verification tasks. Updates intro, summary, conclusions. Addition of middleware refinements.
0.2	Farshid Tavakolizadeh	20/07/2021	Extend middleware refinement.
0.4	Otilia Werner-Kytölä	07/09/2021	Updates to table 4.2
0.5	Otilia Werner-Kytölä	09/09/2021	RenoDSS refinements introduced in section 7.2
1.0	Otilia Werner-Kytölä	22/09/2021	Implemented changes from reviews, updated table 4.2, added table 4.3

TABLE OF CONTENTS

List of Figures *8*

List of Tables *9*

EXECUTIVE SUMMARY *10*

1. INTRODUCTION *11*

1.1 Scope of the Deliverable 11

1.2 Structure of the Deliverable 11

1.3 Relation to the Other Activities and Deliverables 12

1.4 Updates to the First Version of the BIMERR Integrated ICT System 13

1.5 Overview of the BIMERR Systems 13

2. The Integration and Testing Methodology *16*

2.1 Integration Testing 18

2.2 Tracking the Integration Testing Activities 19

3. Packaging, Delivery and Deployment of BIMERR System *22*

3.1 Overview 22

3.2 Version Control and Delivery of BIMERR Components 24

3.3 Deployment Requirements 28

4. Integration and Testing Tasks *29*

4.1 Use Cases and the Components 29

4.2 Use Case Interaction Verification Status 32

5. Secure Information Exchange 40

5.1 Authentication 40

5.2 Authorization 41

5.3 BIMERR Identity Groups and Roles 42

5.4 Status of the Integration between the Identity Provider and BIMERR Components 43

6. Unified UI Designs and UX Alignments 44

6.1 Usage of the Logo 45

6.2 Colors and Color Schema 46

6.3 Icons 49

6.4 Typography 49

7. Refinement Activities 52

7.1 BIMERR Middleware 52

7.1.1 Registry 52

7.1.2 Storage 53

7.1.3 Data Processor 53

7.1.4 OTA Update and Monitoring 55

7.2 BIMERR RenoDSS 56

7.2.1 Integration of components and internal gains 56

7.3 BIMERR ARIBFA 57

7.3.1 Integration of BIF interactions 57

7.4 BIMERR PWMA for residents 58

8. Conclusions 59

9. Bibliography 60

A. Appendix - Updated Use Case Sequence Diagrams 61

B. Appendix: Project Creation and User Management 70

LIST OF FIGURES

Figure 1-1 High level view of components and the data flow (taken from D3.6) 14

Figure 2-1 Different stages of testing and corresponding tasks 17

Figure 2-2 Jira item BIM-52 as part of the realization of UC-01 20

Figure 2-3 Jira item BIM-59 as part of the realization of UC-04 21

Figure 3-1 LinkSmart Thing Directory documentation on GitHub's Wiki 23

Figure 3-2 Sensor data query documentation on BIMERR Wiki 23

Figure 6-1 Example for the application of the color schema 48

Figure 6-2 Type face for titles, headlines, and subtitles - Josefin Sans Medium 500 49

Figure 6-3 Type face for paragraphs – Roboto Regular 400 50

Figure A-0-1 UC–01: As-is Data Capturing and Processing for Scan-to-BIM 62

Figure A-0-2 UC-02: Mapping and annotation process 63

Figure A-0-3 UC-03 Adapt design to the actual building use 64

Figure A-0-4 UC-05 Simulation of renovation process and optimal scheduling 65

Figure A-0-5 UC-06 Process automation and execution 66

Figure A-0-6 UC-07 Request data model from BIF 67

Figure A-0-7 UC-7 Model mapping 68

Figure A-0-8 UC-07 Upload data to BIF 69

Figure A-0-9 UC-09 Monitoring the renovation activity. 69

LIST OF TABLES

Table 1-1 Relation to other BIMERR project's deliverables and tasks 12

Table 2-1 Verification means for interactions. 19

Table 3-1 Packaging, delivery, and documentation of BIMERR components 24

Table 4-1 The use cases, components and the partners responsible for realization 29

Table 4-2 Interactions and the status of their verifications..... 33

Table 4-3 BIMERR ICT System – Level of completeness of UCs at M33 37

Table 5-1. BIMERR to Identity Provider attribute mapping 41

Table 5-2. List of BIMERR identity groups 42

Table 5-3. List of BIMERR identity roles..... 43

Table 6-1 Overview over applications, targeted platforms, responsible partners and affected tasks 44

Table 6-2 Various versions of the logo 45

Table 6-3 Defined colors and their usage 46

Table 6-4 Examples for two-tone library in black and white 49

Table 6-5 Available scale categories 50

EXECUTIVE SUMMARY

The current deliverable is the second 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 the individual components' development tasks. This deliverable presents the version control, delivery and deployment strategies of the individual components. In the end, it also documents the refinements performed to the individual components, as part of the integration and demonstration activities.

System integration testing verifies the functional operation of the end-to-end system with the right information flowing across its different services and applications in a secured manner. Validation of the information flow and end-to-end integration utilize the BIMERR Use Cases as the foundation. The interactions between different components towards the achievement of each use case's objectives are being validated, either manually or programmatically, or with the help of demonstrations.

Securing the BIMERR system and data from unauthorized access is ensured by integrating the BIMERR 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 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 as well as driving the activities 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 second edition of the deliverable series (D8.3, D8.4 and D8.5), this deliverable considers the deliverable D8.3 as the baseline which covers all the aforementioned activities.

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 the 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. Chapter 6 provides guidelines for uniform UI designs and consistent user experience across BIMERR applications. Finally, Chapter 7 covers the refinement activities of different BIMERR components, carried out between M26 and M33.

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

1.3 Relation to the Other Activities and Deliverables

Table 1-1 depicts the relations of this document to other deliverables and tasks within the BIMERR project, that should be considered along with this document for further understanding of its contents.

Table 1-1 Relation to other BIMERR project's deliverables and tasks

Del./Task Number	Deliverable Title	Relations and Contribution
D3.1 [4]	Stakeholder requirements for the BIMERR system	It introduces 16 Use Cases describing the utilization of the BIMERR system. D8.4 uses these Use Cases as the baseline for validating the integrated system
D3.6 [5]	BIMERR system architecture final version	The final version of the BIMERR architecture is delivered. D3.6 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. D8.4 uses it as the basis for the BIMERR software and the information flow between the components.
D8.6 [6]	Report on BIMERR pre-validation activities	It gives an overview of the pre-validation activities of the integrated BIMERR system. Also due on M33.
T8.4	BIMERR ICT System Pre-Validation	It is related to the BIMERR pre-validation activities. It helps in the validation of the integrated BIMERR system and provides constantly feedback to Task 8.3.
T9.3	BIMERR Tool Roll-Out & Deployment for Demonstration & Validation Activities	The findings and feedback from these activities helps in validating the integrated BIMERR system.
T9.4	Demonstration of BIMERR Tools on Real Renovation Projects	The findings and feedback from these activities helps in validating the integrated BIMERR system.

Additionally, the tasks responsible for the development of components as part of WP4-WP8 and their corresponding deliverables are also connected to the current deliverable D8.4 and are listed in Table 3-1 of this document.

1.4 UPDATES TO THE FIRST VERSION OF THE BIMERR INTEGRATED ICT SYSTEM

Since the submission of D8.3 in M26, the BIMERR Integrated ICT System has evolved as in the following:

- Updates to the verification of integration: The integration is verified based on the BIMERR Use Case sequence diagrams described in D3.6. More specifically, in the current deliverable (D8.4), the column *Status* of Table 4-2, documents the progress of the integration work.
- Refinements of the BIMERR components based on their further development and on the outcome of the integration, pre-validation and demonstration activities. The refinements per component were documented into the respective deliverables in M30, i.e., D4.5, D4.7, D4.9, D5.2, D5.4, D5.6, D5.8, D5.10, D6.3, D6.5, D6.7, D6.9, D6.11, D7.2, D7.4, D7.6, D7.8, and D7.10.
- For both BIMERR tools RenoDSS and ARIBFA, further refinements which were finalized between M31-M33 are described in detail in section 7.2 and in section 7.3, respectively.
- The BIMERR middleware has gone through refinements in many of its components, i.e., Registry, Storage, Data Processor, OTA Update and Monitoring. All refinements are detailed in section 7.1. They include, but are not limited to, new API endpoints compliant with the latest version of the W3C WoT Discovery, auto reboot on failure, support of new sensors, improved alerting, compression of sensor measurements, and outlier detection.

1.5 OVERVIEW OF THE BIMERR SYSTEMS

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 the overall flow of information across its 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.

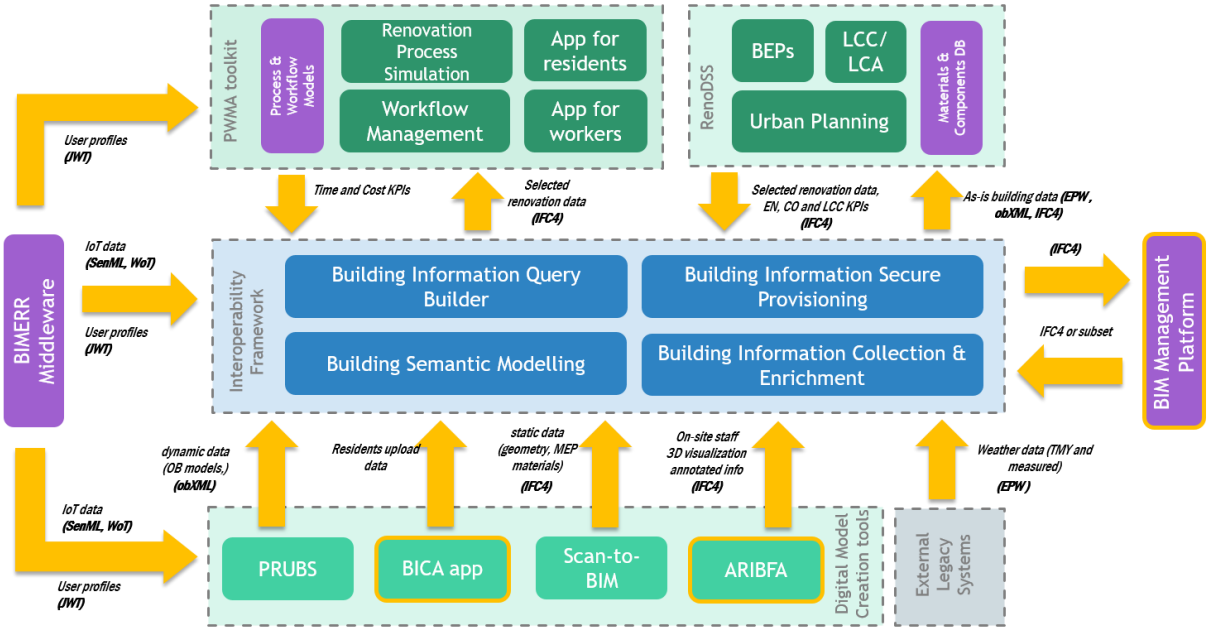


Figure 1-1 High level view of components and the data flow (taken from D3.6)

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 Building Information Collection Application (BICA) to allow the occupants to upload information that cannot be captured by any other means and by the AR-enabled In-situ Building Feature Annotation (ARIBFA) app 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

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

initiates the generation and evaluation of renovation scenarios, which are combinations of renovation measures. Afterwards, the RenoDSS user selects the scenario that meets his/her requirements and Key Performance Indicators (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 design of individual components and their overall role in BIMERR are 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 the 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

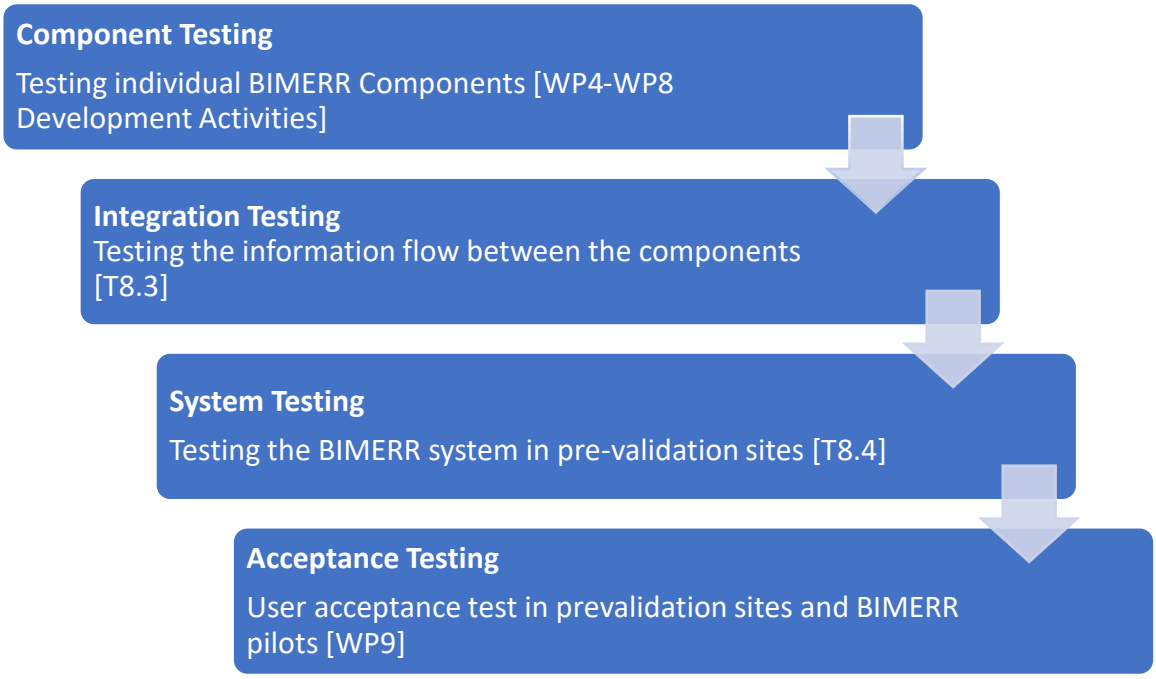


Figure 2-1. We validate different levels of abstraction of the BIMERR software at different testing stages. We identify the stages as: component development, system integration, pre-validation and final validation where we perform component testing, integration testing, system testing and user acceptance testing. These phases are derived from the

specifications defined by Baresi, L. and Pezze, M. [1]. This is an iterative process, and it shall be executed 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 criteria 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. The result of the component development and component testing will be a complete functional component ready to be integrated with the BIMERR system.

