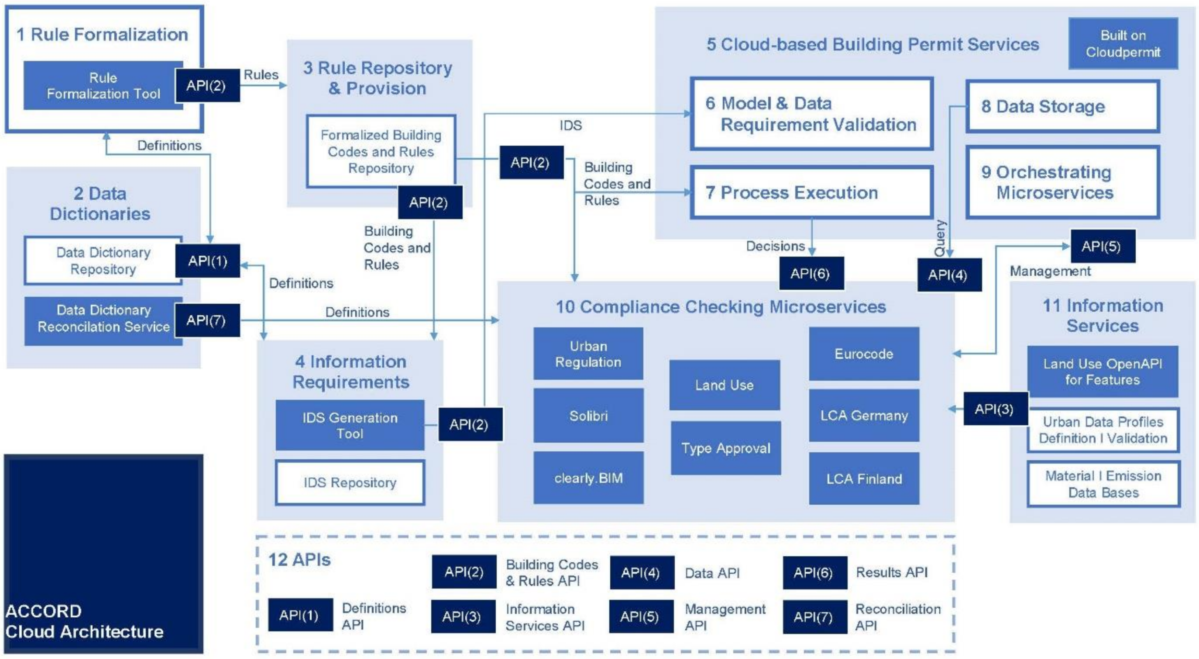
Technical Requirements Elicitation, Analysis and Cloud Architecture Model

February 28, 2024


1 Rule Formalization
Rule Formalization Tool
API(2)
Rules
3 Rule Repository & Provision
Formalized Building Codes and Rules Repository
API(2)
Definitions
2 Data Dictionaries
Data Dictionary Repository
API(1)
Data Dictionary Reconcilation Service
API(7)
Definitions
Definitions
Building Codes and Rules
IDS
API(2)
Building Codes and Rules
5 Cloud-based Building Permit Services
Built on Cloudpermit
6 Model & Data Requirement Validation
7 Process Execution
8 Data Storage
9 Orchestrating Microservices
Decisions
Query
API(6)
API(4)
API(5)
Management
4 Information Requirements
IDS Generation Tool
API(2)
IDS Repository
10 Compliance Checking Microservices
Urban Regulation
Solibri
clearly.BIM
Land Use
Type Approval
Eurocode
LCA Germany
LCA Finland
11 Information Services
Land Use OpenAPI for Features
API(3)
Urban Data Profiles Definition I Validation
Material I Emission Data Bases
ACCORD Cloud Architecture
12 APIs
API(1) Definitions API
API(2) Building Codes & Rules API
API(3) Information Services API
API(4) Data API
API(5) Management API
API(6) Results API
API(7) Reconciliation API

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101056973.

Funded by the European Union

UK Participants in Horizon Europe Project ACCORD are supported by UKRI grant numbers [10040207] (Cardiff University), [10038999] (Birmingham City University) and [10049977] (Building Smart International).

Innovate UK

Funded by the European Union. The views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

Project Title:	ACCORD - Automated Compliance Checks for Construction, Renovation or Demolition Works
Deliverable:	D4.1 Technical Requirements Elicitation, Analysis and Cloud Architecture Model
Type:	Report
Dissemination level:	Public
Work package:	WP4
Deliverable leader:	FhG
Contributing partners:	UNIKO, FhG, BSI, OGC, ONTO, CP, FUI, CU, FUNITEC
Due date:	28 February 2024
Date:	28 February 2024
Status - version, date:	Version 1.2, 28/02/2024
Authors:	Mahmood Al-Doori (UNIKO)
	Katja Breitenfelder (FHG)
	Léon van Berlo (BSI)
	Artur Tomczak (BSI)
	Piotr Zaborowski (OGC)
	Nataliya Keberle (ONTO)
	Ilkka Mattila (CP)
	Rick Makkinga (FUI)
	Thomas Beach (CU)
	Gonçal Costa (FUNITEC)
Reviewer 1:	Jan Jürjens (FHG/UNIKO)
Reviewer 2:	Rita Lavikka (VTT)

DOCUMENT HISTORY

Version	Date	%	Comments	Main Authors (affiliation)
0.1	19/07/2023	4	Initial deliverable structure and Table of Contents.	Mahmood Al-Doori (UNIKO)
0.2	10/09/2023	10	Content to section 1 and 2, detailed structure of section 5 (component outline).	Mahmood Al-Doori (UNIKO)
0.3	16/11/2023	12	Document structure, content to section 2 (methodology for Technical Requirements Elicitation and Analysis) and 3.	Katja Breitenfelder (FHG)
0.4	22/12/2023	18	General contribution from openBIM solutions, content to 5.1 on bSDD.	Léon van Berlo (BSI), Artur Tomczak (BSI)
0.5	29/12/2023	23	Content to sections 5.2, 5.3, 5.9, 5.13.	Nataliya Keberle (ONTO)
0.6	29/12/2023	40	Contribution to various components in section 5.	Piotr Zaborowski (OGC)
0.7	01/02/2024	57	Content to section 4 and 5: Rule formalization approach, API definitions, cloud-based services.	Thomas Beach (BCU), Ilkka Mattila (CP)
0.8	12/02/2024	85	Content to section 2: Final ACCORD Cloud Architecture and component list.	Katja Breitenfelder (FHG)
0.9	22/02/2024	90	Content to section 2, 3 and 5, finalisation; first internal review.	Mahmood Al-Doori (UNIKO), Thomas Beach (CU), Rita Lavikka (VTT), Jan Jürjens (FhG/UNIKO)
1.0	23/02/2024	97	Final content to sections 2, 3, 5 and 6 and Appendices.	Katja Breitenfelder (FhG)
1.1	26/02/2024	100	Final review.	Rita Lavikka (VTT), Jan Jürjens (FhG/UNIKO)
1.2	28/02/2024	100	Final version submission.	Rita Lavikka (VTT)

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

Executive Summary

This deliverable will document the results of the ACCORD project's task *4.1 Technical Requirements Elicitation, Analysis and task 4.2 Cloud Architecture Model*. These tasks belong to Work Package 4 Solutions development.

Task 4.1 will define technical requirements and analyze them based on the ACCORD semantic framework. Moreover, the functional and non-functional requirements will be accordingly detailed, and the resulting requirements will be in line with the user requirements defined in T1.3.

Task 4.2 will design the cloud computing architecture for the storage, processing, analysis, and retrieval of administrative and regulatory information related to construction, renovation and demolition works. This will be done by building on the ACCORD semantic framework incorporating the User Requirements Specification (T1.3) and the technical requirements specification (T4.1). The task will also define the ACCORD demo-specific alignment, integrating the cloud-based architecture, consortium partners' and existing (micro-)services according to the specific needs of the demonstrator projects.

This document presents the ACCORD technical requirement specification as well as the documentation of the ACCORD cloud architecture, identification of responsible partners and identification of needed APIs. This is broken down into the following ACCORD Cloud Architecture components:

1. Rule Formalization Component
2. Data Dictionaries Component
3. The Rule Repository and Provision Component
4. Information Requirements Provision Component
5. The Cloud-based Building Permit Services Component
6. Model- and Data Requirement Validation Component
7. Process Execution Component
8. Data Storage Component
9. Orchestrating Microservices Component
10. The Compliance Checking Microservices Component
11. Information Services Component
12. APIs: The APIs in the ACCORD Cloud Architecture are as follows:
    - API (1) Definitions API
    - API (2) Building Codes and Rules API
    - API (3) Information Services APIs:
    - API (4) Data APIs
    - API (5) Management APIs
    - API (6) Results API
    - API (7) Reconciliation API.

Following the specification of the architecture, the deliverable then documents each component in detail, outlining its structure, behavior, mapping to the requirements and a plan for its implementation. Then, finally, the relationship between the architectural components is formalized, documenting which elements of the ACCORD architecture will be used in each pilot.

Contents

List of Figures

List of Tables

Terminology

Term	Definition
Domain Specific Language	A language that is accommodated to a pre-defined system of understanding or semantics that is only applicable to a certain area or domain.
JSON	The abbreviation for Javascript Object Notation, is a data-exchange format used to store and exchange data objects.
Knowledge Graph or semantic network	Represents a network of real-world entities -objects, events, situations, or concepts -, and their relationships.
Ontology	The definition, categorization, properties assignments and relationship between entities applying to entities in a framework.
OWASP	Short for Open Web Application Security Project is a non-profit project aimed at identifying the most recent software vulnerabilities.
Requirement Category	A detailed definition of the quality metric that a requirement aims to address, e.g. maintainability.
Requirement Existence	The requirement's status as either already applied in a running system/ environment (pre-existing requirement) or not (Novel Requirement).
Requirement Potential Source	The stakeholder involved with the requirement.
Requirement Priority	The degree of importance of a certain requirement.
Requirement Role Specificity	The stakeholder's direct or indirect role in applying the requirement. Defines whether the involved stakeholder in writing the requirement has a
Requirement Type	A clear definition and distinction between the requirements that depict an operational purpose (functional requirements) and those which describe an overall quality aspect (non-functional requirements).
Rule Formalization Process	The ACCORD process of formalising regulations into machine-readable ones is based on the ACCORD building compliance rule language.
Semantics	The process of adding meta -definition, classification, relationships, rules-, information to the properties of a building information model.
Software System	A term used to refer to the intercommunication between hardware and software components, respectively.
System	In the context of this document, it refers to the ACCORD software system.
Technical Requirement (TR)	An informal textual representation of the expected technical capabilities of a software system. A technical requirement can either be functional or non-functional.
TR Analysis	The process by which the requirements are further checked for validity and correctness so that a final list is agreed upon by all the stakeholders in the ACCORD consortium.
TR Collection	The process by which the technical requirements were collected from all the partners in the ACCORD project.
TR Elicitation	The process by which a technical requirement is clearly defined in the context of a system. The result of this process is a list of requirements that feed into the upcoming stage of the development.
YAML	YAML short for Yet Another Language is a human-readable serialization language used to configure applications that deal with data being used for the purpose of transmission or storage.

1 Introduction

1.1 The ACCORD Project

The ACCORD project's objective is to provide a framework for digitalising building permitting and compliance processes using BIM and other data sources, with the end goal of improving the productivity and quality of design and construction processes, supporting the design of climate-neutral buildings and advancing a sustainable built environment in line with the EU Green Deal and New European Bauhaus initiative.

ACCORD is based on the principles that these digitised processes must be human-centred, transparent, and cost-effective for the permit applicants and authorities and, above all, relevant to the industry within which they are to be employed.

To achieve this, ACCORD is developing a Semantic Framework for European digital building permitting processes, regulations, data, and tools. This framework will drive rule formalisation and integration of existing compliance tools as microservices. Solutions and tools are to be developed, providing consistency, interoperability and reliability with national regulatory frameworks, processes, and standards. It will enable the integration of technical solutions for automating compliance checking of buildings in their design, construction, and renovation/demolition lifecycle phases.

To ensure the industry relevance of the project work, the first work package of the ACCORD project analysed the complex landscape of built environment compliance checking and building permitting across Europe to ascertain the requirements for the future digitalisation of this complex interdisciplinary field. The project partners conducted a landscape review and analysis of the current adoption of the concept of digitalisation of building permit- and compliance checking, including a survey into the attitudes of stakeholders to the prospective digitalisation of this domain in a range of European countries. This work was reported in the D1.1 Landscape Review Report that focused on a) academic projects and methods, b) relevant software tools and technologies, and c) national adoption efforts in the field.

This solid basis will pave the way for a framework that has the potential to achieve real change and drive forward the digitalisation of this area. Evidence of this will be collected through the implementation and demonstration of construction projects in various EU regulatory contexts: UK, Finland, Estonia, Germany, and Spain.

1.2 Aims and Objectives

This deliverable documents the results of ACCORD's tasks "4.1 Technical Requirements Elicitation and Analysis" and "T4.2 ACCORD Cloud Architecture Model". Both WP4 tasks aim to facilitate the development of innovative technology solutions for automating building permitting and compliance processes across Europe. The results support meeting the overall premise of WP4 to implement the ACCORD Semantic Framework that will integrate and provide access to building compliance and permitting services, allowing for the storing, processing, analysis and retrieval of administrative and regulatory information related to construction, renovation and demolition works.

First, Task 4.1 provides the technical requirements specification, which will be the basis for the development of the ACCORD implementation along with its user requirements specification specified in D1.2. This specification will be supported by defining the different requirements and usage scenarios and aligning the requirements with the components defined in the ACCORD Framework. Then, Task 4.2 will design the ACCORD Cloud Computing Architecture for the storage, processing, analysis, and retrieval of administrative and regulatory information related to construction, renovation and demolition works. This will be done by building on the ACCORD Semantic Framework incorporating the User Requirements Specification (T1.3) and the Technical

Requirements Specification (T4.1). The task will also define the ACCORD demo-specific alignment, integrating the ACCORD Cloud Architecture, consortium partners' and existing (micro-)services according to the specific needs of the demonstrator projects.

1.3 Deliverable Structure

This deliverable explores the processes and results that were conducted in Task 4.1 and Task 4.2. Thus, this document is divided into two main parts. The first part aims to address Task 4.1 Technical Requirements Elicitation and Analysis and consists of Section 2. The second part addresses task 4.2 Cloud Architecture Model, and is contained within Sections 3, 4, and 5.

More specifically, Section 2 provides an overview of the methodological approach conducted to develop, elicitate and analyse ACCORD Technical Requirements and summarises the results of the elicitation and analysis process. Section 3 outlines the approach to create the ACCORD Cloud Architecture and presents the results. Section 4 provides a detailed view of the project's rule formalization approach. Section 5 offers a technical description of the various ACCORD Cloud Architecture components to be developed in the context of Work Package 4. It also discusses the demo-specific alignment of the ACCORD Cloud Architecture components with ACCORD's various country-level implementations. Finally, Section 7 will conclude the deliverable.

2 Technical Requirements Elicitation and Analysis

2.1 Introduction

This section aims to develop, elicit, and analyse Technical Requirements for the ACCORD Cloud Architecture. Actors, functional- and non-functional requirements and interfaces are defined. The focus of the analysis is to come up with requirements that are compatible and, thus, complete the results of the ACCORD User Requirements Specification (T1.3). The results of the Technical Requirements Elicitation and Analysis will, therefore, be a direct precursor to the software models in sections 3, 4 and 5.

The process of eliciting requirements for computer-based systems denotes the activities of seeking, uncovering, acquiring, and elaborating requirements (Zowghi & Coulin, 2005). The requirements usually come from many sources and have different levels of abstraction. It is crucial to understand these requirements as they emerge and develop in the elicitation process (Bruegge & Dutoit, 2009).

The focus of this document and section is on the technical requirements that are directly bound to the ACCORD system itself. These requirements refer to the functional and non-functional aspects of the system.

The functional requirements describe the services the system can offer, how the different components can relate to each other, what functionalities the users can have, and what the expected constrains of the system are. The non-functional requirements, on the other hand, describe the performance, security- or operational aspects of the system, i.e. what the expected performance bottlenecks are.

This section starts with a summary of the ACCORD Users Requirements Specification (T1.3) in section 2.2 and describes how its results will serve as data-input during the following technical requirements elicitation process. The following section 2.3 outlines the methodological approach for the Technical Requirements Elicitation and Analysis which is based on three main phases and subphases as described in the following sections:

1) Technical Requirements Collection
   a. Technical Requirements Collection Phase 1 (Section 2.3.1)
   b. Technical Requirements Collection Phase 2 (Section 2.3.2)
2) Technical Requirements Elicitation
   a. Technical Requirements Elicitation Phase 1 (Section 2.3.3)
   b. Technical Requirements Elicitation Phase 2 (Section 2.3.4)
3) Technical Requirements Analysis (Section 2.3.5).

In these sections, details on the requirements nature and its various elicitation and analysis processes are thoroughly explained in detail. Finally, section 2.4 presents the results of the Technical Requirements Elicitation and Analysis process.

2.2 Data input - ACCORD Framework User Requirements

Task 1.3 of the ACCORD project was dedicated to the specification of the ACCORD Framework and the elicitation of its linked user requirements. The task results were published as deliverable D1.2 ACCORD Framework and User Requirements Elicitation. This subsection will provide a short summary of the user- and technical requirements collection approach during T1.3 and on the elicitation of relevant information for further use in the technical requirements elicitation and analysis during T4.1 with its results described in section 2.4 of this document.

Section 5 “Demo User Requirements” of deliverable D1.2 ACCORD Framework and User Requirements Specification defined the country-specific to-be building permit processes and new technologies linked to them. Further on, the following user- and technical requirements were defined: (1) User requirements from the perspective of relevant stakeholders involved in the building permit process, and (2) Preliminary technical requirements for relevant components that should be newly and/or further developed in each country (see subsections 5.x.3 of D1.2). Since its delivery, the results of D1.2 have been further elaborated and updated according to demo-specific needs. In the case of updates on a country level, those updates were included and used as final data input.

In the next step, the ACCORD Framework User Requirements specifying high-level requirements of the ACCORD framework were elicited. To do so, the high-level requirements from the user and technical requirements developed by each country have been selected and merged with those requirements collected in the project preparation phase (referring to the ACCORD vision) and with high-level requirements resulting from the landscape survey of 11 European countries, reported in D1.2. Appendix 1. ACCORD Framework User Requirements (D1.2) shows the elicited list of ACCORD Framework User Requirements together with their origin: ACCORD Vision (project proposal); High level (landscape survey), German demo, Spanish demo, UK demo, Finish demo and Estonian demo.

The user requirements in D1.2 are mainly divided into two categories: (1) high-level requirements and (2) demo-specific requirements. The high-level requirements describe the general aspects that need to be met in the ACCORD Framework. The high-level requirements were obtained from the following three main sources: analysis of previous projects, observation of the standards, and a survey. The total number of elicited requirements in this stage is 48 user requirements. The user requirements are, in one way or another, related to the technical requirements, and possible cross-cuts were to be identified during the Technical Requirements Elicitation Phase 2 (see Section 2.3.4) using those requirements as data input.

2.3 Methodological Approach

The methodological approach to eliciting and analysing the technical requirements for the ACCORD Cloud Architecture and its components comprises the following phases and subphases.

Table 1 presents an overview of the phases and subphases of the Technical Requirements Elicitation and Analysis and links them to data sources and results. *Figure 1* illustrates the workflow based on this methodological approach.

The methodological approach for each phase and subphase is described in the following sections.

No.	Technical Requirements Elicitation and Analysis Phases & Subphase	Data sources	Results
1	Technical Requirements Collection		
1.1	TR Collection Phase 1	- ACCORD Framework and stakeholder definitions - Pre-acquired Expert Knowledge: TR Elicitation criteria - Stakeholder Input	TR collected (Phase 1)
1.2	TR Collection Phase 2	- ACCORD Framework - TR Collection - Phase 1 - Stakeholder Input	TR collected (Phase 2)
2	Technical Requirements Elicitation		
2.1	TR Elicitation Phase 1	- Criteria for TR collection - Results of TR Collection - Phase 2	TRs elicitated and assigned to ACCORD Framework components
2.2	TR Elicitation Phase 2	- User Requirements collection criteria - Elicitated ACCORD Users Requirements (D1.2) - ACCORD Framework components - ACCORD Cloud Architecture draft	ACCORD Framework User Requirements elicitated, assigned to ACCORD Cloud Architecture components (Appendix 2)
3	Technical Requirements Analysis		
	TR Analysis Phase	- Results of TR Elicitation Phase 2	Documentation, Tables.

Table 1. Technical Requirements Elicitation and Analysis phases: Data sources and results.

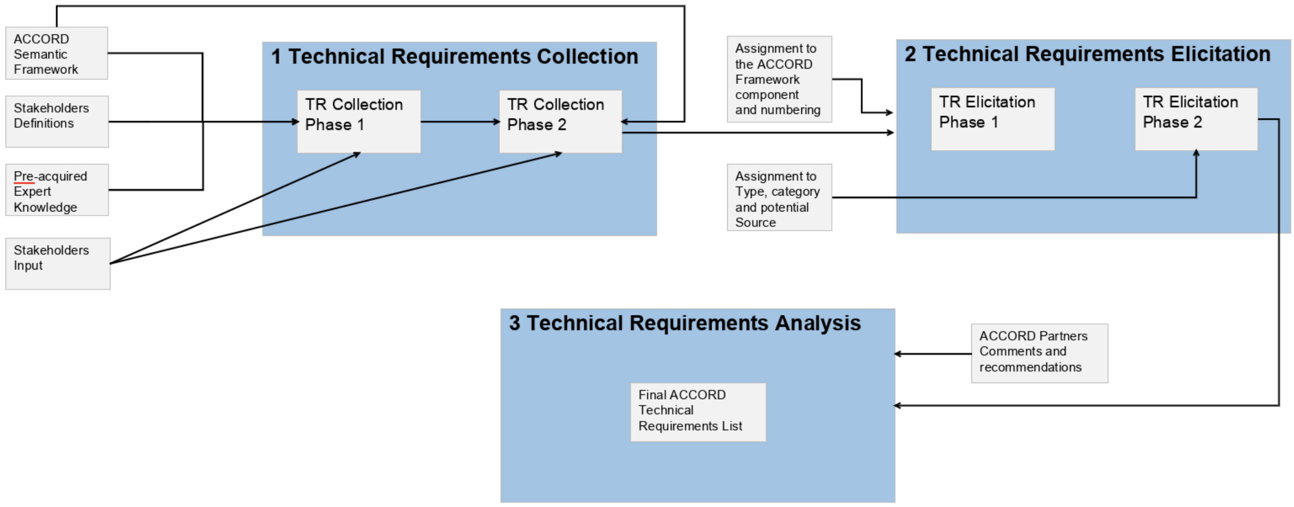

ACCORD Semantic Framework
Stakeholders Definitions
Pre-acquired Expert Knowledge
Stakeholders Input
1 Technical Requirements Collection
TR Collection Phase 1
TR Collection Phase 2
Assignment to the ACCORD Framework component and numbering
Assignment to Type, category and potential Source
2 Technical Requirements Elicitation
TR Elicitation Phase 1
TR Elicitation Phase 2
3 Technical Requirements Analysis
Final ACCORD Technical Requirements List
ACCORD Partners Comments and recommendations

Figure 1. Technical Requirements Elicitation and Analysis: Methodological Approach.

2.3.1 Technical Requirements Collection Phase 1

The first phase of the technical requirement collection follows three major steps: 1) defining technical requirements elicitation criteria, 2) stakeholder input: collecting technical requirements, and 3) examining results, iterations, and redundancy removal.

1. Defining Technical Requirements Elicitation Criteria

A crucial part of defining the technical requirements elicitation criteria was investigating in data sources in accordance with the proposed methodology illustrated in *Figure 1.* Technical Requirements Elicitation and Analysis: Methodological Approach.The selected data sources are:

ACCORD Framework and Stakeholder Definitions: The ACCORD Framework components and the country-specific stakeholders definitions developed in T1.3 served as input for defining the technical requirements elicitation criteria.

Pre-acquired Expert Knowledge: The University's of Koblenz expert knowledge, pre-acquired in various projects, contributed to defining the methods and practices for the elicitation of ACCORD Technical Requirements. The focus of ACCORD's elicitation process is on:

- Defining a coherent scope,
- Avoiding requirements communication misunderstandings,
- Capturing requirements through stages of volatility and change.

Based on this sources, the definition of ACCORD Technical Requirements Elicitation Criteria followed two steps: (1) preselection of technical requirements elicitation criteria in System and Software Engineering based on expert knowledge, and (2) definition of additional actors (later on called "potential source"), that are specific to ACCORD project.

The following **Technical Requirements Elicitation Criteria** are defined for the ACCORD project, a summery of definitions is provided in *Table 3.* Listing of Technical Requirements Elicitation criteria and definitions.

- **Type**: The type of a requirement refers to whether a requirement is functional or non-functional. Technical requirements can encompass both categories. In the context of the ACCORD project, functional requirements define:
  - The interaction between components,
  - The interaction between components and users,
  - The underlying implementation technology,
  - The existing services, API, data formats, or pre-existing services that can be used.

  The non-functional requirements describe tangible aspects that describe a software system, such as performance KPIs, security aspects and technological compatibility.
- **Category**: The category of a requirement defines the focus of a given requirement. It is meant to determine which descriptive type a requirement falls under. Categories are defined in ISO/IEC 25010. The main categories of the requirements elicitation process are: Functional Suitability, Performance Efficiency, Interoperability, Usability, Accessibility, Reliability, Maturity, Security, Maintainability, Portability, Localization, Scalability, and Compliance.
- **Potential Source**: The potential source of a requirement denotes the actor who engages with the requirement. The source can well be the local (building permit) authority, the end-user who engages a building permit, or a software developer who develops a certain service. For the ACCORD project, two additional stakeholders, the "Building Permit User" and the "Local Authority", are added as potential sources to actors in system and software engineering and could be selected during technical requirements collection.
- **Source**: To track the progress of a certain requirement, the source of the requirement needs to be clearly stated. The source is either direct or indirect. A direct source is a stakeholder who writes the requirements without further referral, i.e. a software developer suggests a technical requirement related to software development in the future. An indirect source is a

person writing a requirement for another stakeholder; an example of an indirect requirement is when a software analyst writes a requirement for a software developer.

- **Priority**: The priority of technical requirements is either “high”, “medium” or “low”. High-priority requirements are requirements that must be implemented, medium priority requirements are recommended to be developed and low-priority requirements are requirements which provide a certain additive value to the system, but their fullfillment is not crucial.
- **Existence**: The requirements are either “existing” or “novel”. Existing requirements denote the features that already exist through current tools and/or other systems in the domain of building permit compliance checking. Novel requirements represent the potential features that are going to be developed specifically in the realm of the ACCORD system.
- **Requirements Specificity:** The requirement specificity distinguishes between “overall-“ and “country-specific requirements”. The overall requirements are general requirements that apply to the whole ACCORD system. These requirements contain functional aspects that are universally applicable to ACCORD demo-countries. An example for a country-specific requirement is the compliance with national legislation.
- **Related Task(s):** The related task(s) provide input to which task of ACCORD project the solution development is assigned. The main goal of creating this entry is to make sure that the technical requirements are matched with the information provided in the ACCORD Proposal.
- **Description**: The description is the body of the technical requirement itself.
- **Rationale**: The rationale defines the reason why a certain technical requirement is chosen and provides an extra level of explanation on its selection.

In addition, the following criteria were added for collecting technical requirements:

- Comments: Additional comments and links to documentation,
- Responsible Partner / Person.

2. Stakeholder Input: Collecting Technical Requirements

Based on the defined criteria, a technical requirement collection template for phase 1 and the according task assignment were prepared. The technical requirements collection and elicitation approach and the defined criteria were presented to WP4 partners. Individual questions like “What is the difference between user- and technical requirements?” or “What is the required granularity level of the “Description” and “Rationale”?, were answered by one-to-one communication or during regular WP4 meetings. Finally, WP4 partners were invited to provide technical requirements for solutions to-be developed using the Technical Requirements Collection template for phase 1.

3. Examining Results, Iterations and Redundancy Removal

The third step is dedicated to examine results, iterations and redundancy removal of technical requirements collected in phase 1. Examination criteria and linked actions are as follows:

1. **Completeness:** Negative assessment result either in terms of not providing enough written text or unclear meaning for the criteria “Description” and “Rationale”. The performed action can be to demand missing information from contributing partners, when necessary.
2. **Format:** Negative assessment result for a writing style of the technical requirements “Description” and “Rationale” not being compliant to the equivalent format for technical requirements. The expected action to be performed would be, for instance, to demand changes by contributing partners, when necessary.
3. **Redundancy:** Assessing the possible redundancy of technical requirements. Negative assessment results for technical requirements being similar or equal to others collected. The performed actions were to remove requirements being similar or equal or to merge such requirements with other ones collected.