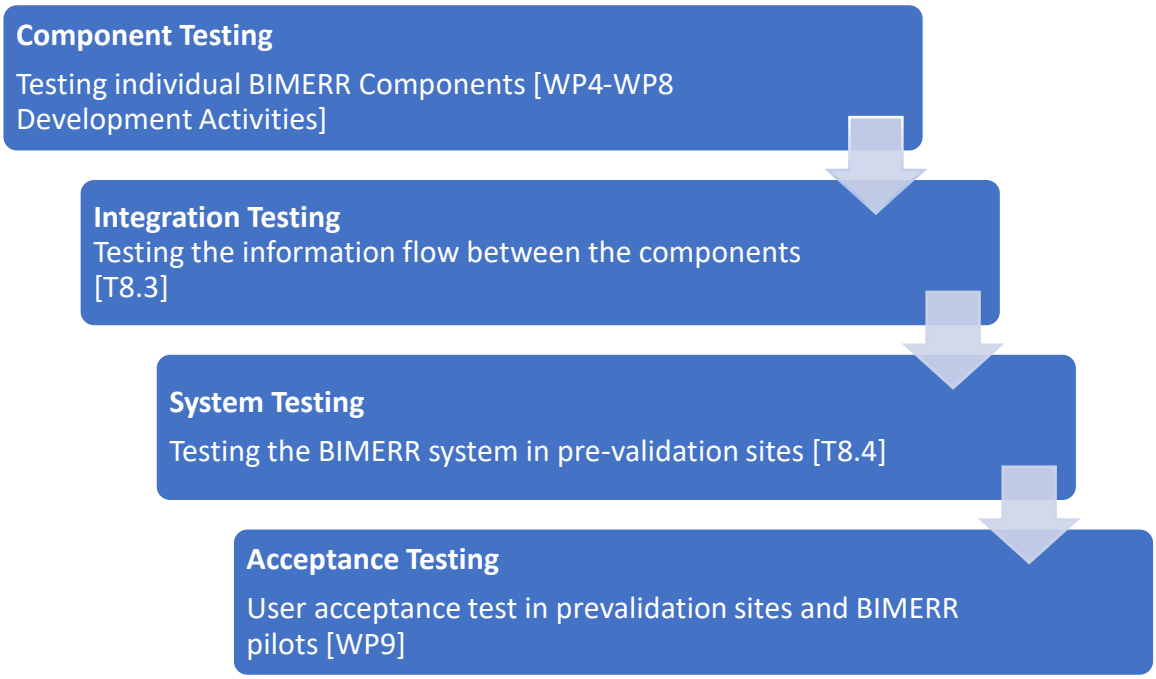


Figure 2-1 Different stages of testing and corresponding tasks

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 the 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 activities help the BIMERR consortium to refine the components and the overall BIMERR system. The results 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 can fulfill the user expectations. The stakeholders related to the pre-validation sites and 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 results 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 module [3].

As mentioned earlier, the sequence diagrams of the BIMERR 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 of 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 visual walkthrough (images, video) or an undoctored recorded presentation. The materials 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. Some examples are shown in Figure 2-2 and Figure 2-3. 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

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

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.

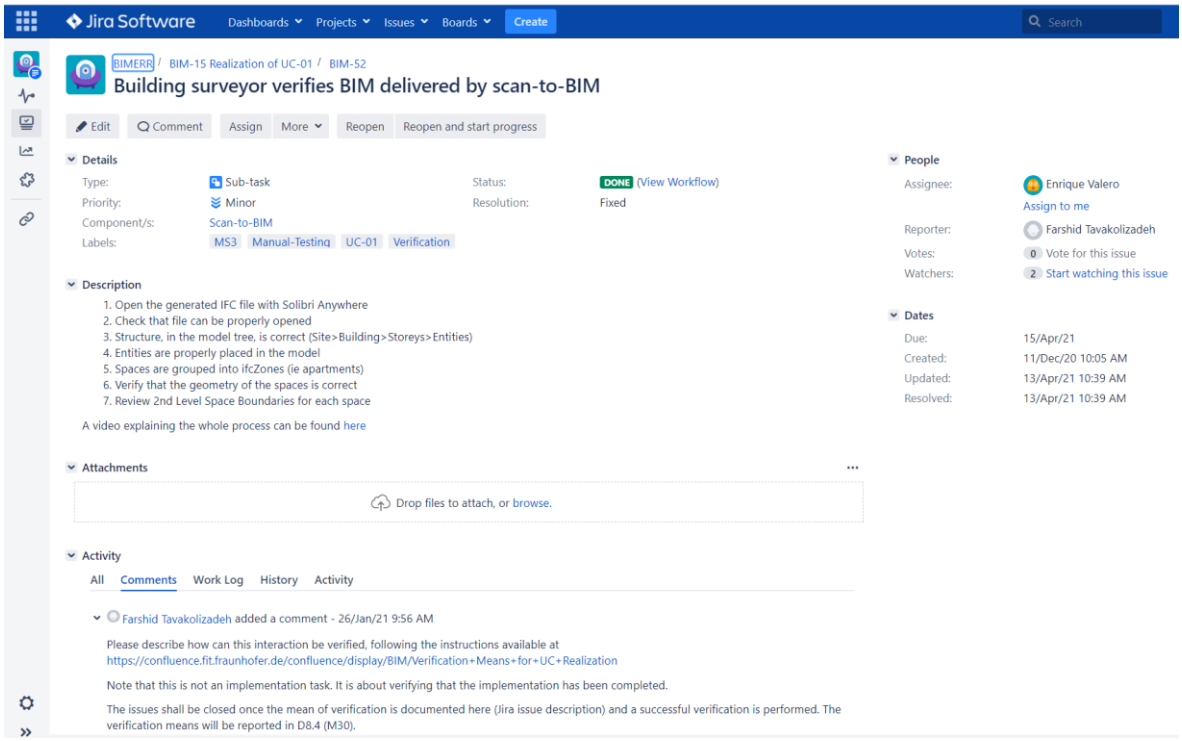


Figure 2-2 Jira item BIM-52 as part of the realization of UC-01

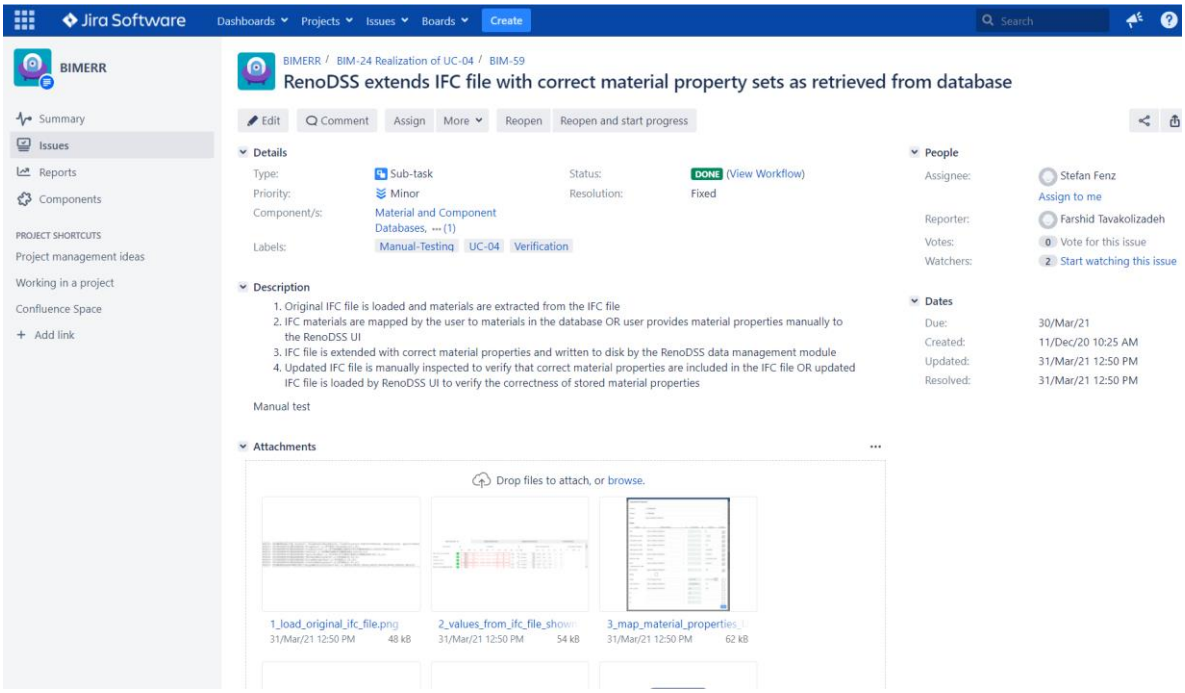


Figure 2-3 Jira item BIM-59 as part of the realization of UC-04

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

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

Documentation of the BIMERR components is found in one of:

- Wiki page of a public repositories, if applicable (Figure 3-1)
- BIMERR Deliverables corresponding to the components
- Project's internal wiki pages accessible to the project partners (Figure 3-2)