TR Eliciation Criteria	Defintion
Type	**Functional Requirement** or **Non-Functional Requirement**.
Category	Defines the category of requirement. As a reference, we have included the most relevant categories based on expert knowledge and the categories of ISO 25010. ISO 25010: is a standard jointly coined by the International Organization for Standardization and (the International Electrotechnical Commission) aimed at defining a concrete set of metrics to improve the quality of software systems development. The following categories are based upon it. **Functional Suitability:** denotes the degree to which the system can provide the prescribed functions and features. **Performance Efficiency:** denotes how capable the system can be in doing a certain task while maintaining a minimum usage of the underlying resources. **Interoperability:** denotes the degree to which the system or a component of a system can operate with other components or systems without presenting a serious risk of remodelling or reinventing certain parts of the system. **Usability:** defines how easy to learn and convenient to use should the system be to new or seasoned users. **Accessibility:** defines how easy to use the system is to its expected or targeted users. **Reliability:** defines how dependable the system is in different functioning conditions, e.g. when there is a high data through-put or access demand. **Maturity:** defines how well-developed a system function really is. **Security:** defines the **Maintainability**: determines how easy it is to add or subtract features while the system is running. **Portability:** defines the system's ability to be flexibly adapted to other environments such as new hardware or operating systems. **Localization:** defines the system's ability to be adapted to different markets and user cultures. **Scalability:** defines the system's ability to accommodate increasing numbers of users, thus handling workload variations with neglectable setbacks. **Compliance:** defines a system's compatibility and congruence with standards and best practices. **Others:** No category applicable.
Potential Source	Defines the stakeholder involved in or affected by the requirement. The suggested stakeholders are as follows: **None:** No stakeholder defined. **Internal Technical Analyst:** A software engineer or developer who takes part in the technical development of the system. **Internal Security Analyst:** A software engineer or developer who takes part in the technical development of the system that mainly focuses on security. **End-user:** Any user who uses the system. **System Customer:** A person who uses the system from the outside, e.g. a permitting applicant. **Local Authority:** The stakeholder who is part of the local permitting authority responsible for providing the permits to the applicant. **Building Permit User:** A user who is part of the building authority. **Software Developer:** A stakeholder whose task is the development of components. **Software Penetration Tester:** A stakeholder whose task is the security testing and safeguarding of components.

Table 2. Listing of Technical Requirements Elicitation criteria and definitions.

TR Eliciation Criteria	Defintion
Potential Source	**Software Architect:** A stakeholder who focuses on the developing the architecture of the system. **Software Compliance Engineer:** A stakeholder who focuses on making sure that the system complies with regulations and standards, i.e. security standards, or legal aspects such as ISO compliance. **Software Tester:** A stakeholder who mainly focuses on testing the system as a whole or different components thereof. **Other:** A stakeholder not being specified in the previous list.
Source Specificity	Defines whether the source of a requirement is direct or indirect. **Direct:** The person providing the requirement is a direct Potential Source, i.e. an analyst providing a requirement of an analyst. **Indirect:** The person providing the requirement is an indirect Potential Source, i.e. an analyst filling a requirement for a software developer because he/she knows that it may be relevant.
Priority	Defines the priority of the requirements being either high, medium or low. **High Priority:** A requirement that MUST be included and implemented in the system either from an overall- or a country-specific perspective. **Medium Priority:** A requirement that SHOULD add extra functionalities to a system but shouldn't be treated as important as the latter category. **Low Priority:** A requirement that COULD prove important but should be treated with the same level of urgency as the latter two.
Existence	Defines whether the requirement is already existing or novel. **Existing:** fully developed, mature and running. **Novel:** not developed yet, and it needs to be included in the ACCORD solution.
Requirements Specificity	Distinguishes between overall - and country-specific requirements. **Overall Requirement:** a requirement relevant to all the developments of all the countries in the consortium. **Country-specific Requirement:** is a requirement that is specific to one or more countries i.e. Certain Privacy Law or Legislation(s).
Related Task(s)	The respective relevant task as per the proposal is specified in this entry.
Description	The description provides the details regarding a certain requirement i.e. for a security requirement, the description can be the following: "The system must provide secure data links between itself and other systems".
Rationale	The reason why a requirement is relevant within the context of the system. For the same example, the rationale can be "In order to make sure that the data sharing amongst the many stakeholders is secure and the risk is always reduced".
Comments and Links	Comments or references to the documentation, if any.
Responsible Partner/ Person	The name of the Partner (Organization), the name of the person writing the requirement and the respective reachable email should be included. In case of an indirect source, the same data for both of the person entering the data and the person who is in charge respectively are included.

Table 3. Listing of Technical Requirements Elicitation criteria and definitions.

2.3.2 Technical Requirements Collection Phase 2

The second phase of the technical requirements collection has three major steps: 1) Technical requirements to ACCORD Framework components assignment, 2) a second run of collecting the technical requirements, and 3) examining results, iterations, and feedback loop. The main data sources used in this phase are: (1) requirements collected in the previous phase, and (2) further input from the stakeholders.

In the first step of phase 2, the technical requirements were restructured and assigned to the components of the ACCORD Framework, as outlined in *Figure 1*. Based on this, the technical requirements collection table for phase 2 was developed. During the second step, WP4 parterns were invited to provide new information, feedback, and possible additional technical requirements. The final and third step of the technical requirements collection phase 2 was the examination of results based on the following criteria and linked actions:

1. **Completeness**: Negative assessment result either in terms of not providing enough written text or unclear meaning for the criteria “Description” and “Rationale”. The performed action was to demand missing information from contributing partners, if necessary.
2. **Format:** Negative assessment result for a writing style of the technical requirements “Description” and “Rationale” not being compliant to the equivalent format for technical requirements. The performed action was to demand changes by contributing partners, if necessary.
3. **Feedback:** is set to address open questions when they arise, with the main goal of supporting partners by providing relevant feedback to their contributions.

2.3.3 Technical Requirements Elicitation Phase 1

During the Technical Requirements Elicitation Phase 1, the collected requirements are reorganised and sorted according to the steps illustrated in *Table 4*: (1) Elicitation of ACCORD Framework components by (a) assigning requirements to one or multiple ACCORD Framework components, (b) sorting by elicitation criteria following the order of definitions according to *Table 3*, (c) specification of description and rationale (if necessary), and (d) documentation of changes to original contributions by ACCORD partners.

No.	Step	Description
1	Elicitation of ACCORD Framework components	Based on the preliminary assignment of requirements to ACCORD Framework components (results of collection phase 2).
a	Assignment to ACCORD Framework components and numbering	Each requirement is assigned to one or multiple ACCORD Framework components or defined as generic requirement.
b	Sorting by elicitation criteria and definitions	Each requirement is sorted by category, potential source, source specificity, priority, existence and requirements specificity.
c	Specification of description and rationale	The description and rationale of requirements are specified, if necessary.
d	Documentation of changes	All changes to original contributions by ACCORD partners are documented in a “Change Log”.

Table 4. Technical Requirements Elicitation Phase 1: Organizational steps.

2.3.4 Technical Requirements Elicitation Phase 2

The Technical Requirements Elicitation Phase 2 is based on the results of of the first elicitation phase and structured in the following steps (see *Table 5*): (1) elicitation of ACCORD User Requirements (D1.2, see **Appendix 1**) by (a) mapping to ACCORD Framework components and numbering, (b) supplement of information on elicitation criteria, (c) specification of description, and (d) alignment with country- and use case-specific requirements; (2) elicitation of ACCORD Cloud Architecture requirements by (a) assigning requirements to one or multiple ACCORD Cloud Architecture components or defining as generic requirement based on the draft of the cloud architecture under development - this step includes the definition of new components; (b) final sorting of the technical requirements by ACCORD Cloud Architecture component number(s) and elicitation criteria following the order illustrated in *Table 3*, and (c) documentation of changes. In addition, short descriptions for technical requirements are generated, when necessary, taking description and rationale as a basis.

No.	Step	Description
2.1	Elicitation of ACCORD Users Requirements (D1.2)	Technical Requirements are selected.
a	Assignement to ACCORD Framework components and numbering.	Each requirement is mapped to one or more ACCORD Framework component.
b	Supplement of information on elicitation criteria.	The elicication information for each requirement is supplemented.
c	Specification of description (and rationale)	The description for each technical requirement is modified in accordance with the elicitation criteria definition.
d	Alignment with country- and use case-specific requrirements.	ACCORD Users Requirements defined as visionary- or high-level- and country-specific requirement are elicitated as “overall requirement” and additionally aligned to demo-countries.
2.2	Elicitation of ACCORD Cloud Architecture Requirements	
a	Assignement to ACCORD Cloud Architecture components and numbering	Requirement are assigned to one or multiple ACCORD Framework components or defined as generic requirement. New components are established, if necessary.
b	Final sorting by ACCORD Cloud Architecture component numbers and elicitation criteria/ definitions.	Each requirement is sorted by ACCORD Cloud Architecture component number(s), and by category, potential source, source specificity, priority, existence and requirements specificity.
c	Documentation of changes	All changes to original contributions are documented in a “Change Log”.

Table 5. Technical Requirements Elicitation Phase 2: Organizational steps.

2.3.5 Technical Requirements Analysis

The final list of technical requirements to ACCORD Cloud Architecture components serves as the main source for the following analysis of technical requirements: (1) analysis on technical requirements elicitation phases, (2) analysis on technical requirements elicitation criteria, and (3) analysis on the ACCORD Cloud Architecture including the proven assignement of all technical requirements to at least one ACCORD Cloud Architecture component. The results of the analysis will be presented in the following section 2.4.

2.4 Results

This section describes the results of the Technical Requirements Elicitation and Analysis, namely the comprehensive list of technical requirements to ACCORD Cloud Architecture components, the analysis results on technical requirements elicitation phases and -criteria, and finally the analysis results on the ACCORD Cloud Architecture and its components.

2.4.1 List of Technical Requirements to ACCORD Cloud Architecture Components

Resulting from the Technical Requirements Elication Phases 1 and 2 is a comprehensive "**Technical Requirements to ACCORD Cloud Architecture Components"** list presented in **Appendix 2.** The list made used of the technical requirements collected upon ACCORD partners in T4.1 and included the ACCORD Framework User Requirements (D1.2) in the elicitation process.

2.4.2 Analysis Results on Technical Requirements Elicitation Phases

The results of the Technical Requirements Analysis with regard to the Technical Requirements Elicitation phases are:

Technical Requirements Elicitation Phase 1

- A total number of 93 technical requirements were elicited from technical requirements collected during collection phases 1 and 2.
- A total number of 5 technical requirements resulting from merging two requirements.
- Information on the elicitation criteria of technical requirements was corrected and the description and rationale specified, if necessary. All changes were documented.

Technical Requirements Elicitation Phase 2.1

- A total number of 47 out of 48 user requirements were elicited as technical requirements.
- A total number of 41 technical requirements were elicited from the ACCORD Users Requirements (D1.2) and added to the ACCORD Techncial Requirements list.
- Information on elicitation criteria was provided for all new technical requirements, and descriptions were specified. All changes were documented.
- The complete ACCORD Technical Requirements list comprises a total number of 134 technical requirements sorted by information on elicitation criteria.

Technical Requirements Elicitation Phase 2.2

- A total number of 134 techncial requirements were elicitated and assigned to one or multiple ACCORD Cloud Architecture components or defined as generic requirement.
- A total number of 7 technical requirements elicited from ACCORD Users Requirements and assigned to ACCORD Framework components could not be assigned to ACCORD Cloud Architecture components.
- New components for the ACCORD Cloud Archiecture were defined (see section 2.4.4), the process being linked to the components refinement which is described in section 0.
- All changes were documented.

2.4.3 Analysis Results on Technical Requirements Elicitation Criteria

The results of the Technical Requirements Snalysis with regard to the Technical Requirements Elicitation criteria are:

- **Type**
  - A total number of 20 non-functional- and 114 functional elicited technical requirements.
- **Category:** Technical requirements elicited for the following categories:
  - Functional Suitability
  - Interoperability
  - Suitability
  - Security
  - Compliance
  - Localization
- **Potential Source:** Technical requirements elicited for the following potential sources:
  - End-user
  - Local Authority
  - Building Permit User
  - Software Architect
  - Software Developer
  - Security Compliance Engineer
  - System Customer
  - Others (specified as “Standardization Body”).
- **Priority**
  - A total number of 90 technical requirements elicited of “High Priority”.
  - A total number of 39 technical requirements elicited of “Medium Priority”.
  - A total number of 5 technical requirements elicited of “Low Priority”.
- **Existence**
  - A total number of 25 technical requirements elicited as “Existing”.
  - A total number of 109 technical requirements elicited as “Novel”.
- **Requirements Sepcificity**
  - A total number of 99 technical requirements specified as “Overall Requirement”.
  - A total number of 3 technical requirements specified as “Overall Requirement” and additionally elicited as country-specific requirements resulting from previous elicitation as User Requirements based on both criteria.
  - A total number of 35 technical requirements specified as “Country-specific”.

2.4.4 Analysis Results on ACCORD Cloud Architecture

The results of the Technical Requirements Analysis with regard to the ACCORD Cloud Architecture and its components can be summerised as follows:

Definition of new ACCORD Cloud Architecture components
The Technical Requirements Elicitation Phase 2.2 revealed the necessity to define new ACCORD Cloud Architecture components and to separate one previous subcomponnt of the ACCORD Framework, namely

- Definition of new component number 3 “Rule Repository and Provision”,
- Definition of new component number 4 “Information Services”,
- Separation of component number 2 “Data Dicitionaries”.

This elicitation phase also facilitated to distinguish between subcomponents (being further described in section 0), namely

- Definition of compliance checking microservices to be developed,
- Definition of APIs.

Assignment to ACCORD Cloud Architecture components

The assignment of all technical requirements to the ACCORD Cloud Architecture and its components has been proven. *Table 6* shows the reference and total numbers of technical requirements assigned to the ACCORD Cloud Architecure as generic requirement and to ACCORD Cloud Architecture components. *Table 9* provides a visual overview on the technical requirements assignement.

- In total 19 technical requirements are assigned as generic requirements to the ACCORD Cloud Architecture focussing on non-functional requirements dedicated to interoperability, security and usability aspects.
- A total number of 115 technical requirements are assigned to ACCORD Cloud Architecture components. Hereof, a total number of 55 technical requirements are assigned to multiple (two or more) components.
- The highest number of technical requirements is assigned to component number 5 “Cloud-based Building Permit Services” with 40 requirements in total.

ACCORD Cloud Architecture Component	TR Reference Number	Total no. of TR
0. ARCCORD Cloud Architecture (generic requirements)	1-19	19
1. Rule Formalization	20-35, 79-94	30
2. Data Dictionaries	79-81	2
3. Rule Repository and Provision	36, 79, 89	3
4. Information Requirements	37-40, 79, 95-96	7
5. Cloud-based Building Permit Services	41-47, 82-88, 95, 97-122	40
6. Model and Data Requirement Validation	48-51, 96, 123	6
7. Process Execution	52-54, 124	4
8. Data Storage	55-56, 89, 97-98, 125-126	7
9. Orchestrating Microservices	57, 124	2
10. Compliance Checking Microservices	58-75, 83-88, 90-93, 95, 97-122, 125, 127-131	58
11. Information Services	80, 115-116, 127-130, 132-134	10
12. APIs	76-78, 81, 83-87, 89, 91-95, 97-98, 117-123, 125-126, 129-134	32

Table 6. Technical Requirements numbers by ACCORD Cloud Architecture components.

TR	0	1	2	3	4	5	6	7	8	9	10	11	12
1	🟦												
2	🟦												
3	🟦												
4	🟦												
5	🟦												
6	🟦												
7	🟦												
8	🟦												
9	🟦												
10	🟦												
11	🟦												
12	🟦												
13	🟦												
14	🟦												
15	🟦												
16	🟦												
17	🟦												
18	🟦												
19	🟦												
20		🟦											
21		🟦											
22		🟦											
23		🟦											
24		🟦											
25		🟦											
26		🟦											
27		🟦											
28		🟦											
29		🟦											
30		🟦											
31		🟦											
32		🟦											
33		🟦											
34		🟦											
35		🟦											
36				🟦									
37					🟦								
38					🟦								
39					🟦								
40					🟦								
41						🟦							
42						🟦							
43						🟦							
44						🟦							
45						🟦							
46						🟦							
47						🟦							
48							🟦						
49							🟦						
50							🟦						
51							🟦						
52								🟦					
53								🟦					
54								🟦					

Table 7. Technical Requirements to ACCORD Cloud Architecture mapping.

TR	0	1	2	3	4	5	6	7	8	9	10	11	12
55									■				
56									■				
57										■			
58											■		
59											■		
60											■		
61											■		
62											■		
63											■		
64											■		
65											■		
66											■		
67											■		
68											■		
69											■		
70											■		
71											■		
72											■		
73											■		
74											■		
75											■		
76													■
77													■
78													■
79		■	■	■	■								
80		■	■									■	
81		■	■										■
82		■				■							
83		■				■					■		■
84		■				■					■		■
85		■				■					■		■
86		■				■					■		■
87		■				■					■		■
88		■				■					■		
89		■		■					■				■
90		■									■		
91		■									■		■
92		■									■		■
93		■									■		■
94		■											■
95					■	■					■		■
96					■		■						
97						■			■		■		■
98						■			■		■		■
99						■					■		
100						■					■		
101						■					■		
102						■					■		
103						■					■		
104						■					■		
105						■					■		
106						■					■		
107						■					■		
108						■					■		

Table 8. Technical Requirements to ACCORD Cloud Architecture mapping.

TR	0	1	2	3	4	5	6	7	8	9	10	11	12
109						■					■		
110						■					■		
111						■					■		
112						■					■		
113						■					■		
114						■					■		
115						■					■	■	
116						■					■	■	
117						■					■		■
118						■					■		■
119						■					■		■
120						■					■		■
121						■					■		■
122						■					■		■
123							■						■
124								■	■	■			
125									■		■		■
126													■
127											■	■	
128											■	■	
129											■	■	■
130											■	■	■
131											■		■
132												■	■
133												■	■
134												■	■

Table 9. Technical Requirements to ACCORD Cloud Architecture mapping.

2.5 Conclusion

This section has summarised and discussed the methodological approach, processes and steps related to the definition, elicitation and analysis of technical requirements to the ACCORD Cloud Architecture and its components. To achieve this goal, the following aspects have been covered:

- Provision of an overview on the phases and methodological steps of the Technical Requirements Collection, Elicitation and Analysis.
- Presentation of the developed Technical Requirements Elicitation criteria.
- Inclusion of ACCORD Framework User Requirements (D1.2) in the Technical Requirements elicitation process.

The main results of the Technical Requirements Elicitation and Analysis are resumed as follows:

- A summery of results on the Technical Requirements Elicitation Phases and on Technical requirements Elicitation criteria was presented.
- Technical Requirements Elicitation revealed the necessity to define new ACCORD Cloud Architecture components.
- The assignement of all technical requirements to the ACCORD Cloud Architecture and its components has been proven.
- A comprehensive list of Technical Requirements mapped to the ACCORD Cloud Architecture and its components was developed.
- The results of the Technical Requirements Elicitation and Analysis were used to redefine the ACCORD Cloud Architecture (see section 0) and supported the specification of its components and the alignment with demo use cases (described in section 5). They will

further on facilitate solution developments in Tasks 4.3 and 4.4, and the testing, validation and quality assurance in Task 4.5 respectively.

3 ACCORD Cloud Architecture Model

3.1 Introduction

Designing a software architecture needs to consider all the factors and functionalities that contribute to a system (Hofmeister et al., 2000; Bruegge & Dutoit, 2009). It also needs to take into account the various ways of improving the software evolvability (Breivold et al., 2012). The reliability of a software architecture, especially in the building regulation field, is a key quality of a given system (Atef et al., 2010). The ACCORD Cloud Architecture not only will consider these aspects, but also state-of-the art classifications and strategic aspects for developing data platforms, dataspaces and (data) ecosystems and recent, relevant EU legislations in the field (see Strnad and Schöning, 2023).

The ACCORD Cloud Architecture realizes these qualities by improving the process of building regulations through a seamless connection of all parties involved in the building regulation process. The created solution will improve building permit processes, this entails alls process steps from the moment a building permit is requested by applicants until the moment it will be granted by the building permit authority. To this end, the development of the ACCORD Cloud Architecture is based on the following assumptions:

- The building authorities and the local permitting authorities are interlinked.
- Data exchange works smoothly between involved parties and across components during the building permit process.
- The solutions can be catered towards country-specific- and overall European needs.

The ACCORD Cloud Architecture is developed with the following aims:

- It will improve the building process integration of many European countries (especially those represented in the project consortium). This will aim at enhancing interoperability among the countries by providing the general software infrastructure required for building permit processes. At the same time, the ACCORD solution will maintain the ability of setting independent services answering country-specific needs.
- It will provide a set of pre-defined building permit tools which are applicable to all participating countries, with the ability to further develop and adapt these tools according to country-specific requirements.
- It will improve the functional suitability of the building process through the novel requirements are going to be developed.

3.2 Methodological Approach

The methodological approach for developing the ACCORD Cloud Architecture is illustrated in *Figure 2.* It is based on an important artifact, namely the ACCORD Semantic Framework developed in T1.3. The ACCORD Framework provides an overview of the main components to be developed in ACCORD project. In the following, defining the specifics of the components to be developed is done through a questionnaire distributed upon ACCORD WP4 partners. Based on the questionnaire results, the ACCORD partners contributing to each component and the type of contributions are identified. The information gained is used to define the ACCORD Cloud Architecture components as described in section 3.3. Further demo-specific requirements and intended developments are taken into consideration in order to clearly outline the specifics of each component. The modifications of the ACCORD Semantic Framework are then mapped with the final ACCORD Cloud Architecture.

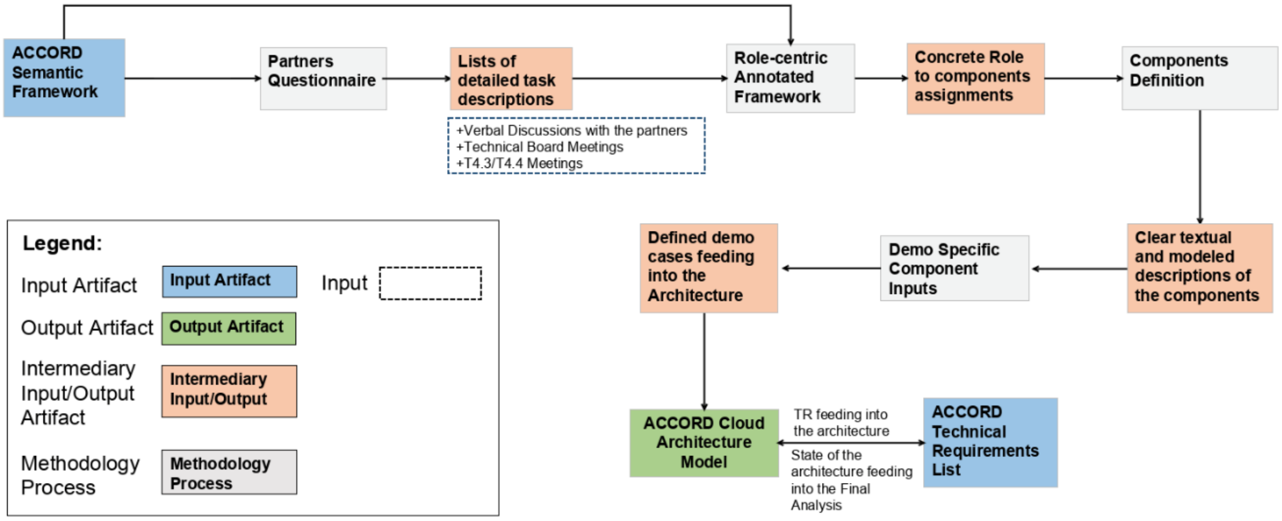

ACCORD Semantic Framework
Partners Questionnaire
Lists of detailed task descriptions
+Verbal Discussions with the partners
+Technical Board Meetings
+T4.3/T4.4 Meetings
Role-centric Annotated Framework
Concrete Role to components assignments
Components Definition
Clear textual and modeled descriptions of the components
Demo Specific Component Inputs
Defined demo cases feeding into the Architecture
ACCORD Cloud Architecture Model
TR feeding into the architecture
State of the architecture feeding into the Final Analysis
ACCORD Technical Requirements List
Legend:
Input Artifact
Input Artifact
Input
Output Artifact
Output Artifact
Intermediary Input/Output Artifact
Intermediary Input/Output
Methodology Process
Methodology Process

Figure 2. Methodoloy for developing the ACCORD Cloud Architecture.

3.2.1 ACCORD Semantic Framework

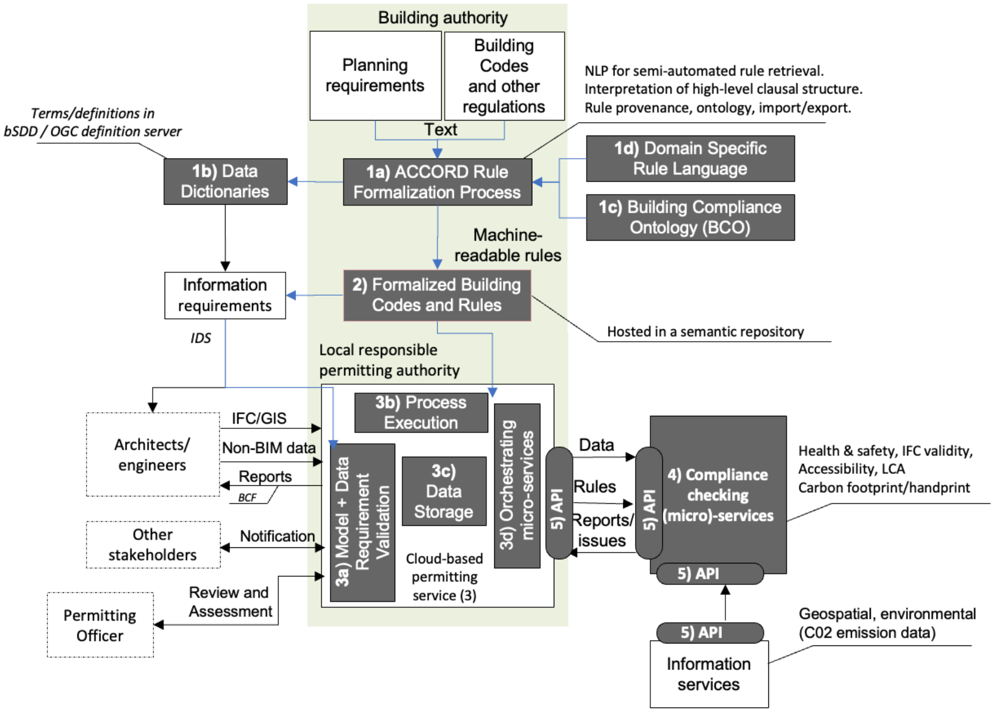

Building authority
Planning requirements
Building Codes and other regulations
Text
NLP for semi-automated rule retrieval.
Interpretation of high-level clausal structure.
Rule provenance, ontology, import/export.
Terms/definitions in bSDD / OGC definition server
1b) Data Dictionaries
1a) ACCORD Rule Formalization Process
1d) Domain Specific Rule Language
1c) Building Compliance Ontology (BCO)
Machine-readable rules
Information requirements
2) Formalized Building Codes and Rules
Hosted in a semantic repository
IDS
Local responsible permitting authority
Architects/ engineers
IFC/GIS
Non-BIM data
Reports
BCF
Other stakeholders
Notification
Permitting Officer
Review and Assessment
3b) Process Execution
3a) Model + Data Requirement Validation
3c) Data Storage
3d) Orchestrating micro-services
Cloud-based permitting service (3)
5) API
Data
Rules
Reports/ issues
5) API
4) Compliance checking (micro)-services
5) API
Health & safety, IFC validity, Accessibility, LCA
Carbon footprint/handprint
5) API
Information services
Geospatial, environmental (C02 emission data)

Figure 3. ACCORD Semantic Framework

The ACCORD Semantic Framework provides an overview of the main components and data flows to be developed and provided in the ACCORD project (see *Figure 3*). It is the result of the work conducted in Task 1.3. The ACCORD Framework provides clear designations of components (e.g. building permit authority) and information on the connections between components and subcomponents of the system including important API developments. The question of how to build the development of the ACCORD Cloud Architecture on this framework, was discussed by ACCORD partners during Technical Board-, work package 4 and task 4.3 and task T4.4 meetings focussing on building compliance checking components, -microservices and APIs. Throughout these meetings, the main identified knowledge gaps were the following:

1. **No clear API definitions:** In the ACCORD Semantic Framework, no formal APIs are defined.
2. **Detailed list of compliance checking microservices: E**ach compliance checking microservice needs to be clearly identified based on the specifications of demo use cases.

3.2.2 Partner's Questionnaire

The partners' questionnaire was designed based on a top-down approach, as outlined and explained by (Linaker et al., 2015). This top-down approach recommends defining the goals first, followed by designing the single questions of the questionnaire. In the case of the ACCORD project, the goal of the questionnaire was to elicit the contributions of ACCORD partners to the development of ACCORD Cloud Architecture components based on the information provided by partners in the most detailed manner. The questionnaire was conducted and completed by 12 ACCORD consortium partners in June 2023 and the following months. The background of the contributing ACCORD partners is shown in *Table 10*.

Type of organization	Number of Partners	Name of organization
Industry	4	CloudPermit (CP), Ontotext (ONTO), Solibri (SOL), Future Insight (FUI)
Academia	4	Cardiff University (CU), FUNITEC, Jönköping School of Engineering (JTH)
Research Organization	2	VTT, Fraunhofer IBP (FhG)
Standardisation Body	2	Open Geospartial Consortium (OGC), "Urban Data Platform - Planen und Bauen, Leitstelle XPlanung/XBau, Landesbetrieb Geoinformation und Vermessung der Freien und Hansestadt Hamburg" (HAM)

Table 10. ACCORD Cloud Architecture: Contributing Partners' Background.