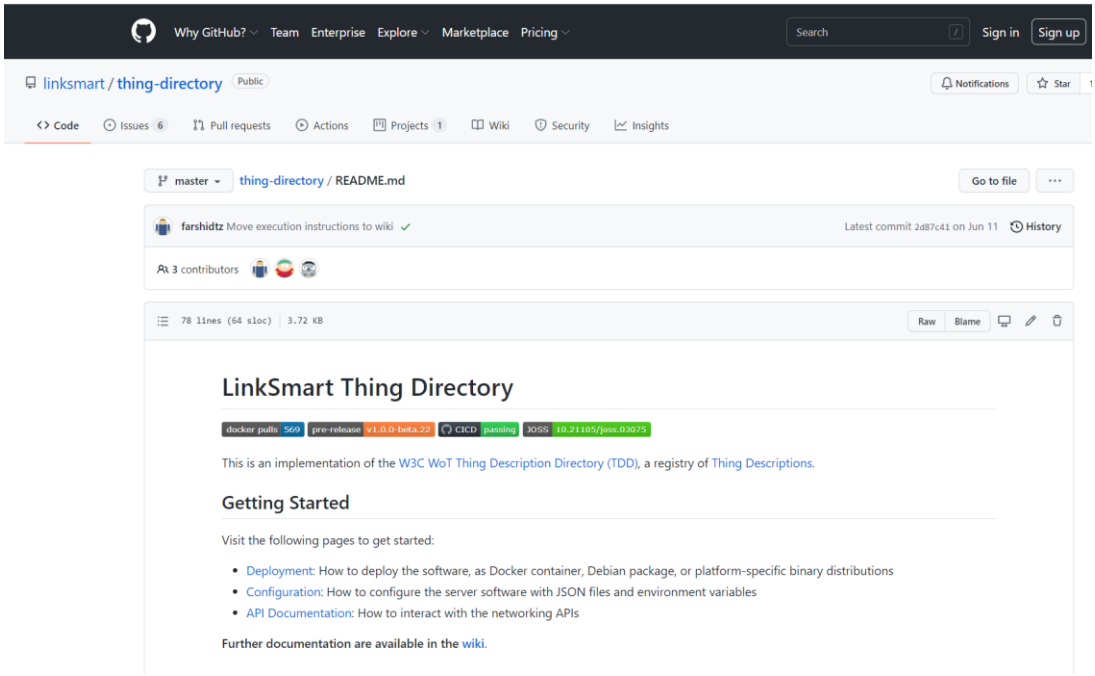


Figure 3-1 LinkSmart Thing Directory documentation on GitHub's Wiki

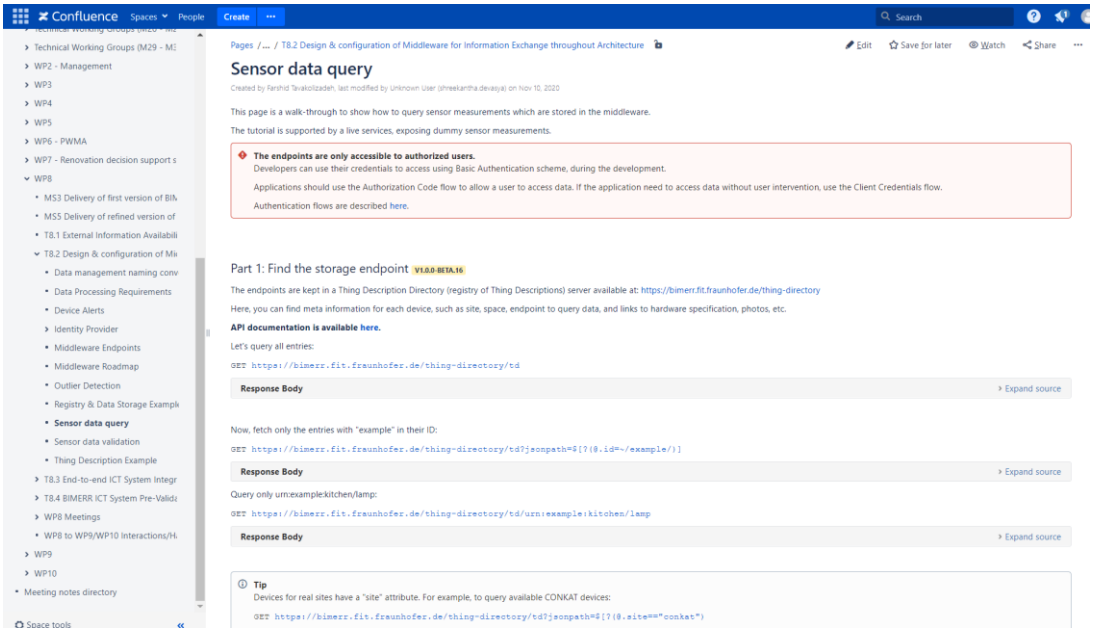


Figure 3-2 Sensor data query documentation on BIMERR Wiki

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)	Private GitHub repositories	Deliverables D4.4 and D4.5	Closed source Note: UPM component (OMF) is available as open source
BIF: Building Information Collection & Enrichment	SUITE5	Software as a service (SaaS)	Private GitHub repositories	Deliverables D4.6 and D4.7	Closed source Note: UPM component (KGG) is available as open source
BIF: Building Information Query Builder	UBITECH	Software as a service (SaaS)	Private Gitlab repository	Deliverables D4.8 and D4.9	Closed source
BIF: Building Information Secure Provisioning	UBITECH	Software as a service (SaaS)	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	Private GitHub repository Will be released as open source by the end of project.	• 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)	Private repository	Deliverable D5.1 and D5.2	Closed source
ARIBFA	CERTH	HoloLens UWP App Binary	Private Gitlab repository	Deliverables D5.9 and D5.10	Closed source
BICA	SUITE5	Android Mobile App in Play Store	Private GitHub repository	Deliverables D5.5 and D5.6	Closed source
PRUBS	HYPERTECH	Software as a service (SaaS)	Private Gitlab repositories	Deliverables D5.7, D5.8	Closed source
RenoDSS: Building Material and Components Database	XYLEM	Software as a service (SaaS)	Private Gitlab repositories	Deliverables D7.1 and D7.2	Closed source
RenoDSS: Renovation LCA/LCC Module	XYLEM	Software as a service (SaaS)	Private Gitlab repositories	Deliverable D7.3 and D7.4	Closed source
RenoDSS: Building Energy Performance Modelling Module	HYPERTECH	Software as a service (SaaS)	Private Gitlab repositories	Deliverables D7.5 and D7.6	Closed source
RenoDSS: Urban Planning Module	XYLEM	Software as a service (SaaS)	Private GitLab repositories	Deliverable D7.7 and D7.8	Closed source

Component	Responsible partner	Delivery (distribution) form	Source code repository	Documentation	Software License
RenoDSS: Decision Support System Engine and UI & Module Integration	XYLEM	Software as a service (SaaS)	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)	Private GitLab repositories	Deliverables D6.4 and D6.5	Closed Source
PWMA: Renovation Process Simulation tool	BOC	Binary distribution	GitLab (public)[12]	Deliverables D6.4 and D6.5	Apache 2.0
PWMA: Adaptive Workflow Management and Automation tool	NT	Software as a Service (SaaS)	Private GitLab repositories	Deliverables D6.6 and D6.7	Closed Source
PWMA: Smart glass application for on-site renovation worker support	NT	Both as Mobile App (downloadable package) and Hololens App	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	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

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, 1997 [2].

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 the different BIMERR 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 the different use cases and the components involved in them. 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 several BIMERR partners. As part of the integration testing activities, we shall validate their realization. Some Use Cases have their realization highly coupled with other Use Cases. In this situation, the integration testing activities are merged together, as for UC-08 and -09 and for UC-11 and -12 as well as UC-13, -14, -15 and -16.

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

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components
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
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
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

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components
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 • BIF
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.		
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
UC-14	Energy performance assessment to be elevated at a life-cycle perspective including relevant LCA-LCC metrics		

Use Case ID	Name	Partners (*: the partner leading the UC realization)	Components
UC-15	Energy performance simulations to assess not only energy metrics, but also accurately evaluate occupants' comfort and indoor air quality		
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.

As already explained in the integration and testing methodology section, more specifically in section 2.2, the Jira tool is used to track the integration activities as shown in Figure 2-2 and Figure 2-3.

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.

Editor's note:

https://jira.fit.fraunhofer.de/jira/issues/?filter=14404 > export as CSV with "current fields" > download to a csv and create the following table using import functions of the excel file (https://fraunhofer.sharepoint.com/:x:/s/BIMERR-FIT975/EXQ62Qr7AdJGnLw3nEcaflYBjt-Iw3rH1kd2H5g0EAvyjA?e=DnJtFQ) using Data->"Queries & Connections"

Table 4-2 Interactions and the status of their verifications

Use Case/s	Summary	Components	Means of Verification	Status
UC-01	Building surveyor verifies BIM delivered by scan-to-BIM	Scan-to-BIM	Manual-Testing	Done
UC-01	BIM Management Platform checks IFC geometry and sends file back to building surveyor	BIM Management Platform	Automated-Testing	Done
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	Done
UC-01	Building surveyor creates a new project and uploads the BIM model (i.e., IFC file) to the BIF	BIF	Manual-Testing	Done
UC-02	User (building occupant) logs in to her BICA app account.	BICA	Manual-Testing	Done
UC-02	BICA receives from BIF the related parts of the apartment's IFC, components, comfort data.	BICA, BIF	Manual-Testing	Done
UC-02	BIM-MP Receives BIM IFC from ARIBFA	ARIBFA, BIF, BIM Management Platform	Manual-Testing	Done
UC-02	ARIBFA Receives Error report from BIM-MP	ARIBFA, BIM Management Platform	Manual-Testing	Done
UC-02	ARIBFA Receives BIM Model from BIF	ARIBFA, BIF	Manual-Testing	Done
UC-02	BICA receives from Middleware the sensor measurements from the apartment.	BICA, Middleware	Manual-Testing	Done
UC-02	The other BIMERR tools get the submitted information in BICA from BIF	BICA, BIF	Manual-Testing	Done
UC-02	BIF Receives BIM IFC from ARIBFA	ARIBFA, BIF	Manual-Testing	Done
UC-02	BICA displays the apartment information to the user through the UI.	BICA	Manual-Testing	Done
UC-02	User submits additional information through the BICA UI	BICA	Manual-Testing	Done

Use Case/s	Summary	Components	Means of Verification	Status
UC-03	BIF-RenoDSS interaction: RenoDSS receives obXML from BIF	BIF, RenoDSS	Manual-Testing	Done
UC-03	PRUBS - BIF interaction: BIF receives obXML from PRUBS	BIF, PRUBS	Demonstration	Done
UC-03	BICA - BIF interaction: BICA receives and sends OB data to BIF	BICA, BIF	Manual-Testing	Done
UC-03	PRUBS - Middleware interaction: PRUBS receives WoT and SenML data from Middleware	Middleware, PRUBS	Automated-Testing	Done
UC-04	RenoDSS UI retrieves materials and their properties from the database	Material and Component Databases, RenoDSS	Manual-Testing	Done
UC-04	Database receives new/updated material properties from RenoDSS UI	Material and Component Databases, RenoDSS	Manual-Testing	Done
UC-04	RenoDSS extends IFC file with correct material property sets as retrieved from database	Material and Component Databases, RenoDSS	Manual-Testing	Done
UC-05	Project manager selects optimal renovation process and defines KPIs using Workflow modelling and simulation tool	PWMA	Manual-Testing	Done
UC-05	Project manager runs simulation	PWMA	Manual-Testing	Done
UC-05	Project manager redesigns reference model, sets required parameters and formally verifies model using Workflow modelling and simulation tool	PWMA	Manual-Testing	Done
UC-05	Project manager selects renovation project in Workflow modelling and simulation tool	PWMA	Manual-Testing	Done
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 tool sends updated schedule to the PWMA Execution engine	PWMA	Manual-Testing	Done
UC-06	PWMA Workflow and simulation modelling tool gets logs from the PWMA Execution engine	PWMA	Manual-Testing	Done
UC-06	PM creates Initial Schedule in the PWMA Execution engine	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.	BIF, Middleware	Manual-Testing	Done

Use Case/s	Summary	Components	Means of Verification	Status
UC-07	User sees a list of available data assets in the BIF UI, based on the applied access policies.	BIF	Manual-Testing	Done
UC-07	The uploaded data are available for search from BIF users	BIF	Manual-Testing	Done
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	BIF	Manual-Testing	Done
UC-07	Add metadata and access policies to data	BIF	Manual-Testing	Done
UC-07	Mapping verification and finalisation	BIF	Manual-Testing	Done
UC-07	Changes to mapping configuration (before it is finalised)	BIF	Manual-Testing	Done
UC-07	User creates new concept request	BIF	Manual-Testing	Done
UC-07	BIF apply data model changes to previously stored files and communicate changes to Ontology Manager Framework	BIF	Manual-Testing	Done
UC-07	Make changes to the BIMERR data model through the Model Lifecycle Manager	BIF	Manual-Testing	Done
UC-07	The user receives a notification in BIF about the status/progress of her concept request.	BIF	Manual-Testing	Done
UC-07	Mapping corrections through the BIF UI	BIF	Manual-Testing	Done
UC-07	User previews the results of the query in the integrated console in the BIF UI	BIF	Manual-Testing	Done
UC-07	User creates query from the available data assets. BIF provides the user with an API endpoint to retrieve data	BIF	Manual-Testing	Done
UC-07	User uploads data as file to BIF	BIF	Manual-Testing	Done
UC-07	User/app uses the API and the authorization token she has been provided from BIF to retrieve the data	BIF	Manual-Testing	Done
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	Manual-Testing	Done
UC-07	BIF receives from the user the sample data file.	BIF	Manual-Testing	Done
UC-07	User creates a new data collection process in BIF	BIF	Manual-Testing	Done
UC-08, UC-09	As worker moves through the construction site, his/her location is continuously identified	ARIBFA	Demonstration	Done

Use Case/s	Summary	Components	Means of Verification	Status
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 sent to PWMA On-site support app. Worker can see appropriate work orders according to his/her actual location.	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	ARIBFA, PWMA	Manual-Testing	Done
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 notification to the Site manager.	PWMA	Manual-Testing	In Progress
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	Manual-Testing	Done
UC-12	PWMA for Residents receives BIM Model from BIF	BIF	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-13, UC-14, UC-15, UC-16	Data Management retrieves IFC, obXML, and epw files from BIF	BIF, RenoDSS	Manual-Testing	Done
UC-13, UC-14, UC-15, UC-16	RenoDSS UI receives Projects from Data Management	RenoDSS	Manual-Testing	Done
UC-13, UC-14, UC-15, UC-16	BIF retrieves selected renovation scenarios as IFC files and KPIs as JSON file	BIF, RenoDSS	Manual-Testing	Done

Use Case/s	Summary	Components	Means of Verification	Status
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	RenoDSS	Manual-Testing	Done
UC-13, UC-14, UC-15, UC-16	Renovation scenarios are generated by Data Management based on set renovation measures	RenoDSS	Manual-Testing	Done
UC-13, UC-14, UC-15, UC-16	RenoDSS UI retrieves baseline scenario Energy, LCA/LCC, and Urban Planning KPIs from Data Management	RenoDSS	Manual-Testing	Done

In a nutshell, Table 4-3 shows the level of completeness of each UC at M33 related to the number of subtasks of each UC, whereby no differentiation is made between the complexity of subtasks.