The main objectives of this questionnaire are as follows (see *Table 11*):

1. **To capture the contributions of the partners:** This is done by providing a detailed description of each of the respective contributions.
2. **To narrow the scope of contribution for each partner:** Each partner should provide adequate responses to which component development they expect to contribute to, either based on their expertise or discussions during the project.
3. **Provide an expected outcome:** outcomes are the development of components, a service that is provided to the component, or a tool that can facilitate the development.
4. **Provide a context for collaboration amongst partners:** The aim is to capture desires for collaboration between project partners, or technical interoperability between developed solutions.

The questionnaire provided the following questions:

Q1	What is your role in the Organization?
Q2	To which component(s), in particular, are you contributing?
Q3	What your contribution exactly is?
Q4	What is the foreseen outcome of your contribution? i.e. software artifact?
Q5	With which other component(s) are you expecting the component(s) that you are involved in to interact with?
Q6	Does the component that you are involved in, need an input from other components? if yes, what is the input and component?
Q7	Does the component that you are involved in, give an output to be used by other components? if yes, what is the output and what is/are the targeted component(s)?

Table 11. ACCORD Architecture Questionnaire.

Partner/ Compo-nent No.	1a	1b	1c	1d	2	3	3a	3b	3c	3d	4	5
FUI				🟦	▨	🟩					🟩 ▨	🟩 🟥
FUNITEC	▨					▨ 🟥					🟩	
FhG	▨	🟩	▨	▨	🟩	🟩 🟦 ▨ 🟥					🟩	🟩
VTT						🟦 ▨ 🟥					🟩	
HAM		🟦			🟦	🟦					🟦	
OGC	🟦 🟥	🟩			🟩	🟩 🟦 🟥						
Legend												
Direct contribution:						🟩						
Direct Interaction:						🟦						
Input to the contributor's component						▨						
Output of the contributor's component:						🟥						

Table 13 illustrates the results of the questionnaire. The numbering of the components at this stage is based on the ACCORD Semantic Framework. The table distinguishes between the following types of partner's contributions:

1. **Direct contribution:** answers Q2 and Q3, defining the key components a partner contributes to.
2. **Direct Interaction:** answers Q5, determining the component(s) the partner's main to be developed component(s) are interfacing with.
3. **Input to the contributor's component:** answers Q6, defining and explaining the input dependencies with regard to other component(s).
4. **Output of the contributor's component:** answers Q7, providing a view on the component(s) that receive(s) input from the contributor's component(s).

Partner/ Component No.	1a	1b	1c	1d	2	3	3a	3b	3c	3d	4	5
CP	⬛				🟦	🟩					🟦 ⬛ 🟥	🟩
ONTO	🟩	🟩		🟩	🟩	⬛ 🟥					🟦	🟩
SOL			🟩	🟩		⬛ 🟥					🟩 ⬛	🟩 🟦 ⬛ 🟥
CU	⬛ 🟥	⬛ 🟥	⬛ 🟥	⬛ 🟥	⬛ 🟥	🟦	⬛ 🟥	⬛ 🟥	⬛ 🟥	⬛ 🟥	🟦	🟩
JTH	🟩	🟩	🟩	🟩	🟦 🟥							
OGC	🟦 🟥	🟩				🟩	🟩 🟦 🟥					
Legend												
Direct contribution: Direct Interaction: Input to the contributor's component Output of the contributor's component:						🟩 🟦 ⬛ 🟥						

Table 12. Mapping of ACCORD partner's contributions to ACCORD Framework components.

Partner/ Component No.	1a	1b	1c	1d	2	3	3a	3b	3c	3d	4	5
FUI				🟦	⬛	🟩					🟩 ⬛	🟩 🟥
FUNITEC	⬛					⬛ 🟥					🟩	
FhG	⬛	🟩	⬛	⬛	🟩	🟩 🟦 ⬛ 🟥					🟩	🟩
VTT						🟦 ⬛ 🟥					🟩	
HAM		🟦			🟦	🟦					🟦	
OGC	🟦 🟥	🟩			🟩	🟩 🟦 🟥						
Legend												
Direct contribution: Direct Interaction: Input to the contributor's component Output of the contributor's component:						🟩 🟦 ⬛ 🟥						

Table 13. Mapping of ACCORD partner's contributions to ACCORD Framework components.

3.2.3 Role-centric Annotated Framework

In this step, the main objective is to assign each component to a set of partners who are responsible (a) for the development of the component and (b) for writing the deliverable section which shall explain the technical details of each component (see section 5). The assignment was done based on the following inputs:

1. **Input of the questionnaires:** The questionnaires have defined the main roles of all the participating partners in T4.2. In light of this information, the areas of expertise and concrete development tasks of the partners are defined (see section 3.2.2).
2. **The ACCORD Semantic Framework:** The artefact resulting from T1.3 is also used to feed the questionnaire by providing an overview of the main components and their relations. The ACCORD Semantic Framework is forming the basis of the ACCORD Cloud Architecture.
3. **Discussions with ACCORD partners:** The discussions with ACCORD partners took place both, in the WP4 bi-weekly meetings and in task meetings. They were targeted towards gaining a comprehensive understanding on the information provided in the questionnaires. Focus was on identifying individual contributions of partners to solution dvelopment and leadership of component development.
4. **Technical Board Meetings:** Discussions and brainstorming sessions during Technical Board meetings supported the ACCORD partners in gaining a common understanding on what solutions should be developed and by whom and which technologies should be integrated.
5. **WP4 Task Meetings:** T4.3 and T4.4 meetings were focussing on the development of APIs and the implementations of the compliance checking microservices.
6. **Berlin Project Meeting:** In September 2023, a 4-hour workshop session was dedicated to the presentation of solutions to be developed by ACCORD partners followed by brainstorming sessions and immediate feedback by participants. An Intense desktop work allowed to elicit the workshop results and fill the knowledge gaps from previous process steps, respectively.

The process of the roles' assignment started with aligning the information gathered on the ACCORD Semantic Framework and through the partner’s questionnaire. The following actions were performed:

- Information provided by ACCORD partners in the questionnaire allowed to identify the types of contributions to the development of ACCORD Framework components (see *Table 12*).
- All the assignments were discussed with all the partners, and further comments and suggestions were received. In this step, the identification of a leading partner for every component is finalized and agreed upon.
- A detailed view of who has developed the subcomponents has been gathered throughout the discussions during the bi-weekly and technical board meetings.
- This list served as the basis for the later assignment of parter contributiuon to the ACCORD Cloud Architecure components after redefinement (see *Table 14*).

3.2.4 ACCORD Framework Components Refinement

In the following, the ACCORD Framework components were refined by performing the following activities:

- The structure of each component was captured through either a structural model (UML components diagram) or detailed descriptions of the interfaces of each component.
- The components' behaviour was captured through either a dynamic models (UML sequence diagram) or detailed descriptions that trace the main functions the component should be able to perform.
- Refining the inputs and outputs of each component.

- Defining clear interfaces between the components.
- Defining the technologies.
- Providing detailed information on compliance checking microservices and defining microservices for demo use cases. Each use case requires one or more microservice to be developed by ACCORD partners.
- Categorizing the APIs to be developed based on their individual purpose.

The refinement of ACCORD Framework components allowed to model the ACCORD Cloud Architecture with its components (see section 3.3) and to specify each component, the results being presented in section 5.

3.2.5 Demo-specific Component Inputs

In this stage, communication was made with ACCORD partners that intend to develop or integrate compliance checking microservices or to provide their off-the-shelf products or own implementations of components. Furthermore, demo-specific requirements to components were aligned with contributions proposed by WP4 partners. Based on this alignement, a comprehensive application list was established that will facilitate the mapping of changes and updates during solution development, if any.

3.2.6 ACCORD Technical Requirements List

The ACCORD Technical Requirements list is the result of the Technical Requirement Elicitation and Analysis described in section 2. The comprehensive list of 134 elicited technical requirements assigned to ACCORD Cloud Architecture components is presented in **Appendix 2** of this document. As outlined in Section 2, the second phase of the Technical Requirements Elicitation facilitated the definition of new ACCORD Cloud Architecture components and subcomponents that were not present in the ACCORD Framework, such as compliance checking microservices and APIs.

Based on this final component list, the intented contributions by ACCORD partners to solution development were mapped with previous results. Additional information was provided by component leaders, (1) on information elicitated from technical requirements that should be considered in the final architecture, and (2) feedback on the applicability of technical requirements to the ACCORD Cloud Architecture.

3.3 Structure of the ACCORD Cloud Architecture

The ACCORD Cloud Architecture stems from two main sources: (1) the ACCORD Semantic Framework and the definition of its components (see section 3.2.1) and (2) the elicited Technical Requirements (see section 3.2.6).

The final structure of the ACCORD Cloud Architecture distinguishes between twelve main components:

1. **Rule Formalization:** The Rule Formalization Tool is responsible for all the various activities on the building authority side. Its subcomponents oversee various tasks that are related to formalize building codes and regulations, providing a domain-specific rule language and managing building codes and rules.

2. **Data Dictionaries:** This component provides access to data dictionaries with the aim to map explicit definitions and terms being present in regulatory documents. It also provides a reconciliation capability to conduct fuzzy mapping and other lookups when terminologies do not precisely align.

3. **The Rule Repository and Provision:** This component provides the ability to store and to translate or interpret the codes and rules provided by the Rule Formalization Tool.

4. **Information Requirements:** This component will be responsible for providing formalized definitions of information requirements for building permit processes to other components of the ACCORD Cloud Architecture.

5. **The Cloud-based Building Permit Services:** This component manages the overall process of building permitting, and acts as an intermediary between the rule provision component and the compliance checking microservices.

6. **Model- and Data Requirement Validation:** This component validates buildings model (IFC format) against bSDD and IDS specifications and validates additional external data sources (e.g. a zoning plan) against input requirements.

7. **Process Execution:** This component executes the process flow for a building permit, from the first initial application to final granting of the permit. It needs to interact with the microservice orchestration component in order to execute checks as part the process flow.

8. **Data Storage:** This component is ‘single source of truth’ location to store the building models in IFC format. It is accessed by other components via API for retrieving (parts of) the models with the aim to execute compliance checks.

9. **Orchestrating Microservices:** This component orchestrates (coordinates) the API communication between components of the ACCORD Cloud Architecture.

10. **Compliance Checking Microservices:** This component summerizes the various compliance checking services of ACCORD. It runs compliance checks based on the input data (building models in IFC and others) and rules, and returns the results back to the process execution component.

11. **Information Services:** The information services provide additional information and reasoning capabilities to the compliance checking microservices such as geospatial or environmental data.

12. **APIs:** The APIs act as an intermediary between all the various components of the ACCORD Cloud Architecture. 7 APIs are distinguished as follows:
    - API (1) Definitions API
    - API (2) Building Codes and Rules API
    - API (3) Information Services APIs
    - API (4) Data APIs
    - API (5) Management APIs
    - API (6) Results API
    - API (7) Reconciliation API.

This ACCORD Cloud Architecture with its various components is illustrated in

Figure 4, and the responsible partners for each component are listed in *Table 14*.

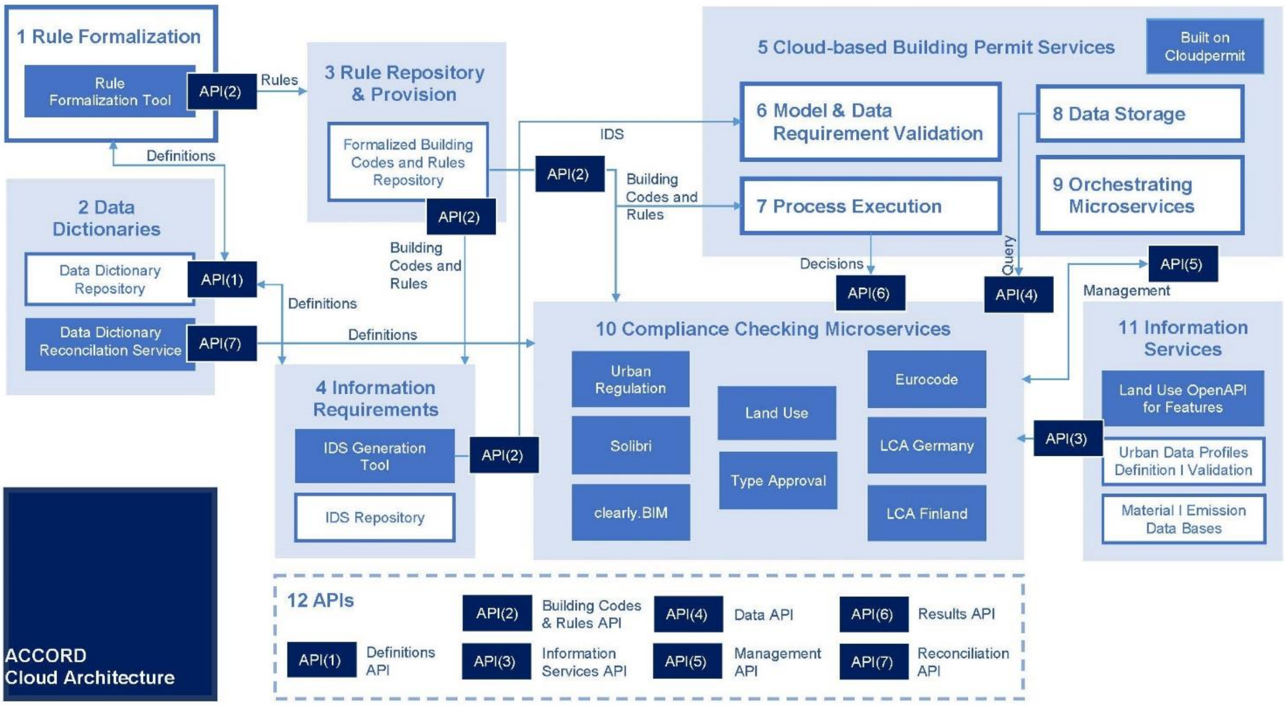

1 Rule Formalization
Rule Formalization Tool
API(2)
Rules
Definitions
3 Rule Repository & Provision
Formalized Building Codes and Rules Repository
API(2)
2 Data Dictionaries
Data Dictionary Repository
API(1)
Data Dictionary Reconcilation Service
API(7)
Building Codes and Rules
Definitions
Definitions
IDS
API(2)
Building Codes and Rules
5 Cloud-based Building Permit Services
Built on Cloudpermit
6 Model & Data Requirement Validation
7 Process Execution
8 Data Storage
9 Orchestrating Microservices
Decisions
Query
API(6)
API(4)
API(5)
Management
4 Information Requirements
IDS Generation Tool
API(2)
IDS Repository
10 Compliance Checking Microservices
Urban Regulation
Solibri
clearly.BIM
Land Use
Type Approval
Eurocode
LCA Germany
LCA Finland
11 Information Services
Land Use OpenAPI for Features
API(3)
Urban Data Profiles Definition I Validation
Material I Emission Data Bases
ACCORD Cloud Architecture
12 APIs
API(1) Definitions API
API(2) Building Codes & Rules API
API(3) Information Services API
API(4) Data API
API(5) Management API
API(6) Results API
API(7) Reconciliation API

Figure 4. ACCORD Cloud Architecture Model.

Component Number/Name	**Responsible Partners**
1 Rule Formalization	FUNITEC
2 Data Dictionaries	
2a Data Dictionary Repository	BSI
2b Data Dictionary Reconciliation Service	ONTO
3 The Rule Repository and Provision	ONTO, CU
4 Information Requirements	CU
5 Cloud-based Building Permit Services	CP
6 Model- and Data Requirement Validation	BSI, FUI, SOL
7 Process Execution	CP
8 Data Storage	FUI, ONTO
9 Orchestrating Microservices	CP
10 Compliance Checking Microservices	
(1) Solibri	SOL
(2) Future Insight Clearly.BIM	FUI
(3) LCA Finland	VTT
(4) LCA Germany	FhG
(5) Eurocode Compliance Checking	AE
(6) Urban Regulations Checking	FUNITEC
(7) Land Use Building Compliance Checking	FhG
(8) Type Approval Building Compliance Checking	FhG
11 Information Services	
(1) Land Use OpenAPI for Features	HAM
(2) Urban Data Profile Validation	OGC
(3) Material Emissions Database	VTT
12 APIs	
API (1) Definitions API	BSI
API (2) Building Codes and Rules API	CU
API (3) Information Services APIs	HAM, OGC, FUNITEC, VTT, FhG
API (4) Data APIs	FUI, ONTO
API (5) Management APIs	CP
API (6) Results API	CU
API (7) Reconciliation API	ONTO

Table 14. ACCORD Cloud Architecture components and responsible partners.

3.4 Conclusion

In this section, the methodological approach for creating the ACCORD Cloud Architecture was explained. The methodological steps included gathering the various partners roles in the ACCORD Cloud Architecture through a detailed questionnaire, the assignement of partner contributions to each component, the thorough definition of each component based on the knowledge at that stage. Also, the demo specific contributions were considered and discussed. Finally, the elicited technical requirements were again investigated and applied to the ACCORD Cloud Architecture.

4 Summary of ACCORD Regulation Digitisation Approach

4.1 Introduction

This section will provide a summary of the methodology that has been developed to digitalise construction regulations. This methodology is described in more detail in Deliverable D2.2. However, as it provides important context to the structure of the cloud architecture, it is recapped here.

This methodological process captures the entire process starting from the original document i.e., PDF, to a machine-readable document *(that software can read, parse, and understand the structure of)* to a fully machine-operable document (*that software can use to instigate a set of complex processes*).

The remainder of this section will cover each aspect of this methodology and the supporting software developments. Section 4.2 will cover the building compliance ontology that provides a formalized semantic vocabulary for representing construction regulations. Section 4.3 will cover the overall rule formalisation process. Section 4.4 will describe the domain-specific rule language developed to formalize the specification of decisions within the wider ACCORD digitisation methodology. Section 4.5 will then conclude the section and provide links between these components and the ACCORD cloud architecture components described in Section 0; especially this will consider the rule formalization tool being developed to support and provide an accessible user interface for regulatory authors to use to create machine-operable construction regulations for use within the ACCORD semantic framework.

4.2 Building Compliance Ontology

The Building Compliance Ontology within the ACCORD Framework (Component 1c in *Figure 3*) serves a critical purpose in enabling interoperability and harmonising knowledge from diverse sources as well as providing a robust semantic foundation for the representation of digitised building regulations within the ACCORD project.

The ACCORD building compliance ontology (named Architecture, Engineering, Compliance Checking and Permitting Ontology (AEC3PO)) acts as the primary interface connecting knowledge-providing (i) building codes, regulations, and standards, (ii) compliance and permitting processes and documentation, and (iii) compliance and permitting actors. AEC3PO aims to model all aspects of compliance and permitting on the AEC domain, across different regulatory systems. It is organised into modules each of which represents one of the key aspects that the AEC3PO ontology provides a semantic representation for. The modules are summarised below, and more detail on the AEC3PO ontology can be found in Deliverable D2.2:

- **Document:** Describes building-compliance related documents, their subdivisions, down to statements and atomic tagged strings or figures.
- **Statement:** Describes things stated in a building compliance-related document.
- **RASE_Statement:** Describes statements decomposed following the Requirement Application Selection and Exception (RASE) methodology.
- **Data_Requirement:** Describes the data requirements that are elicited from a statement.
- **Evidence:** Describes the evidence that an actor in the compliance and permitting process needs to provide to prove that the requirements derived from a Statement have been met.
- **CheckMethod:** Models the operationalisation of a check statement, usually executed to control the conformance of some entity.
- **Feature_Of_Interest:** Describes an entity (feature) of a site, building, or piece of infrastructure that is of interest for some purpose. Typically, this will be a building component that needs to be compliant to regulations or be documented in the permitting process.
- ***Checking_Act:*** Describes the act of checking some entities for something and generating a compliance verification report.
- **Model:** Provides a description of the metadata of BIM models.
- **Legal_Verifier:** Describes actors of the compliance and permitting process that have the legal capacity to verify that a specific statement of a compliance document has been met in a satisfactory manner.
- **Table:** Describes tables as representations of data in rows and columns.

There is a risk when IFC data is not structured according to the correct ontology and/or semantics. This is further elaborated in T3.2 and the deliverable about information reliability measures (from WP3).

4.3 ACCORD Rule Formalization process

This section will summarise the ACCORD rule formalization process (Component 1a in Figure 3), originally documented in D2.2. This approach produces, as an output, a knowledge graph representing a machine-operable construction regulation. This knowledge graph will utilise the semantics defined within the Building Compliance Ontology (described in the previous section).

Several methods to achieve this (which are described in more detail in D2.2): (1) a manual approach, (2) an NLP derived automated approach and (3) a hybrid approach, combining elements of both previous approaches.

The motivation for providing multiple approaches is because; (a) there is a significant potential for the automated approaches to dramatically increase the rate of digitisation of construction regulations, (b) however, the accuracy level of automated translation methods is not yet fully proven. Thus, a hybrid approach has the potential to provide some level of automated conversion where an acceptable level of accuracy can be achieved, coupled with manual refinement where acceptable accuracy cannot be achieved.

The ACCORD methodology proposes five abstract steps, of which we will provide a manual, automated (NLP), and hybrid approach for each of them. These steps and the specific manual, NLP and hybrid implementations are shown in *Figure 5* and described in more detail in D2.2.

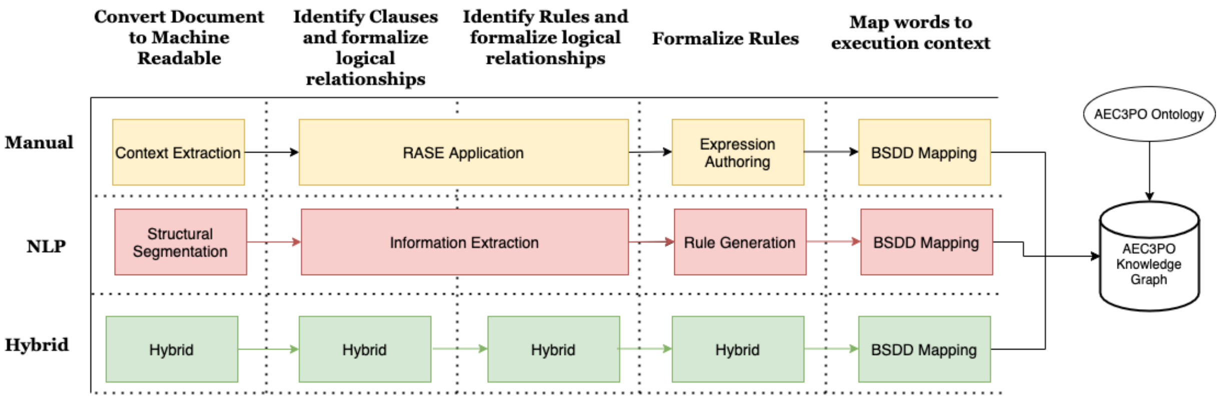
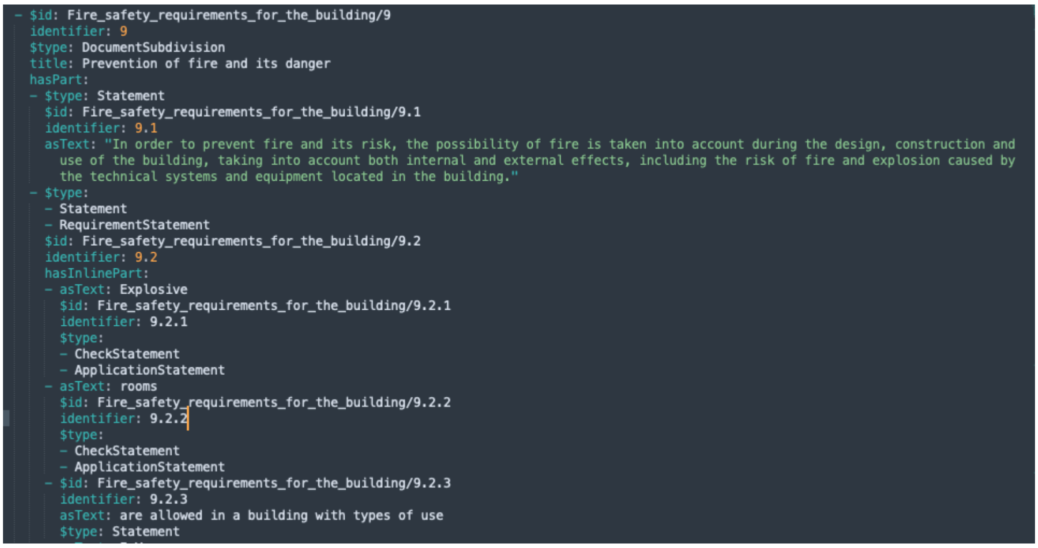


Figure 5. ACCORD Digitisation Methodology.

4.4 Domain Specific Rule Language

This section will summarise the ACCORD domain specific rule language (Component 1d in *Figure 3*), elicited in D2.2. The domain specific rule language represents the serialisation of the machine-operable regulations, including their structure, the logical relationships within the structure and the formal expression of any rules within the document. An example of this serialisation is shown in *Figure 6*. This figure shows a regulation document serialised, using the concepts from the AEC3PO ontology using YAML-LD.



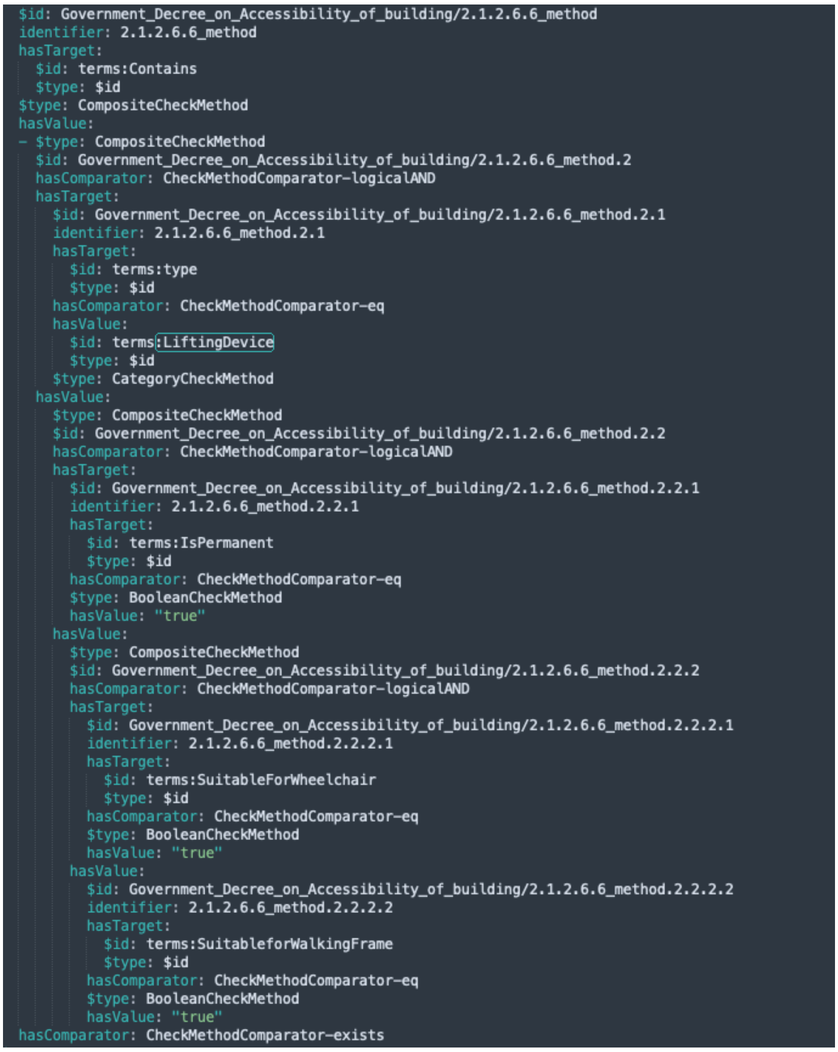
```
- $id: Fire_safety_requirements_for_the_building/9
  identifier: 9
  $type: DocumentSubdivision
  title: Prevention of fire and its danger
  hasPart:
  - $type: Statement
    $id: Fire_safety_requirements_for_the_building/9.1
    identifier: 9.1
    asText: "In order to prevent fire and its risk, the possibility of fire is taken into account during the design, construction and
      use of the building, taking into account both internal and external effects, including the risk of fire and explosion caused by
      the technical systems and equipment located in the building."
  - $type:
    - Statement
    - RequirementStatement
    $id: Fire_safety_requirements_for_the_building/9.2
    identifier: 9.2
    hasInlinePart:
    - asText: Explosive
      $id: Fire_safety_requirements_for_the_building/9.2.1
      identifier: 9.2.1
      $type:
      - CheckStatement
      - ApplicationStatement
    - asText: rooms
      $id: Fire_safety_requirements_for_the_building/9.2.2
      identifier: 9.2.2
      $type:
      - CheckStatement
      - ApplicationStatement
    - $id: Fire_safety_requirements_for_the_building/9.2.3
      identifier: 9.2.3
      asText: are allowed in a building with types of use
      $type: Statement
```

Figure 6. YAML Expression Example.

In addition to documenting the structure of the regulation document, the domain-specific language formalised the specification of rules. Some examples of these rules are shown in Table 15, and their serialisation in the domain-specific language is shown in *Figure 7*.