Table 4-3 BIMERR ICT System – Level of completeness of UCs at M33

Use Case ID	Name	Level of completeness of UCs based on its number of subtasks
UC-01	Rapid scanning of the geometry of the building, semantic modeling and accurate representation in a BIM	6/7 Done 1/7 In Progress
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.	12/15 Done 3/15 In Progress
UC-03	Adapt design to the actual building use, including accurate information about occupancy and schedules, comfort requirements/ preferences and energy uses.	5/6 Done 1/6 In Progress

Use Case ID	Name	Level of completeness of UCs based on its number of subtasks
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).	5/5 Done
UC-05	Accurate scheduling of activities and assessment of their efficiency through simulation and verification.	6/6 Done
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.	6/8 Done 2/8 In Progress
UC-07	Stakeholders' systems exchange appropriate and "understandable" data between each other.	21/21 Done
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.	9/9 Done
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)	9/9 Done

Use Case ID	Name	Level of completeness of UCs based on its number of subtasks
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).	3/6 Done 3/6 In Progress
UC-11	Identification of threats and dangers and provision of alerts to workforce and occupants through BIM-based apps and UIs	6/9 Done 3/9 In Progress
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.	
UC-13	Perform back-to-back simulations of alternative renovation scenarios to evaluate and select the best energy-performing renovation scenario	6/8 Done 2/8 In Progress
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	
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	

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 desktops, 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[13] 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

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

authentication process does not have to be repeated when a user tries to access other applications in a specified period.

- **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[14] and the OAuth 2.0 Authorization Framework[15] 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/ Data consumer	User	Jane Doe
Project Project ID	Group Group ID	KRIPIS 57f87657-4cc8-4331-9d10-1f89291581ee
Apartment / Zone Zone ID	Subgroup Subgroup's zone_id attribute	Apt-13 3U3HYkrjfEZe6MCxjWbzQ2
User role	Role	Renovation Planner

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

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

Application	Client	BIF

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.

The above attributes can be managed using the user interface (UI) of the Identity Provider; see B. Appendix: Project Creation and User Management.

At the time of writing, authorization is performed at two data providing points: BIMERR 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 presented 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
Construction Worker	BIMERR Construction Worker
Demo Role	This role is for documentations only
Device Maintainer	People who are responsible for maintenance of WSN
Middleware Developer	Application developer
Occupant	Apartment / House occupant
Project Manager	Renovation/Construction project manager
PRUBS Developer	Application developer
PWMA Developer	Application developer with additional access to registration events
RenoDSS developer	Application developer

5.4 Status of the Integration between the Identity Provider and BIMERR Components

As of M33, six of seven BIMERR tools are integrated with the Identity Provider: ARIBFA, BICA, BIF, BIM Management Platform, PRUBS, and RenoDSS. PWMA is being integrated.

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[16] 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[17]	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

[16] https://material.io/

[17] 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
BIF	Suite5	Web browser	Application developers	-	T4.3-T4.6

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

Among the examined logos (Table 6-2), 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). 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

 Black	To be applied on bright colorful backgrounds
 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.

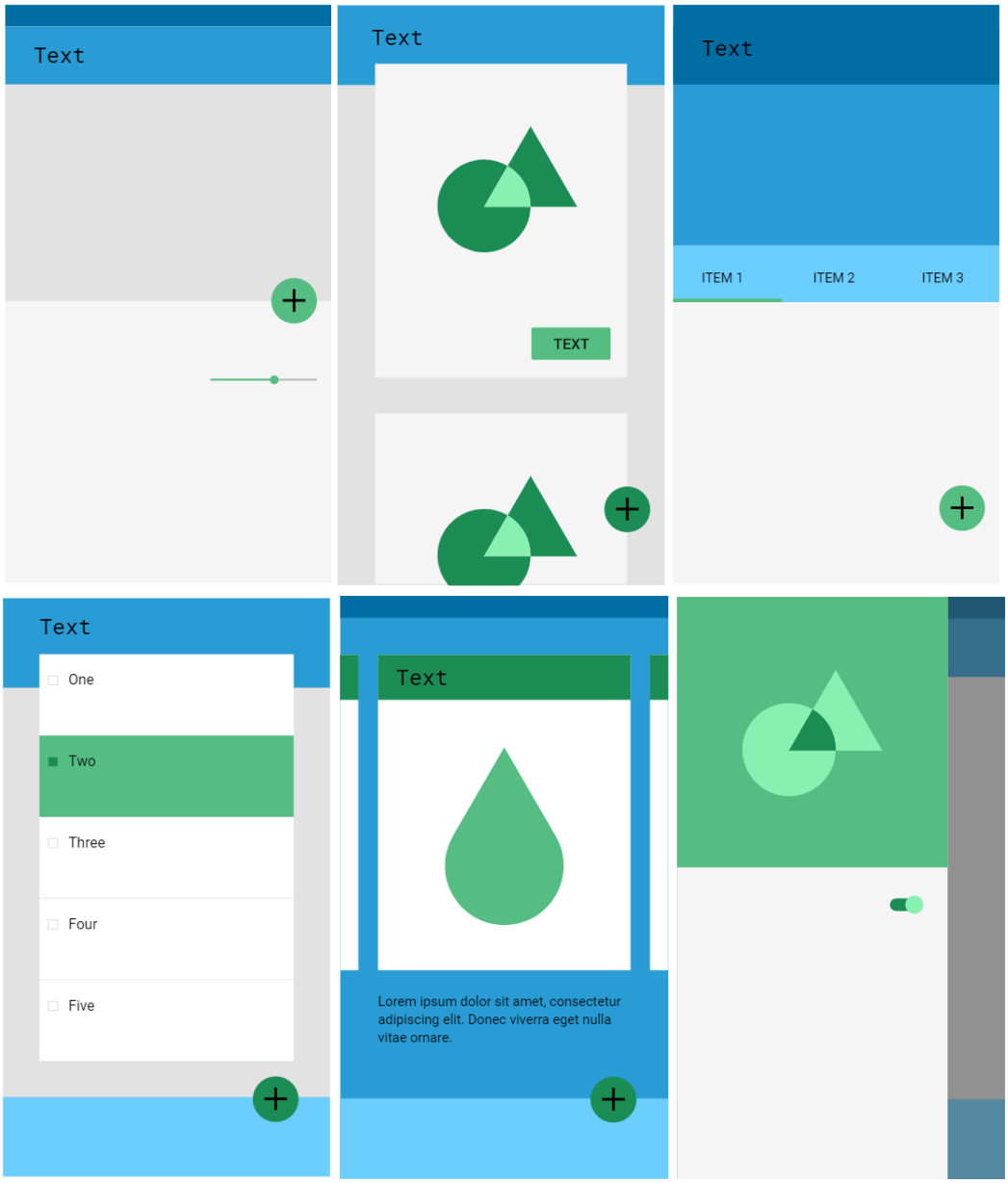


Figure 6-1 Example for the application of the color schema

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[18] 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[19] 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.

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

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

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

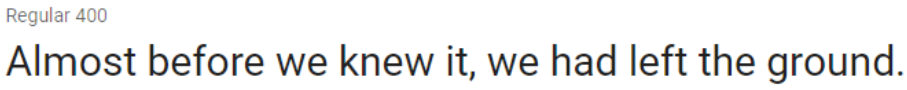


Figure 6-3 Type face for paragraphs – Roboto Regular 400

For headlines, titles, and subtitles the font Josefin Sans Medium 500[20] (Figure 6-2) was selected, whereas paragraphs should be styled in Roboto Regular 400[21] (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

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

[21] 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

7. REFINEMENT ACTIVITIES

The BIMERR components have gone through further refinements as part of the integration and system and acceptance testing. Since the individual tasks related to these components have already finished, the refinements are performed as part of Task 8.3. These refinements are recorded in this section. Note that there are only sub-sections for components that had refinements, excluding components that haven't changed since their last dedicated deliverable.

7.1 BIMERR MIDDLEWARE

The development activities of BIMERR middleware were performed as part of the Task 8.2 and reported in D8.2. In this section we briefly describe the refinements performed on the middleware from M26 to M32.

7.1.1 Registry

The registry component has gone through significant but backward-compatible changes. The refinements are as follows:

- Input validation based on JSON Schema shared by W3C Thing Description specification is added.
- New API endpoints compliant with the latest version of the W3C WoT Discovery specification.
- Notifications are now supported using the Server-Sent Events (SSE) protocol. This notification enables the clients to know the changes to the Thing Descriptions maintained within the Thing Directory.
- Error details are standardized to RFC 7807 (Problem detail for HTTP APIs) specification. Whenever the HTTP requests are not successful, error responses from the server will have a JSON payload describing the problem details.
- A component testing suite for Registry implementations is developed. This can be used by any future Thing directory implementations to validate its compliance to the WoT discovery specifications.

7.1.2 Storage

The core and API of the storage component remains unchanged and stable. The refinements in storage were to extend and optimize the interaction among internal middleware components. The refinements are listed below:

- Error details are standardized to RFC 7807 (Problem detail for HTTP APIs) specification. Whenever the HTTP requests are not successful, error responses from the server will have a JSON payload describing the problem details.
- The gRPC APIs are improved to add logging and error handling.
- Clients can now submit gzip[22] encoded SenML payload to the HDS server. This feature helps the clients to send compressed data to the cloud instance of the Storage and hence reduce network traffic significantly.

7.1.3 Data Processor

The Data Processor was refined to improve reliability, support for new sensors, and improved alerting for failure and remaining battery forecasting. The changes are described below:

Auto reboot on failure

During the live operation on the site, Fibaro Z-Wave gateway (Home Center Lites) has been detected with the problem of continuous failures due to unknown reasons. As a remedy, the Data Retrieval module automatically restarts the gateway whenever such a failure is detected. In some occasions, the gateway completely loses networking access preventing the function of the auto reboot workaround and requiring a physical reboot.

Supporting new sensors

The Data Processor now supports thermostat sensors which are newly procured for the Conkat pre-validation pilot. Moreover, a new set of power meters has been deployed in the Ferrovial site in Spain and integrated with the system for the first time.

[22] https://www.gnu.org/software/gzip/

Improved alerting

The email alerts are improved so that the receivers can have an overall view on the past issues and the newly created and resolved issues. Here is a sample email body:

```
+ [lowBattery] Co Sensor (de:site1:gateway1:22:batteryLevel) last reported 15 at 2021-05-31T00:00:00.000Z
- [offlinePeriod] Fibaro Home Center Lite (FGHCL) (de:site1:gateway1:onlineState) last reported true at 2021-05-31T00:00:00.000Z
- [offlinePeriod] Raspberry Pi Internet Gateway (de:site1:gateway1:pingTime) last reported 75.9 at 2021-05-31T00:00:00.000Z
= [offlinePeriod] Window Sensor 2 (de:site1:gateway1:13:windowState) last reported false at 2020-12-01T23:47:14.000Z

+ : new issue
- : resolved issue
=: existing issue
```

Compressing the sensor measurements

Sensor measurements sent from local gateways to the cloud Storage are now compressed using Gzip encoding to reduce the network traffic usage. The periodic synchronization of compressed sensor data instead of live streaming saves up to 90% in internet upload traffic.

Outlier detection

As the correct and consistent storage of sensor recordings represents the basis of all subsequent data manipulation and analysis methods, the data processor includes a service for unsupervised outlier detection. As described in D8.2, the approach is a multi-step model pursuing the idea of detecting different types of outliers in subsequent filtering stages. One important improvement since then, is the updated initialization procedure, which automatically fetches suitable sensor streams from the registry. An inclusion of streams to this process is done via setting a flag in the corresponding configuration file of the registry. In addition, the online computation of statistical metrics is optimized to more accurately approximate values such as quantiles during incremental outlier detection using a type of random sampling instead of a predefined prioritization of recent values. The result of the outlier detection routine is an indication of potential outliers and their type. While an evaluation of the performance is restricted due to a non-planned feedback loop, the outcome of the outlier detection serves as additional analysis

information to optimize the result of following components and data-related methods. The result is stored as an annotation stream for each sensor in the storage component.

Low battery level prediction

To ease the maintenance routines of deployed sensors and to improve the consistency of resultant measurements, the data processor deals with an early detection of low battery levels. A separate dockerized service approximates the values of earlier battery level behaviors and derives the time until a predefined level will be reached. The current level is set to 20%, meaning that scheduling of multiple sensor replacements can be planned beforehand. While the list of sensors is automatically taken from the registry component, only a restart of the service is needed to collect newly added sensors. The prediction of future battery levels is performed by a regularized regression model. In addition, further constraints on the input data are considered in a preprocessing phase to deal with missing values, irregular battery level behaviors and newly deployed batteries. The outcome of the prediction is an integer embedded in SenML, representing the remaining days until the configured threshold will be reached. Moreover, a recording of predictions over time is additionally done via the middleware storage component.

7.1.4 OTA Update and Monitoring

The original plan was to use the LinkSmart Deployer[23] to enable remote update and monitoring of the software on gateway devices. While this component is open source and available, it is currently not maintained and a security risk. Instead, we chose Salt[24] which offers a better set of features but lacks a powerful graphical user interface (GUI). We utilize Salt's command line interface (CLI) which is well documented and comprehensive.

[23] https://github.com/linksmart/deployer

[24] https://saltproject.io/

Salt's server component (Salt Master) has been deployed on the cloud and instances of Salt's clients (Salt Minions) have been deployed on gateways (Raspberry Pis) in all pre-validation and validation sites. This provides the middleware maintainers a way to quickly diagnose issues and roll-out updates in bulks.

7.2 BIMERR RENODSS

The development activities of BIMERR RenoDSS were performed as part of WP7 and reported in D7.2, D7.4, D7.6, D7.8, and D7.10. At M30, RenoDSS supported the LCA/LCC, energy, urban planning KPI calculation and renovation measure scenario generation in the context of layer-based constructions (walls, floors, roofs, slabs). In this section we briefly describe the refinements performed on RenoDSS from M30 to M33.

7.2.1 Integration of components and internal gains

After the refinement activities RenoDSS takes into consideration the following components and internal gains when generating scenarios related to LCA/LCC, energy, urban planning KPI calculation and renovation measure:

- HVAC components
  - Radiators
  - Boilers
  - VRF Supply components
  - VRF Demand components
  - Thermostats
  - Residential AC units
- Solar collectors
- Water heater/storage
- Photovoltaics
- Space internal gains

The following refinement activities were performed within the RenoDSS workflow in the context of the components and internal gains:

- Extending the BIMERR Material and Component Database, its administration user interface, and REST endpoint to support necessary component properties.

- Extending the RenoDSS data enrichment functionality to support components and internal gains, i.e., the user can now easily add properties which are missing in the IFC file for conducting the KPI calculation.
- Considering relevant component properties at the LCA/LCC and energy KPI calculation.
- Extending the IFC reading and writing service to handle component and internal gain data.
- Extending the user interface of the renovation measure selection to support the components as potential renovation measures.
- Extending the renovation scenario generator to include components in the generated renovation scenarios.
- Adapted the renovation scenario PDF reports and REST endpoint to reflect the components within the renovation scenarios.
- Extended the building visualization within the RenoDSS user interface with regards to component highlighting.

7.3 BIMERR ARIBFA

The development activities of BIMERR ARIBFA were performed as part of WP5 and reported in D5.9 and D5.10, delivered in M23 and M30, respectively. In this section, we briefly describe the refinements performed on ARIBFA from M30 to M33.

7.3.1 Integration of BIF interactions

After the refinement activities, the interactions of ARIBFA with the BIMERR Interoperability Framework (BIF) have been extended. The following refinement activities were performed within the ARIBFA workflow in the context of communicating through BIF:

- The ARIBFA tool was adapted to send annotations to the BIF using the newest version of the annotation data model.
- The ARIBFA tool was extended to send annotations having multiple attached files (i.e. image and video files), after this functionality was supported by the annotation data model in BIF.
- The ARIBFA tool was extended to receive/send the IFC file and the accompanying building data model in JSON format to the BIF, since this capability was finalized in the BIF's API.

7.4 BIMERR PWMA FOR RESIDENTS

The development activities of BIMERR PWMA for residents were performed as part of WP6 and reported in D6.10 and D6.11. In this section, we briefly describe the refinements performed on PWMA for residents from M31 to M33.

After the refinement activities, the interactions of PWMA for residents with the BIMERR Interoperability Framework (BIF) have been extended. Until M30 the application could only upload to the BIF one attachment per reported issue that user submits (using the latest available version of the annotation data model) as BIF was still under development. During these months PWMA for residents managed to send annotations having multiple attached files, after this functionality was supported by the annotation data model in BIF.

8. CONCLUSIONS

This document describes the work carried out within T8.3 during the first 18 months after the kickoff of the task (M15 to M33). 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 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.
- Provided constant support to the pre-validation and demonstration activities. During the support, few of the BIMERR components have gone through refinements.

Further refinement of the BIMERR system shall be carried out until M43 as part of task T8.3. The final deliverable D8.5 will report on a consolidated version of APIs, means of verification for each use case interaction as well as on refinements of each BIMERR tool.

9. BIBLIOGRAPHY

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

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

[3] Leung, H. K., and 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.

[4] BIMERR, "Deliverable D3.1 – Stakeholder requirements for the BIMERR system", 2019.

[5] BIMERR, "Deliverable D3.6 - BIMERR system architecture final version", 2020.

[6] BIMERR, "Deliverable D8.6 – Report on BIMERR pre-validation activities", 2021.

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.