Example Expression
:IsExternal == true
:Width > 1.2 :M
:type == :House
:UsageCategory==:IV
:Walls exists => (:IsExternal == true)
:AdjacentSpaces forAll => (:FireSafeDesign==true)
(:tan((:Slope *(pi/180)))*100) > 5%
:Contains exists => (:type == :LiftingDevice && :IsPermanent == true && :SuitableForWheelchair == true && :SuitableforWalkingFrame == true)

Table 15. Example Expressions.



```
$id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method
identifier: 2.1.2.6.6_method
hasTarget:
  $id: terms:Contains
  $type: $id
$type: CompositeCheckMethod
hasValue:
- $type: CompositeCheckMethod
  $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2
  hasComparator: CheckMethodComparator-logicalAND
  hasTarget:
    $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.1
    identifier: 2.1.2.6.6_method.2.1
    hasTarget:
      $id: terms:type
      $type: $id
    hasComparator: CheckMethodComparator-eq
    hasValue:
      $id: terms:LiftingDevice
      $type: $id
    $type: CategoryCheckMethod
  hasValue:
    $type: CompositeCheckMethod
    $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2
    hasComparator: CheckMethodComparator-logicalAND
    hasTarget:
      $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.1
      identifier: 2.1.2.6.6_method.2.2.1
      hasTarget:
        $id: terms:IsPermanent
        $type: $id
      hasComparator: CheckMethodComparator-eq
      $type: BooleanCheckMethod
      hasValue: "true"
    hasValue:
      $type: CompositeCheckMethod
      $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2
      hasComparator: CheckMethodComparator-logicalAND
      hasTarget:
        $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2.1
        identifier: 2.1.2.6.6_method.2.2.2.1
        hasTarget:
          $id: terms:SuitableForWheelchair
          $type: $id
        hasComparator: CheckMethodComparator-eq
        $type: BooleanCheckMethod
        hasValue: "true"
      hasValue:
        $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2.2
        identifier: 2.1.2.6.6_method.2.2.2.2
        hasTarget:
          $id: terms:SuitableforWalkingFrame
          $type: $id
        hasComparator: CheckMethodComparator-eq
        $type: BooleanCheckMethod
        hasValue: "true"
hasComparator: CheckMethodComparator-exists
```

Figure 7. Domain Specific Language Serialisation.

4.5 Conclusion

This section has summarised the ACCORD methodology for the digitisation of construction regulations. The four key elements of this methodology; (a) the rule formalization process, (b) the building compliance ontology, (c) the domain specific rule language and (d) the rule formalization tool, have been summarised. The key links between these components and the ACCORD Cloud Architecture components are:

- **Data Dictionaries (described in Section 5.2):** Will be utilised to host and serve (via an API) the data dictionaries created by the digitisation process. These data dictionaries will provide mappings between the terms used in within the digitised regulations and how those terms are realised within construction data models. These mappings will be generated using the rule formalization tool and then uploaded to the data dictionary component (Component 1b in *Figure 3*).
- **Formalized Building Codes and Rules (described in Section 5.4):** Will be used to host and serve (via an API) the machine-operable rules. The machine-operable rules will be created following the rule formalization process, supported by the rule formalization tool. This will produce a machine-operable version of the target regulations, in a semantic format,

utilising the terminology described by the Building Compliance Ontology. This will then be uploaded to the formalized building codes and rules component (Component 2 in *Figure 3*).

- **Rule Formalization Tool (described in Section 5.1):** The final component is the user interface responsible for allowing regulatory professionals to formalize regulations into the ACCORD domain specific rule language. This will provide accessible platform that enables the creation of formalized building codes and rules (including their upload to the building codes and rules component (Component 2 in *Figure 3*) using the data dictionaries provided by the data dictionary component (Component 1b in *Figure 3*).

5 ACCORD Cloud Architecture Components

This section will describe the components of the ACCORD Cloud Architecture in more detail. The components (as described in Figure 4) are organised as follows:

Rule Formalization: (1) Rule Formalization tool – described in Section 5.1 Rule Formalization Tool.

Data Dictionaries: (1) Data Dictionary Repository – described in Section 5.2 Data Dictionary Repository, and (2) Data Dictionary Reconciliation Service – described in Section 5.3 Data Dictionary Reconciliation Service.

Rule Repository & Provision: (1) Formalized Building Codes and Rules Repository - described in Section 5.4 Formalized Building Codes and Rules Repository.

Information Requirements: (1) IDS Repository – described in Section 5.5 IDS Repository and (2) IDS Generation Tool – described in Section 5.6 IDS Generation Tool.

Cloud-based Building Permit Services: (1) Model- and Data Requirement Validation – described in Section 5.7 Model & Data Requirement Validation, (2) Process Execution – described in Section 5.8 Process Execution, (3) Data Storage – described in Section 5.9 Data Storage, and (4) Orchestrating Microservices – described in Section 5.10 Orchestrating Microservices.

Compliance Checking Microservices: A collection of 8 microservices – described in Section 5.11 Compliance Checking Microservice(s).

Information Services: A collection of 3 information services – described in Section 5.12 Information Services.

APIs: A set of APIs either re-used or to be created by the ACCORD project. These are (1) Definitions API – described in Section 5.13 APIs, (2) Building Codes and Rules API, (3) Information Services API, (4) Data API, (5) Management API, (6) Results API and (7) Reconciliation API.

For each of the key main components, the following elements will be described:

- **Description and Objective:** Will outline the main purpose of the component and its task within the ACCORD cloud architecture.
- **Structural Description:** Will outline the structure of the component's implementation.
- **Behavioural Description:** Will outline how the component will interact with other components in the ACCORD cloud architecture.
- **Used Technologies:** Will described the technologies to be utilised as part of the component's development.
- **Component Implementation:** Description of any implementation specific details of the component that must be considered.

The section closes with the alignenment of ACCORD Cloud Architecture component to demo use cases.

5.1 Rule Formalization Tool

This section will document the rule formalization tool component of the ACCORD Cloud Architecture.

5.1.1 Description and Objective

The Rule Formalization Tool is a user interface for regulatory authors to create machine-operable construction regulations for use within the ACCORD Semantic Framework. It will consist of a concrete implementation of the ACCORD digitisation approach (summary provided in Section 4 Summary of ACCORD Regulation Digitisation Approach), enabling the import of regulatory texts, addition of rules, contextualisation of rules with concepts from a data dictionary and an upload of digitised regulatory text to the ACCORD formalized building codes and rules repository. In brief, this tool provides a set of user interfaces accessible to regulations experts that will assist them in the process of formalising regulations into the ACCORD domain specific rule language.

This component is primarily developed in WP2 and will be reported in D2.3. So, it will not be described here in detail. However, the requirements realised by the tool are illustrated in **Appendix 2**.

5.2 Data Dictionary Repository

This section will document the data dictionary repository component of the ACCORD Cloud Architecture.

5.2.1 Description and Objective

The data dictionary repository functionality within the ACCORD Cloud Architecture will be provided by bsDD (buildingSMART Data Dictionary).

The bSDD - buildingSMART Solution for Data Dictionaries is an online service hosting classes (terms) and properties, allowed values, units, translations, relations between those and more. It supports a standardised workflow to guarantee data quality, information consistency and interoperability. BIM modellers use the bSDD for easy and efficient access to all kinds of standards to enrich their models. BIM Managers use the bSDD to reference Information Delivery Specifications (IDS) and check BIM data for validity. Content creators benefit from having one entry point to various BIM tools and platforms.

Besides national and international classification systems (e.g. Uniclass, CCI) and domain-specific standards (e.g. ETIM, Ifc for Airport), company-specific standards can be stored in bSDD as well. The bSDD implements the ideas from ISO 12006-3, ISO 23386 and Linked Data standards.

Publishing a data dictionary in bSDD, enables immediate access to the data from the software solutions integrated with the platform, as well as programmatic access via its API. Freely accessible and standardised definitions contribute to interoperability of digital construction projects and consistent terminology in specifications.

The motivation for selecting an existing solution such as bsDD is: (a) it is already widely used within the construction sector, (b) it meets the requirements of the ACCORD project, and (c) it will allow ACCORD development resources to focus on more innovative areas.

5.2.2 Structural Description

bSDD employs a structured schema, facilitating a uniform semantic framework for BIM objects and their properties. This standardization is crucial for data interoperability across different BIM systems. At the heart of bSDD is a database with all dictionaries. Each dictionary may contain list of classes and properties. Classes mainly define physical objects (e.g. a door) and properties are the attributes to describe classes (e.g. thickness). The content of dictionaries can be related to each other, creating a connected graph.

bSDD enables external systems to programmatically query and retrieve standardized BIM class and property definitions through its API (Application Programming Interface). This is how most BIM software and other apps can use the data stored in the bSDD. Apart from that, there is the bSDD Search page, where people can look up the content.

Authors can publish content to bSDD through the API or the dedicated Management Portal structured according to the template in JSON format.

The service provides a mechanism for continuous updates and expansion of its data repository, ensuring alignment with evolving construction standards and practices. The multi-lingual support addresses the challenge of language barriers in global projects.

API based communication will be utilised by other components of the ACCORD Cloud Architecture to retrieve and update data dictionary definitions.

5.2.3 Behavioural Description

There are three foreseen scenarios for the usage of bSDD on the project: (1) register new terms and definitions by the rule formalisation tool, (2) to select registered terms for the creation of information specifications either manually or automatically, (3) to retrieve term definition as part of the compliance checking process conducted by Compliance Checking Microservices. These are illustrated in *Figure 8*.

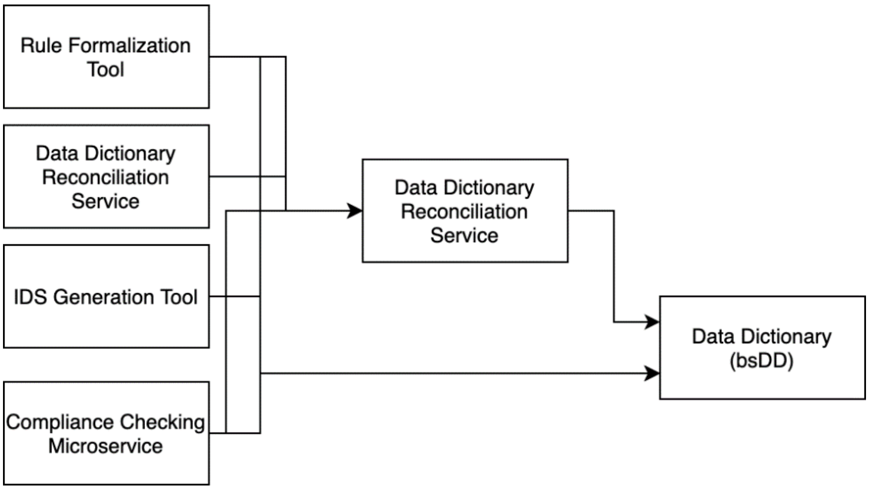

Rule Formalization Tool
Data Dictionary Reconciliation Service
IDS Generation Tool
Compliance Checking Microservice
Data Dictionary Reconciliation Service
Data Dictionary (bsDD)

Figure 8. Data Dictionary Repository Interactions.

5.2.4 Used Technologies

As described previously this component will utilise the existing bsDD. Specifically, the technologies used by the bSDD are:

- Web-Based Platform: bSDD is hosted on a cloud-based platform, ensuring accessibility and scalability. Utilizes web server technologies and cloud storage.
- Database: Relational Database Management System (RDBMS). Stores and manages a vast amount of structured data, including semantic definitions and attributes of BIM objects. The data schema is based on ISO23386 and ISO12006-3 and adjusted to practical needs.
- API interface: Facilitates programmatic access to bSDD data, allowing for integration with various BIM software and custom applications. bSDD offers the RESTful OpenAPI. The primary response format is JSON, but the API also supports other media types including RDF/TTL. Apart from that, the bSDD also offers experimental GraphQL API interface.
- Semantic Web: bSDD utilizes ontology frameworks to facilitate retrieving the structured content in a form of graph.

5.2.5 Component Implementation

Given that this component is utilising an existing solution there is no specific implementation associated with it. However, there are several implementation considerations that other components utilising this component should be aware of:

- There is a risk of following different data structures, hence there may be a need to apply ETL (Extract, Transform, Load) tools or custom scripts for data mapping and transformation.
- Ease of accessibility and integration is crucial for integrating other services and platforms with bSDD. The complex nature of the data schema resultant from the ISO compliance of bsDD can increase the effort needed to integrate with the platform.
- In some cases, the ability to perform reconciliation is needed, when definitions within the data dictionary do not exactly match those used within the regulations, thus ACCORD will provide a reconciliation component that is described in the next section.

5.3 Data Dictionary Reconciliation Service

This section will document the data dictionary reconciliation component of the ACCORD cloud architecture.

5.3.1 Description and Objective

The data dictionary reconciliation services will perform data matching between input requests and data stored within the data dictionary repository. A variety of different matching technologies can be utilised to perform this matching. The purpose of this is to enable definitions that appear in regulatory documents, that do not have an exact match within the data dictionary to be matched to similar definitions within the data dictionary.

5.3.2 Structural Description

The reconciliation service will be implemented as a single component.

5.3.3 Behavioural Description

The interactions between the reconciliation component and other ACCORD components are identical to that of the data dictionary repository, with the reconciliation services used in place of direct usage of the data dictionary repository when reconciliation is required.

5.3.4 Component Implementation

The Data Dictionary Reconciliation Service will be implemented in accordance with the W3C Reconciliation Service API v0.2. It will make use of the following matching techniques:

- LLM (Large Language Model) matching.
- String matching.
- Fuzzy matching.
- Similarity matching.

5.4 Formalized Building Codes and Rules Repository

This section will document the repository of formalised building codes and rules repository component of the ACCORD Cloud Architecture.

5.4.1 Description and Objective

The purpose of this component is to provide a single repository of machine-operable building codes and rules within the ACCORD framework. This component, together with the Building Codes and Rules API will:

a) Provide access for all other components to the digitised regulations in a suitable machine-readable format.
b) Provide the ability for digitised regulations to be created/updated/deleted by appropriate software tools, such as the rule formalisation tool.

5.4.2 Structural Description

The structure of the formalized building codes component is formed solely from a deployment of GraphDB. The reason for this selection is described below. The GraphDB component is utilised to store digitised construction regulations that have been produced through the ACCORD digitisation approach described in Section 0. All access to the GraphDB instance is via the Regulations API and all components will access the GraphDB via this API. This is illustrated in *Figure* 9.

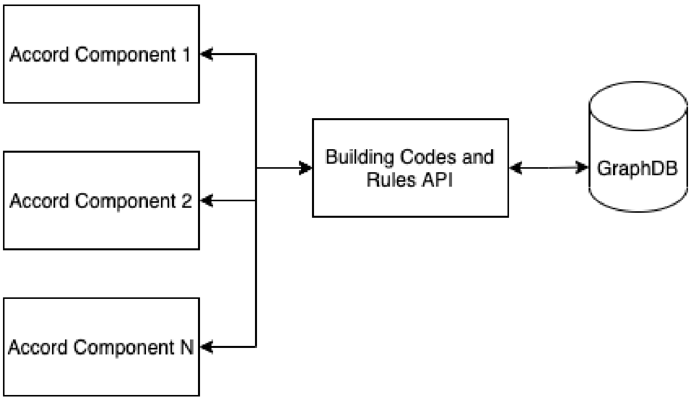


Figure 9. Formalised Building Codes and Rule Component Structure.