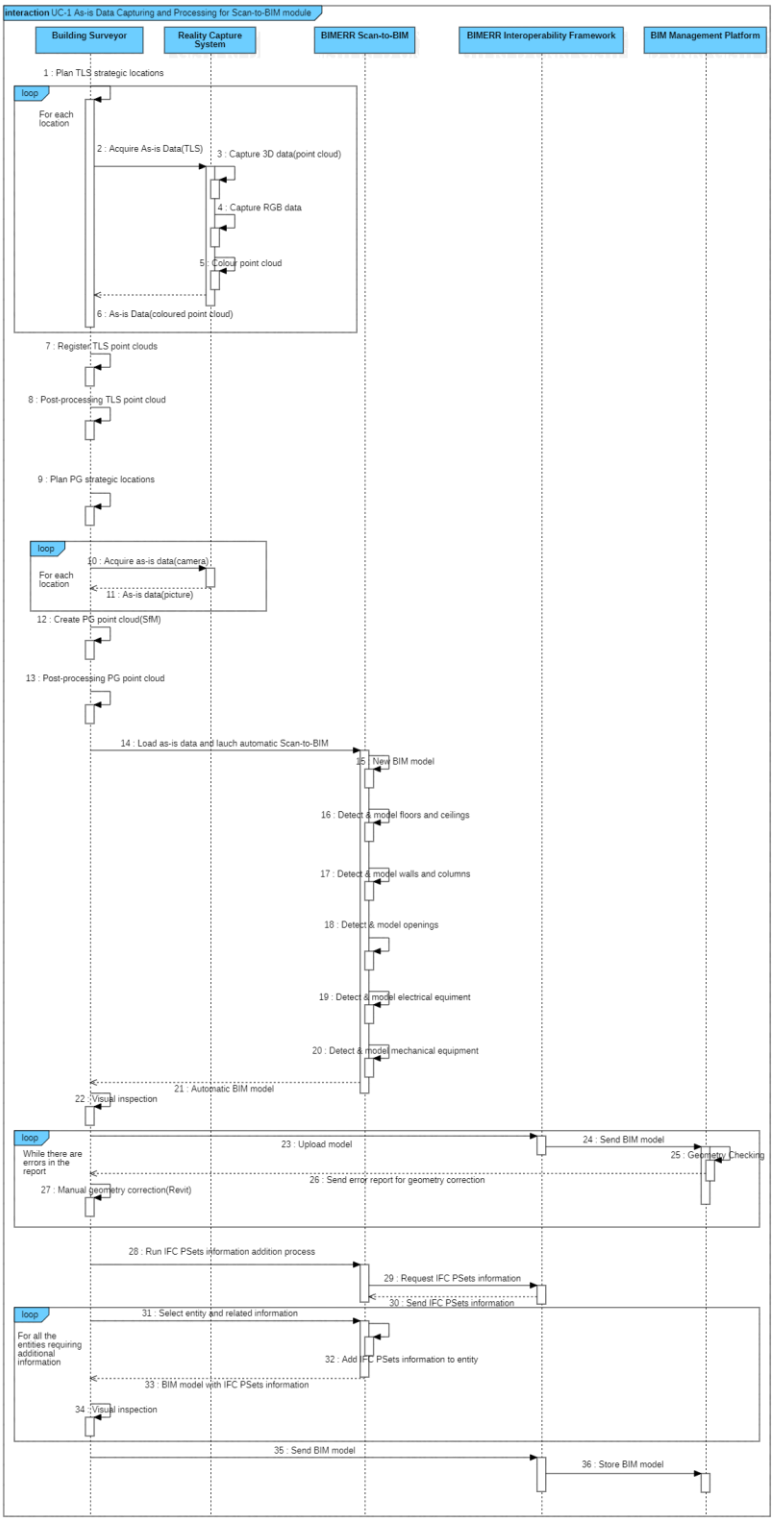


Figure A-0-1 UC–01: As-is Data Capturing and Processing for Scan-to-BIM

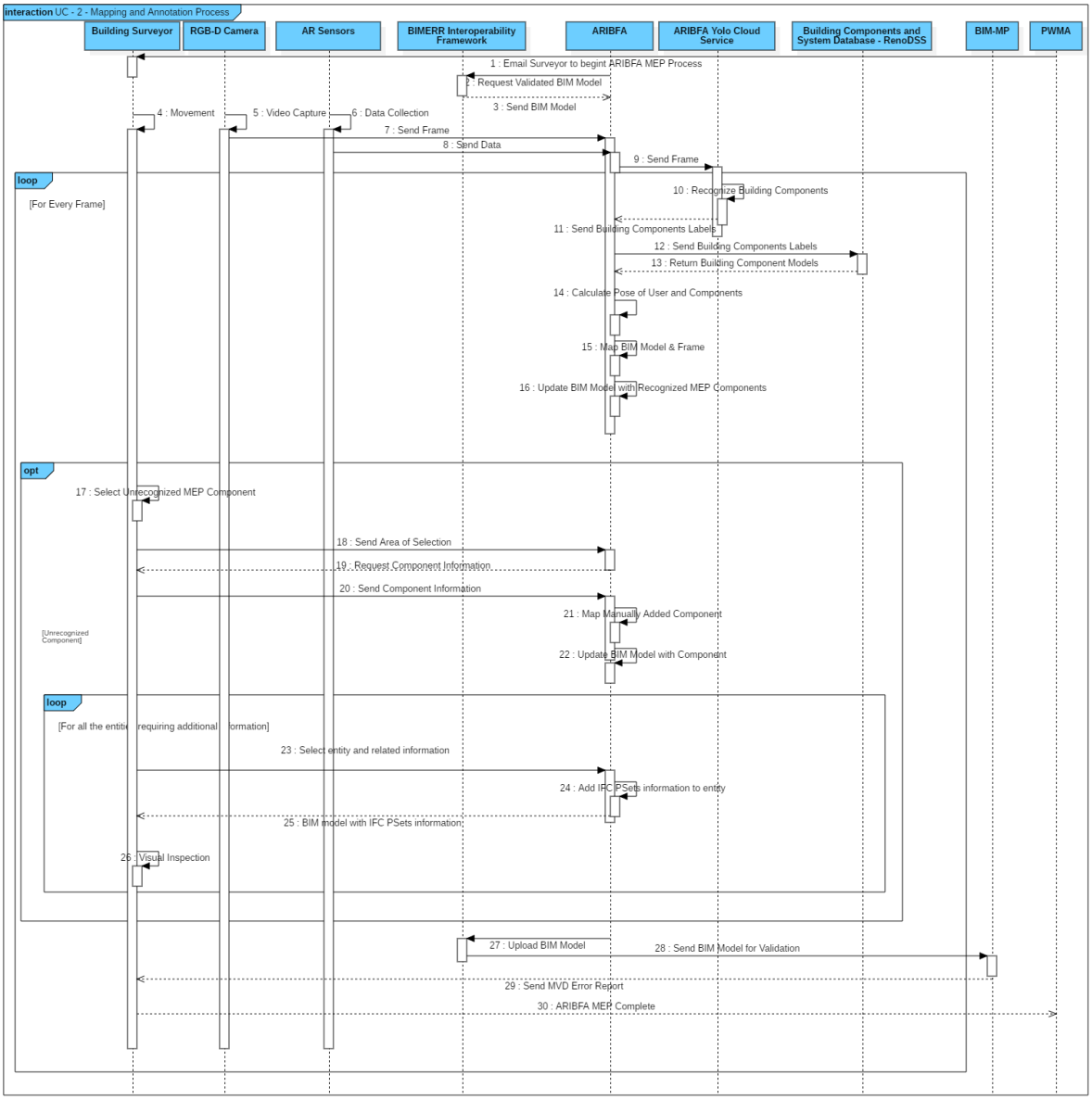


Figure A-0-2 UC-02: Mapping and annotation process

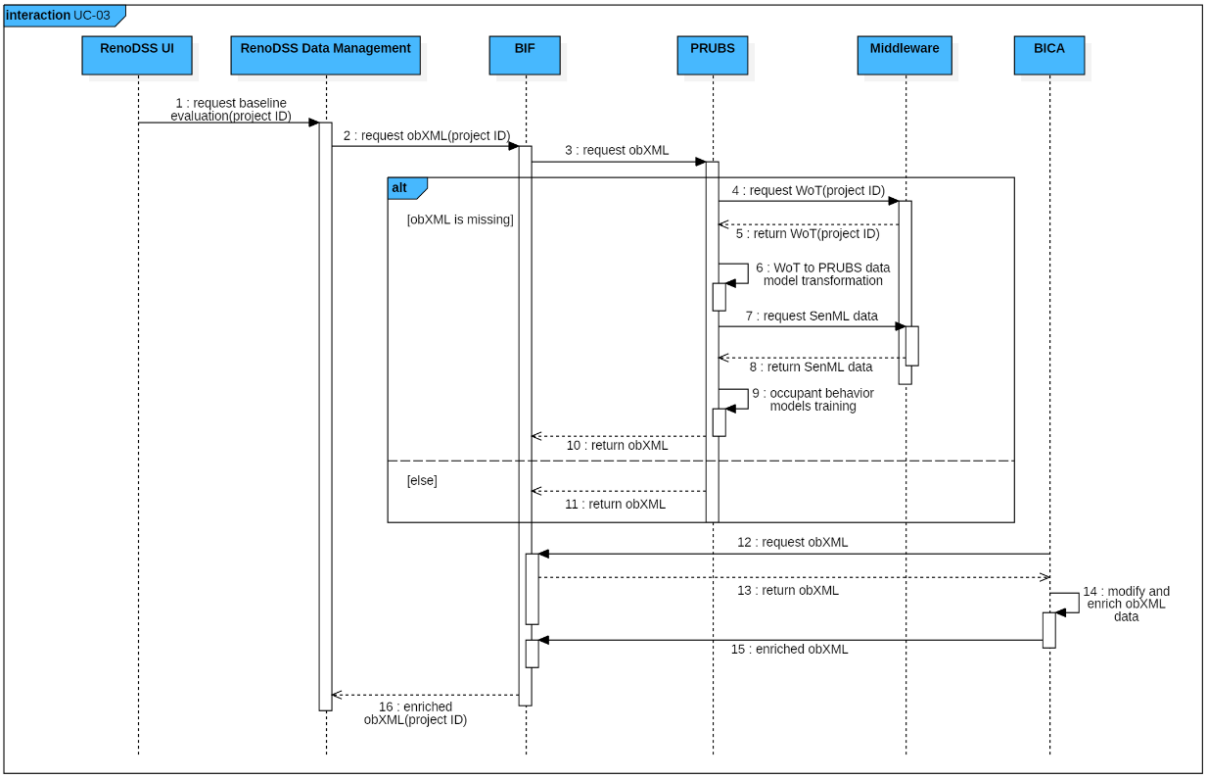


Figure A-0-3 UC-03 Adapt design to the actual building use

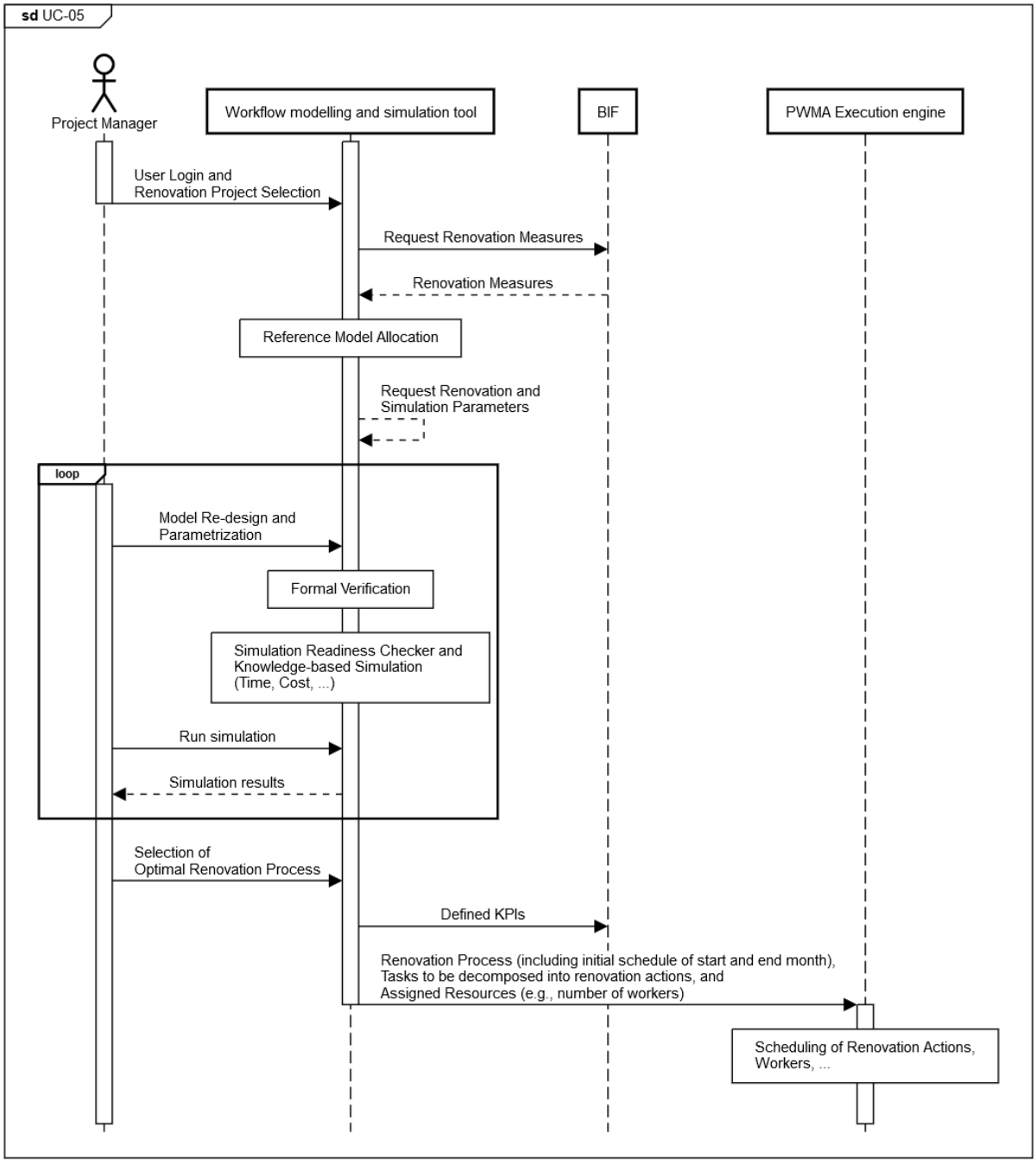


Figure A-0-4 UC-05 Simulation of renovation process and optimal scheduling

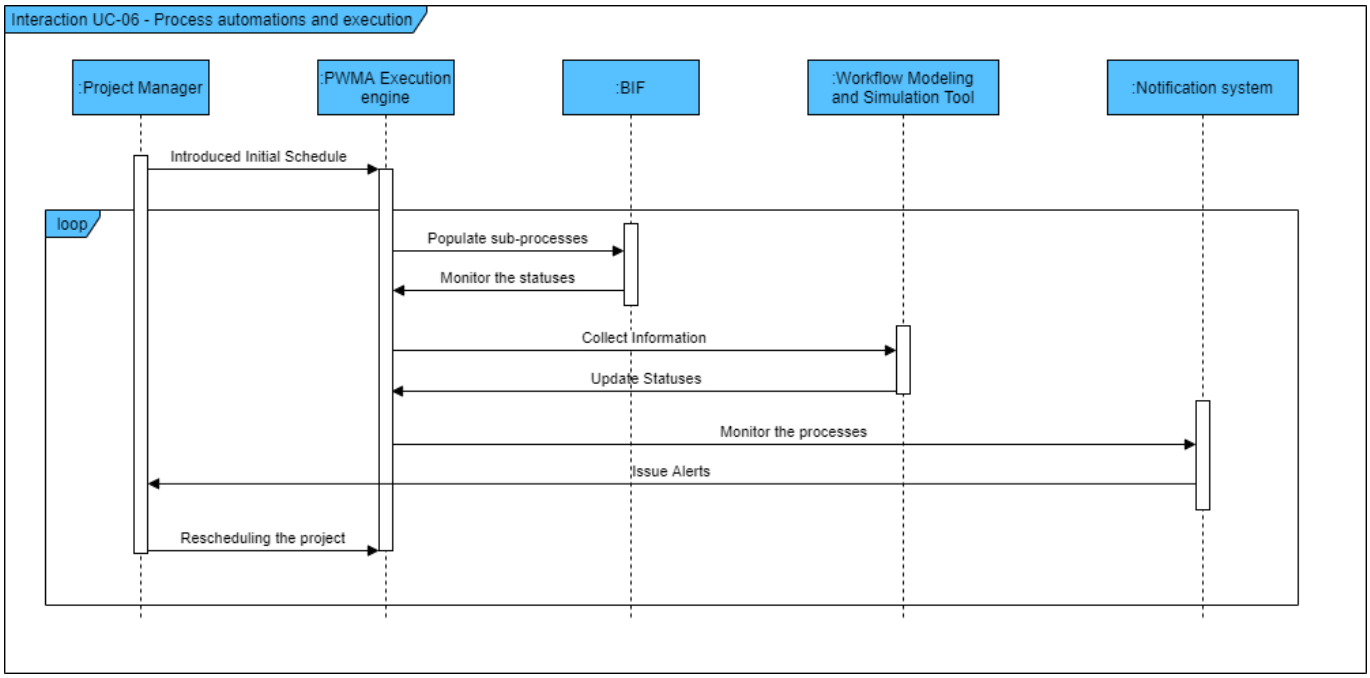


Figure A-0-5 UC-06 Process automation and execution

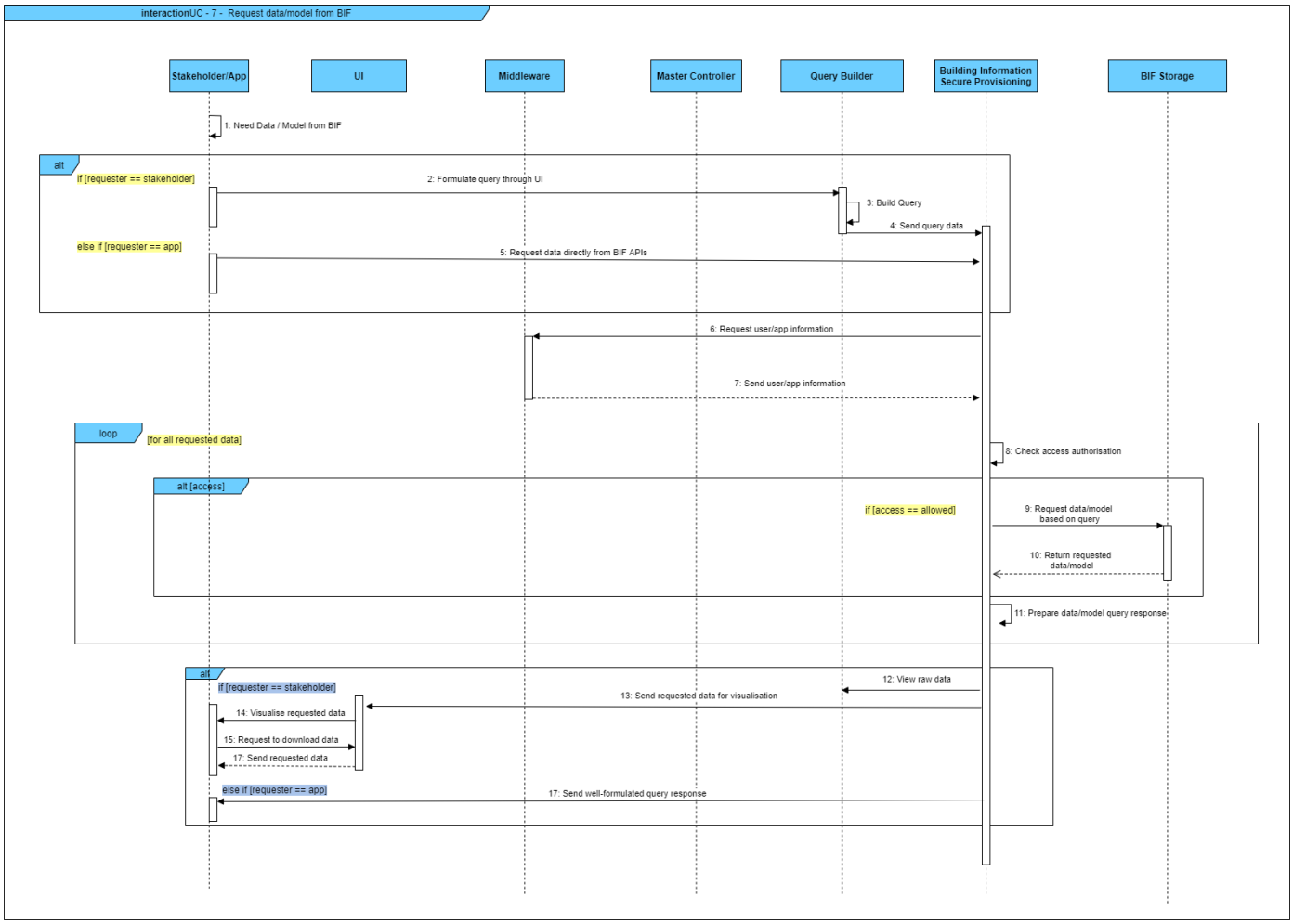


Figure A-0-6 UC-07 Request data model from BIF

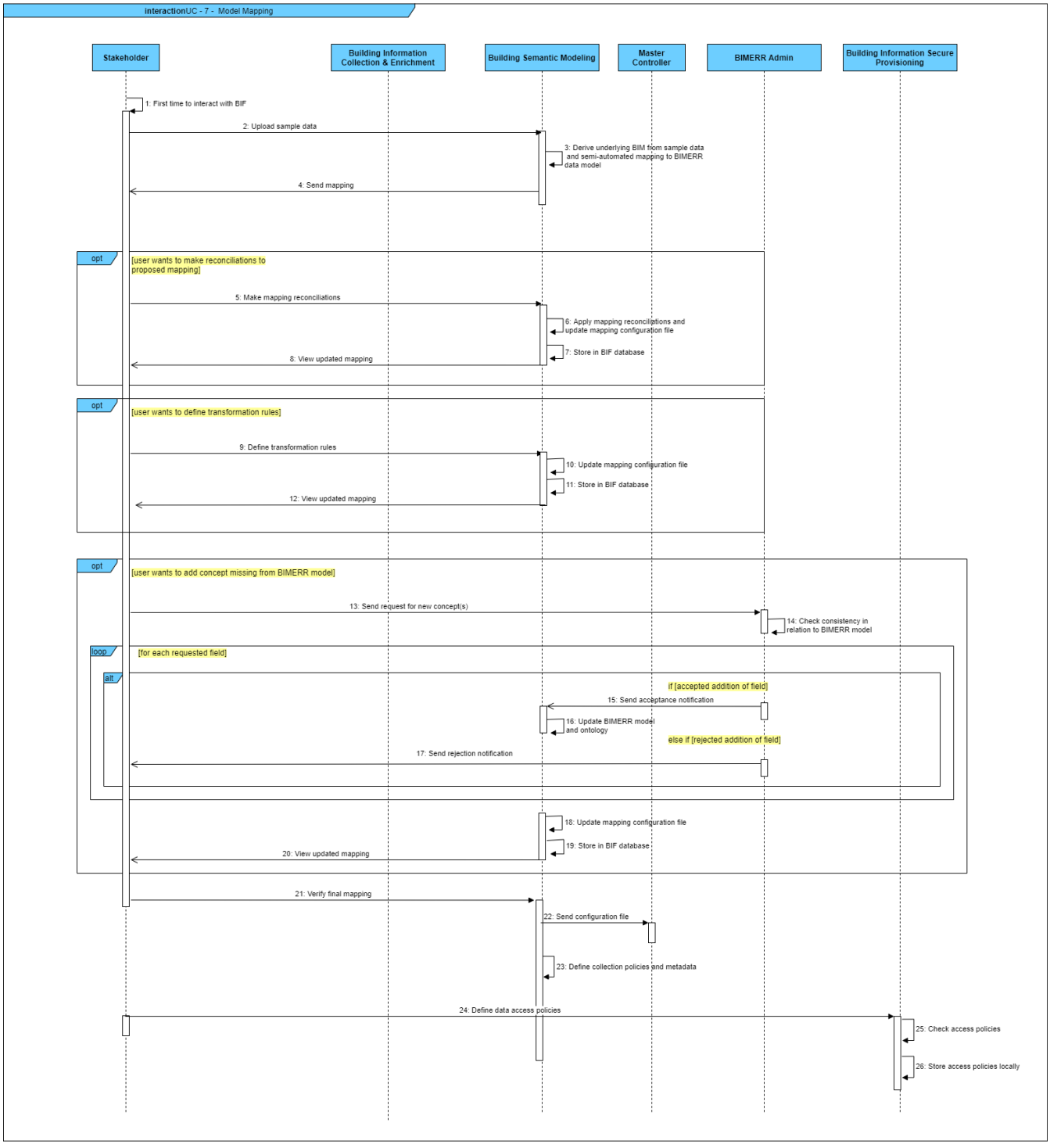


Figure A-0-7 UC-7 Model mapping

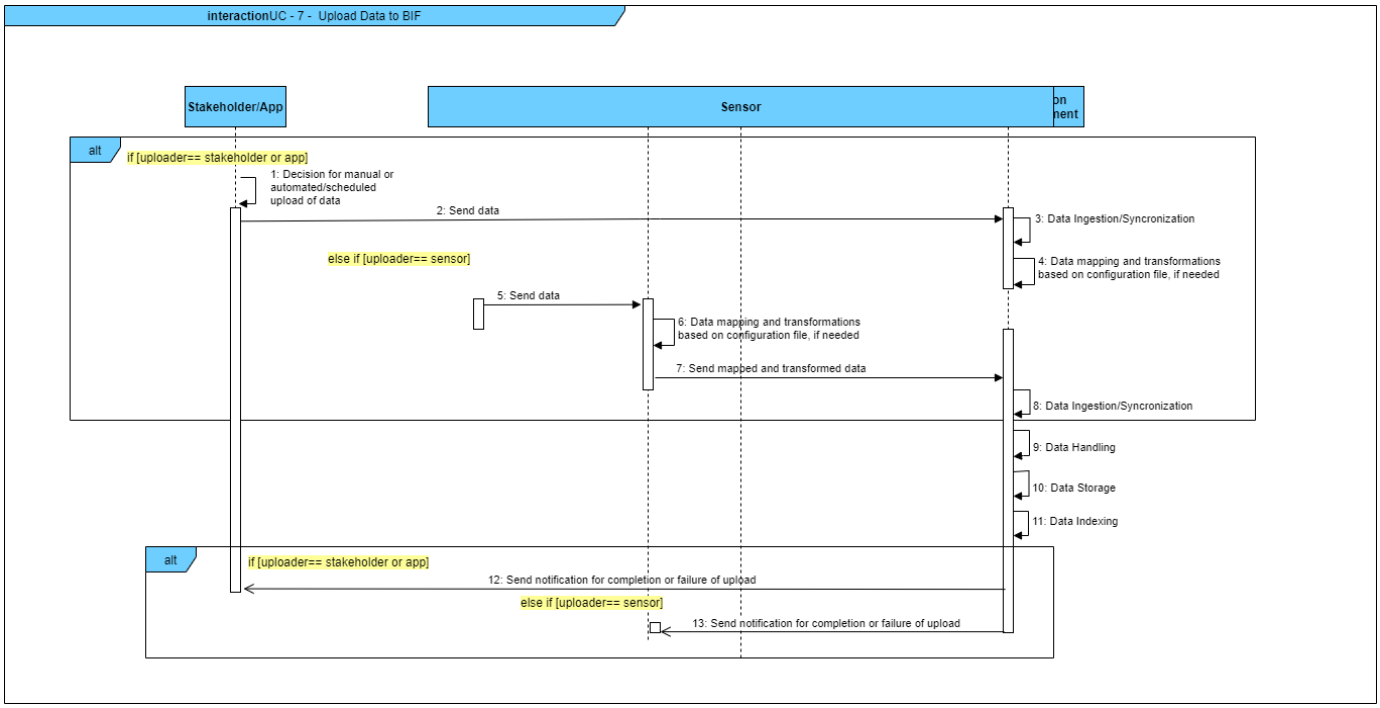


Figure A-0-8 UC-07 Upload data to BIF

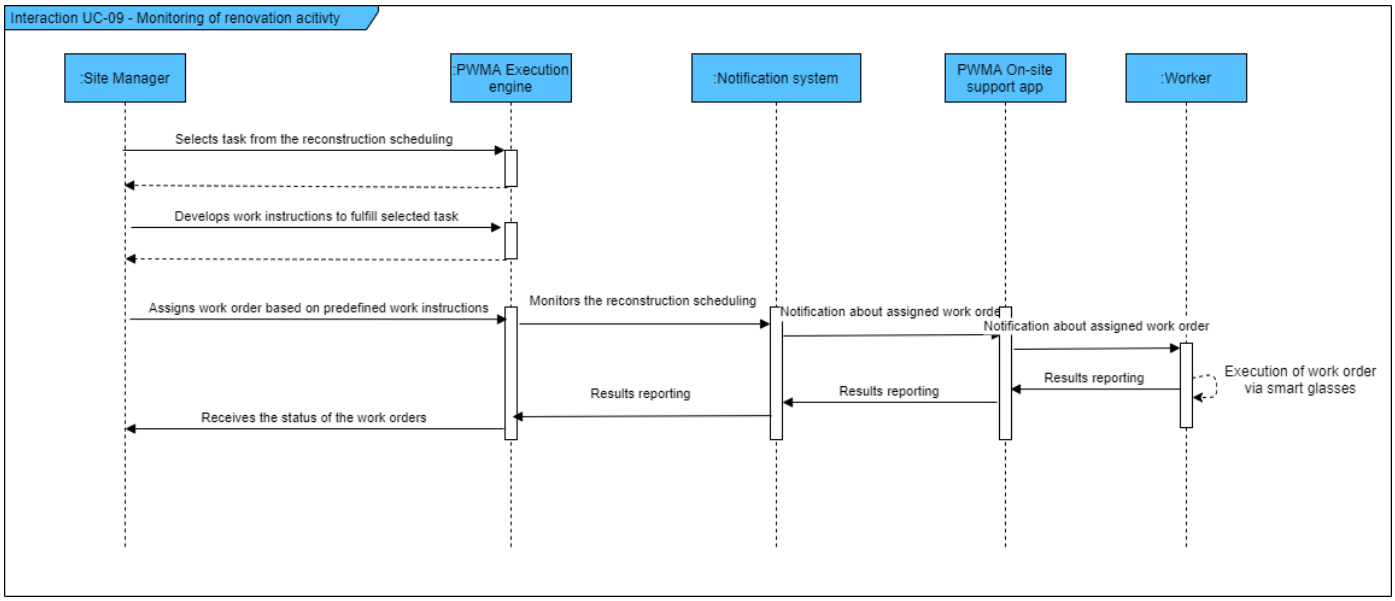


Figure A-0-9 UC-09 Monitoring the renovation activity.

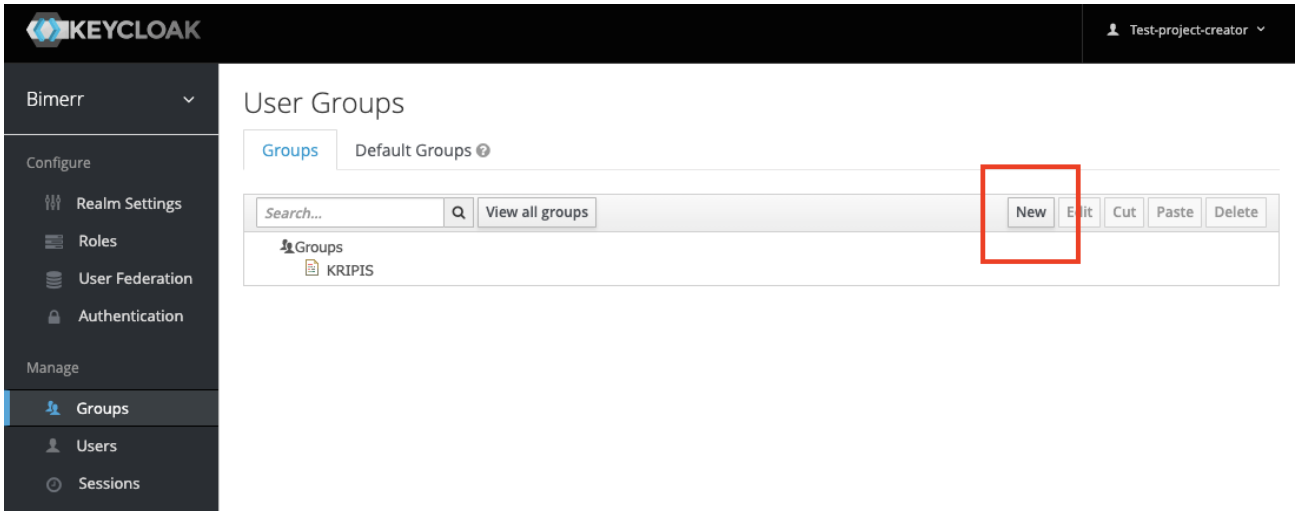
B. Appendix: Project Creation and User Management

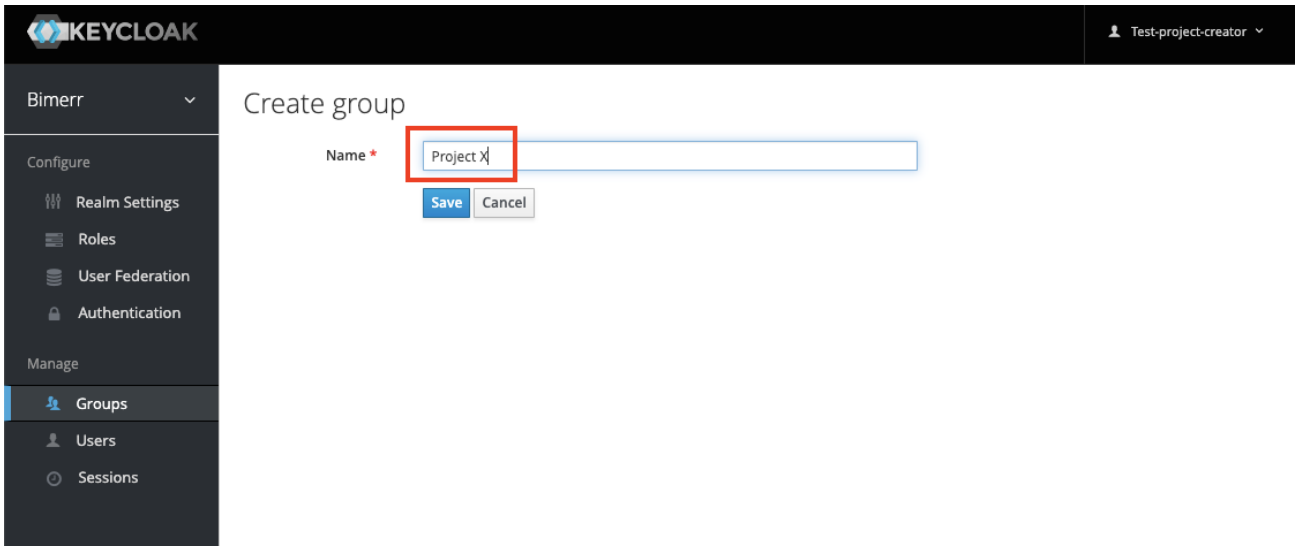
Applications can redirect users to Keycloak's BIMERR Realm **Security Admin Console** for creation of new groups (projects) and assigning groups and roles to existing users. Users can also visit the console via a direct link or from their account page.

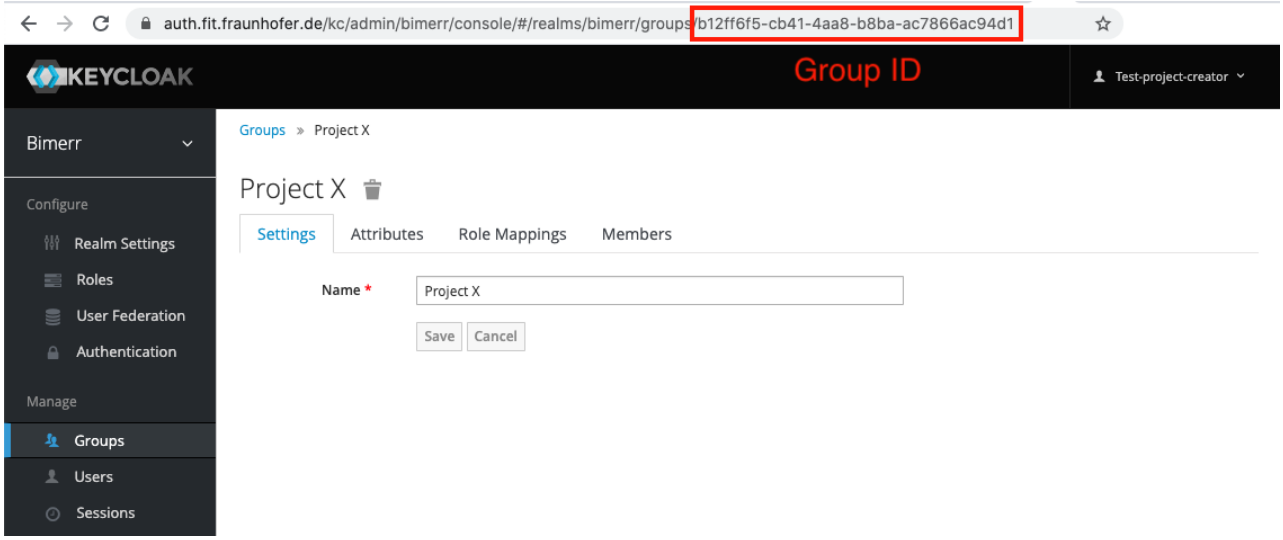
Only users with **BIMERR Identity Manager** role have such access. This role must be given to the user in advance by another BIMERR Identity Manager or a realm admin.

Create a group:

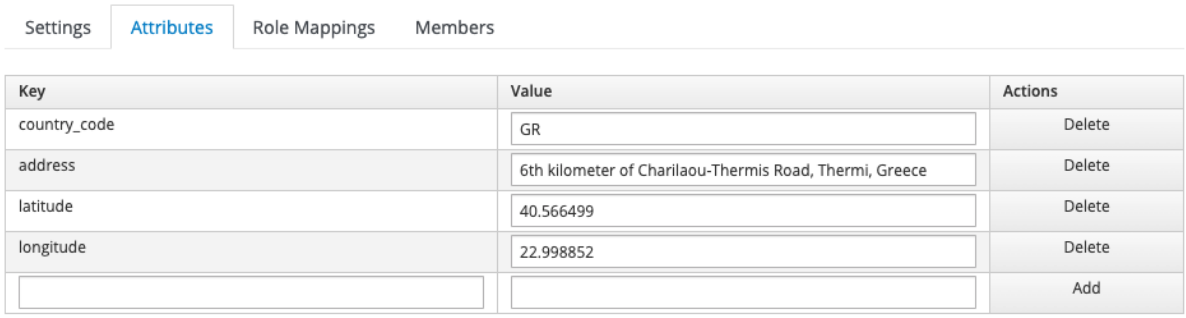
The group ID should be used as the renovation/construction project ID. See Table 5-1







Add metadata attributes:

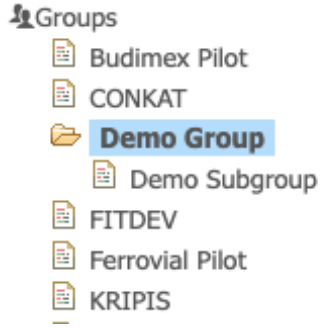


The value type is always array of strings. The delimiter to separate array elements is ##.

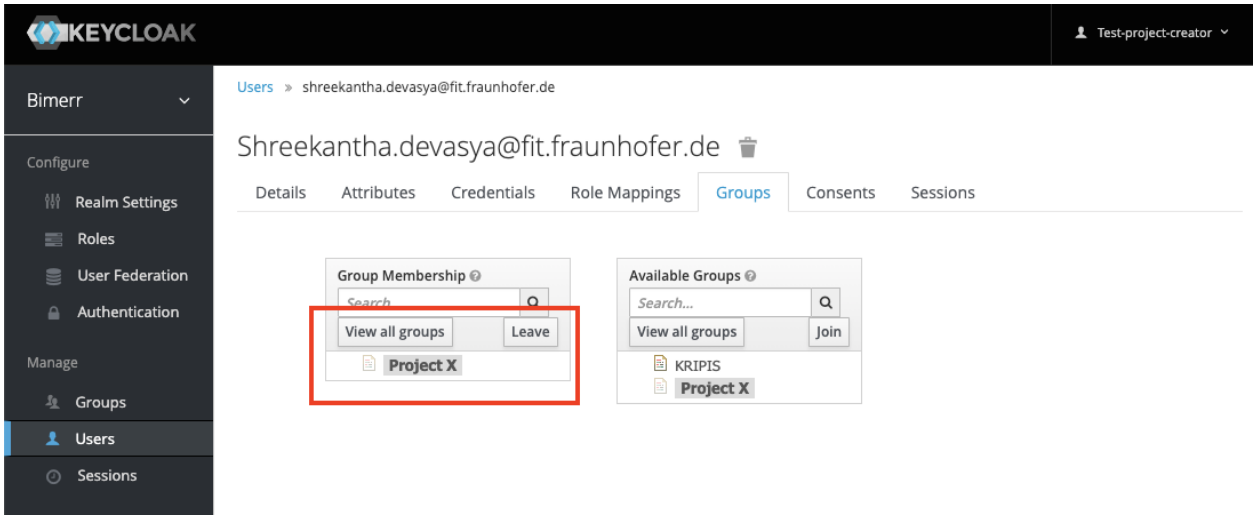
Create a sub-group:

The sub-group ID should be used as the IFC Zone ID.

Select a group and then press the New button. Then follow the same steps as creating a group. The subgroups will appear under the group:

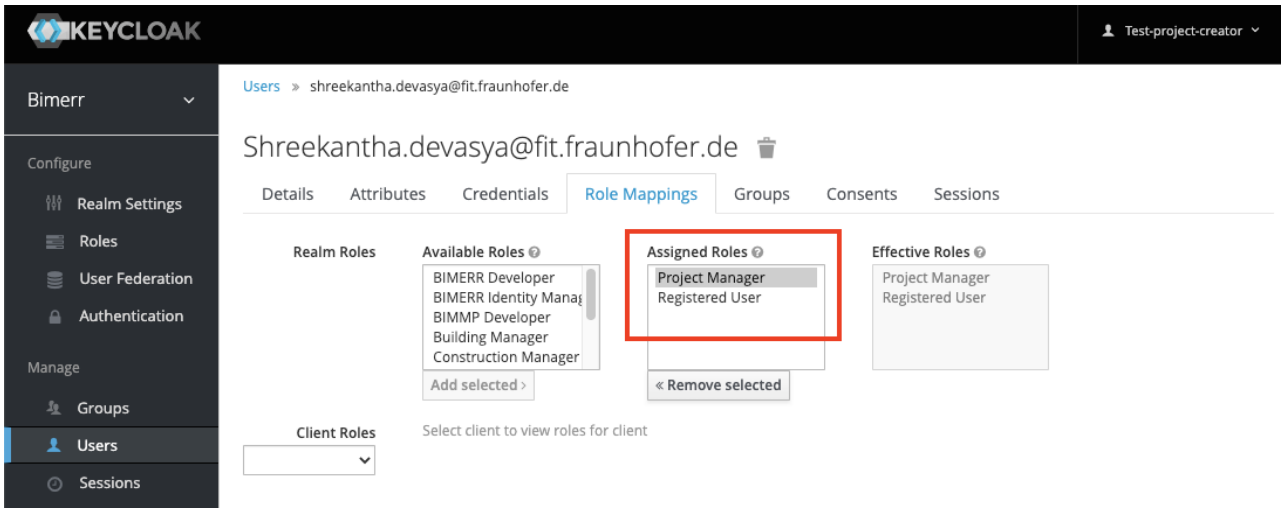


Add user to a group:



Note: The user must be added to a group before adding to subgroups (if any).

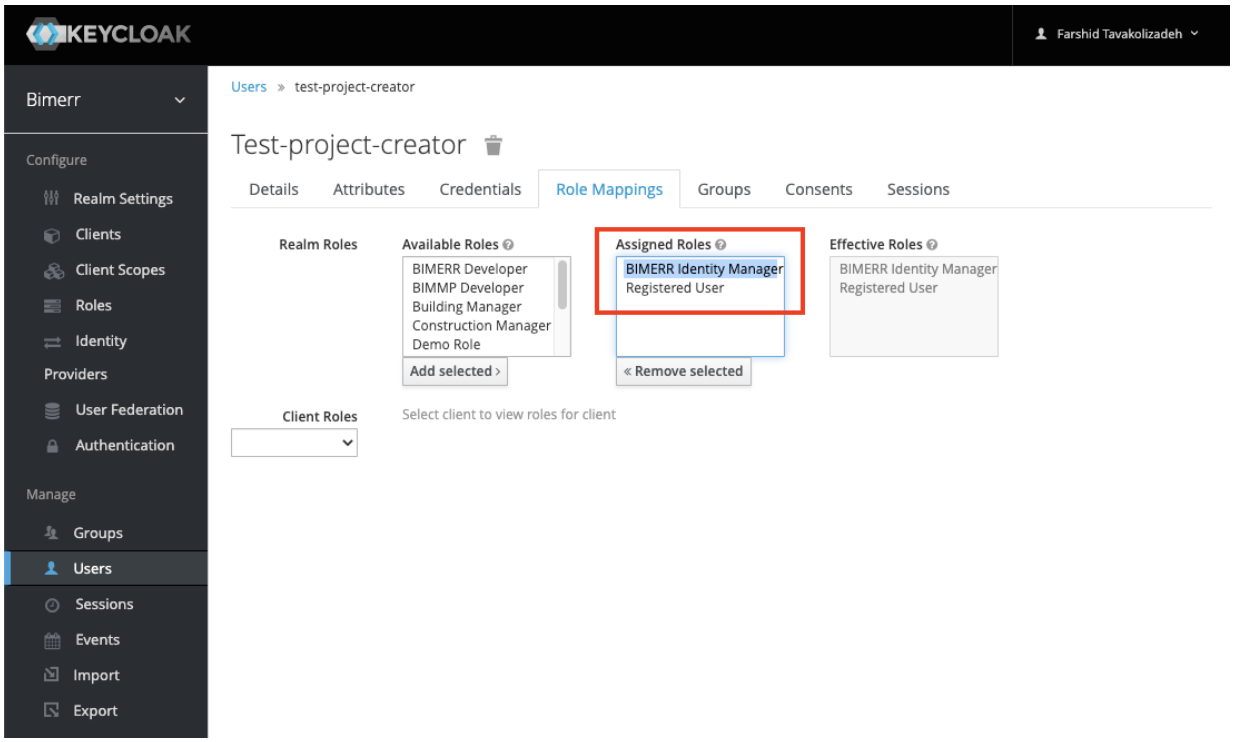
Assign role to a user:



Query programmatically:

To query this information, use the REST API exposed by Keycloak. A subset of useful Keycloak APIs and examples specific to BIMERR are provided in D8.2 Annex C.

In Keycloak, BIMERR Identity Manager has view-realm, manage-users, view-users roles for realm-management client.



The users with BIMERR Identity Manager role have access to all groups and roles. If we really want to isolate different renovation projects within Keycloak's GUI, we need to make a realm for each renovation project. This is probably better in the long run as it scales nicely. However, it adds considerable management complexity in the scope of BIMERR Project as realms have separate settings, clients, users, groups, roles.