5.4.3 Behavioural Description

The behavioural aspects of the Formalised Building Codes and Rules component are covered within Section 5.4.3 Behavioural Description. This is because the interactions between all other ACCORD components and the Formalised Building Codes and Rule component are done via the Regulation API and no other ACCORD components will access this component directly.

5.4.4 Used Technologies

As mentioned, previous, this component will utilise the GraphDB Semantic Graph Database provided by Ontotext. The rationale for selecting this software is:

1. The software developers, Ontotext are project partners so can provide support for development.
2. GraphDB has a free version, enabling exploitation outside of the project without licensing costs.
3. GraphDB support multiple open standards, providing compatibility with the ACCORD rule formalisation approach as well as preventing vendor lock in.

5.4.5 Component Implementation

Given this component is utilising an off the shelf solution, the pathway to implementation is relatively straightforward. There are two key steps needed to implement this component:

1. Deployment of an instance of the GraphDB software in the ACCORD Cloud Environment.
2. Development of the Regulation API to provide an interface between GraphDB and other components (this is described in more detail in Section 5.12).

5.5 IDS Repository

This section will document the IDS repository component of the ACCORD cloud architecture.

5.5.1 Description and Objective

The IDS repository will be a service that enables other components of the ACCORD Cloud Architecture to retrieve a buildingSMART IDS (Information Delivery Standard) file related to a given building code. The IDS file may have been autogenerated utilising the IDS generation tool (described in Section 5.6 IDS Generation Tool) or may have been manually created.

5.5.2 Structural Description

The IDS repository will consist of a single component that is able to serve a list of IDS files available and provide a download of a single IDS file.

5.5.3 Behavioural Description

The IDS repository will primarily interaction with the model and data requirements validation service (described in Section 5.7 Model & Data Requirement Validation) and with the IDS generation tool. This is illustrated in Figure 10.

5.5.4 Used Technologies

As described previous the purpose of this component is to allow the storage and retrieve of IDS files. Thus, this component will utilise standard web server technologies exposed via a REST API.

5.5.5 Component Implementation

Given that an off the shelf web serving solution will be utilised to deliver this component, there are no implementation specific concerns.

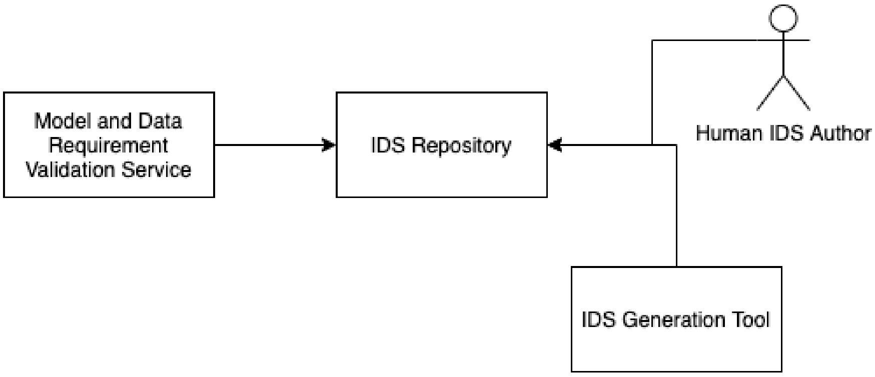


Figure 10. IDS Repository Interactions.

5.6 IDS Generation Tool

This section will document the IDS generation tool of the ACCORD Cloud Architecture.

5.6.1 Description and Objective

The IDS generation tool will be an experimental component develop to automatically generate IDS from the formalized building codes and rules developed within the ACCORD project. This will be useful as it will provide an automated method for generating IDS files, removing the need for manual development of these by a person.

5.6.2 Structural Description

The IDS generation tool will consist of a single component that will, by interfacing with the Building Codes and Rules Repository and the Data Dictionary Repository produce an IDS specification.

5.6.3 Behavioural Description

The IDS generation tool, as mentioned previous, will retrieve data for a given building code from the Building Codes and Rules Repository, the definitions used by the building code from the data dictionary repository. It will then produce an IDS and store it in the IDS repository. This is illustrated in Figure 11.

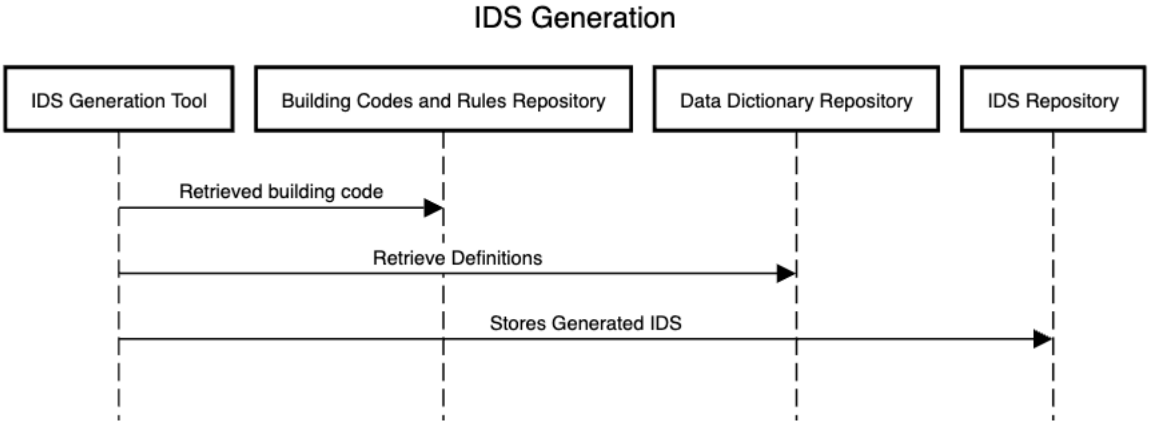


Figure 11. IDS Generation Sequence Diagram.

5.6.4 Used Technologies

The primary technologies utilised in this component will be the formalized regulations produced by the ACCORD project, and the IDS standard itself provided by buildingSMART.

5.6.5 Component Implementation

This will be a custom implemented component. However, existing open-source libraries to parse the IDS format will be utilised where feasible.

5.7 Model & Data Requirement Validation

This section will document the Model & Data Requirement Validation component of the ACCORD Cloud Architecture.

5.7.1 Description and Objective

This component will be responsible for validation of data and models that are used within the ACCORD cloud architecture. More specifically it will fill three distinct use cases:

Verification of IFC Models: Will ensure any IFC models provided to the ACCORD Cloud Architecture are valid IFC models with regards to one of buildingSMART's published schemas. This is required because any invalid models will not be able to be parsed and understood by ACCORD components.

Validation of IFC Models against bSDD: Will ensure that IFC models provided to the ACCORD Cloud Architectureuse the terminology as described in a bSDD. This is required to ensure that sufficient information is present in the IFC file to allow compliance checking against it to take place.

Validation of IFC Models against IDS: Will ensure that IFC models provided to the ACCORD Cloud Architecture meet the requirements of the required IDS files. This is required to ensure that sufficient information is present in the IFC file to allow compliance checking against it to take place.

Validation of GIS Data: Will ensure that any submitted GIS data conforms to agreed GIS specifications and profiles.

5.7.2 Structural Description

Structurally the component will be divided as per the use cases described previously. This is shown in
Figure 12.

It should be noted that two other components are required by the IFC Validation and the GIS validation functionality. For IFC Validation the ACCORD IDS repository will be utilised to retrieve the IDS that the IFC model is required to be checked against. For GIS validation the profile definitions of the GIS model will be retrieve from OGC Rainbow (an external service).

5.7.3 Behavioural Description

The behaviour of the model and data requirement validation service is illustrated in the sequence diagram in *Figure 13*. This illustrates how, when a model is first submitted to the ACCORD Cloud Architecture, it is passed to the model and data requirement service by the process engine. Depending on its type (IFC or GIS) it will then be passed to the appropriate elements of the component to be validated.

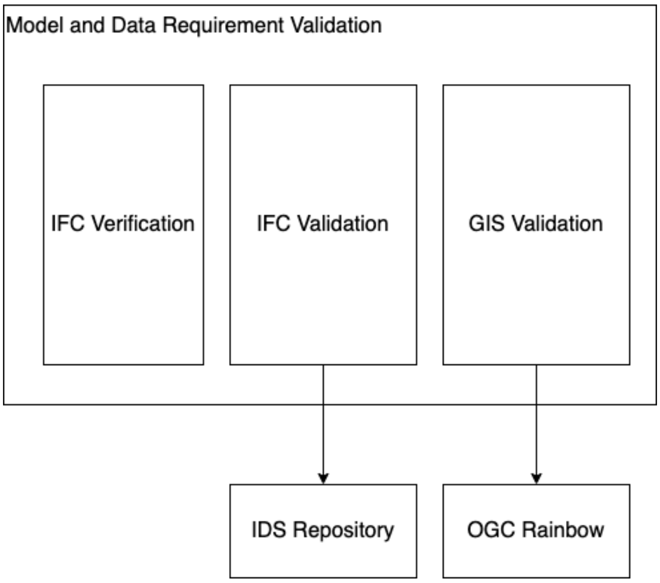


Figure 12. Model and Data Requirement Validation Component Structure.

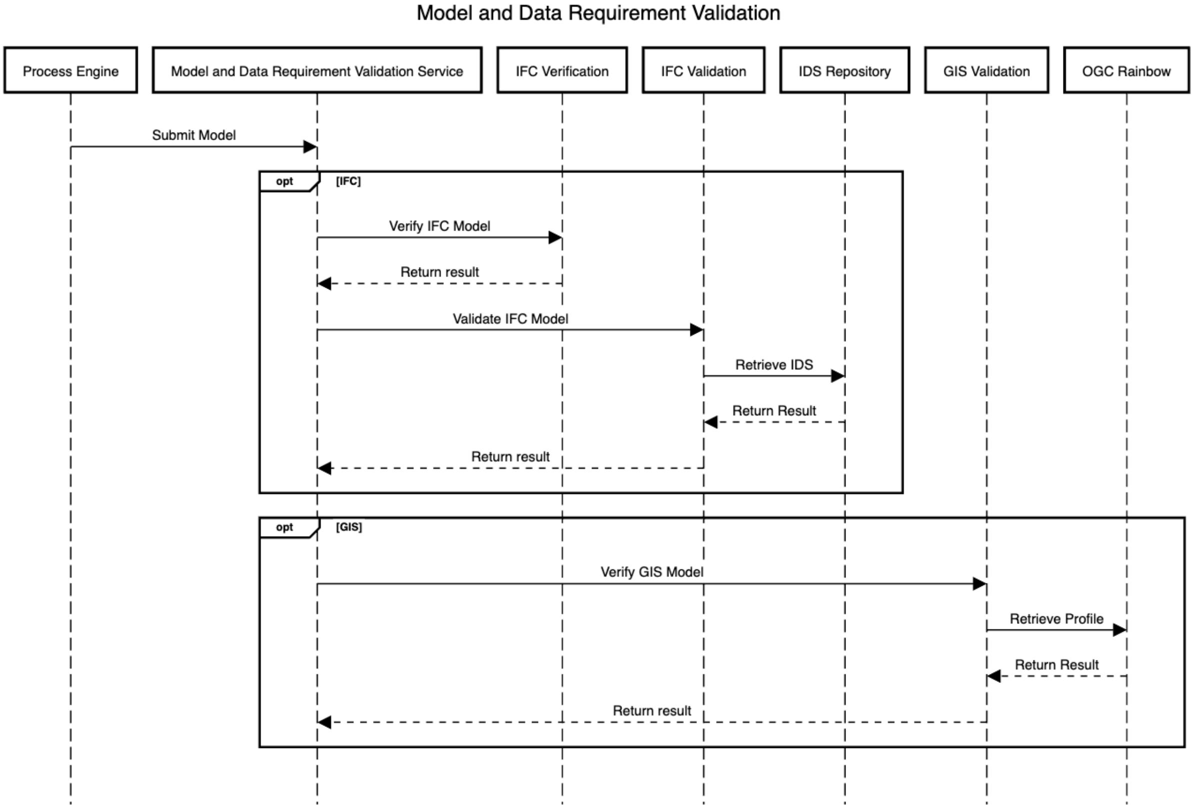


Figure 13. Model and Data Requirement Validation Sequence Diagram.

5.7.4 Used Technologies

Key technologies utilised by this component include:

- GIS profile definitions, sourced for the OGC Rainbow service. These are provided as an OWL/RDF based ontology with sub profiles and vocabularies.
- For the purposes of executing GIS model checking the ontologies retrieved from OGC Rainbow will be used to generate SHACLs and JSON-schemas that can be used for model validation.
- The IDS and IFC standards from buildingSMART will be utilised for verification and validation of IFC models.

In addition, three existing software implementations will be used to develop this component, these are described in the following section.

5.7.5 Component Implementation

Three existing software tools will be integrated to develop this component. These are:

- IFC Verification, provided by buildingSMART ifc verification software, which checks for both IFC validity and bSDD compliance.
- IDS Validation, provided by the Future Insight Clearly.BIM software.
- GIS validation provided by OGC validator software.

5.8 Process Execution

This section will document the Process Execution component of the ACCORD Cloud Architecture.

5.8.1 Description and Objective

The Process Execution component acts as the process backbone of the permitting process. Every permit request follows a predefined process. That process is often prescribed by local law, but in general consists of steps like:

1. An applicant applies for a permit.
2. The applicant uploads relevant documentation, including models.
3. The local permitting authority reviews the permit request and evaluates it against the relevant laws and regulation.
4. The local permitting authority comes to a decision, the permit is issued, additional information or changes in the plans is required, or the permit is denied.

This component will be the coordinating behind implementing the ACCORD digitised processes for building permitting. It will thus integrate and coordinate the other components of the ACCORD Cloud Architecture, enabling the execution of building permitting process flows specific to the local process required by permitting authorities.

5.8.2 Structural Description

The process execution component will consist of several subcomponents:

A process engine, that embeds permitting processes modelling using an open standard (i.e. BPMN) enabling it to be flexible and support various processes (as these most likely will differ between countries, type of permit, etc.). This component will also be responsible for managing required information in each process step, as well as controlling when the process moves to the next step (i.e. on model upload/validation etc.)

A frontend to enable human users to interact, complete and manage tasks executing within the process engine.

5.8.3 Behavioural Description

The process engine will interact with the following components: (1) Model & Data Requirement Validation, (2) Microservice Orchestration, (3) Data Storage and (4) Compliance checking microservices. These interactions are depicted in the simplified sequence diagram shown in *Figure* 14. It should be noted that this represents a generic process designed to illustrate the interaction with other components. The specific processes for ACCORD demonstration pilots will be more complex.

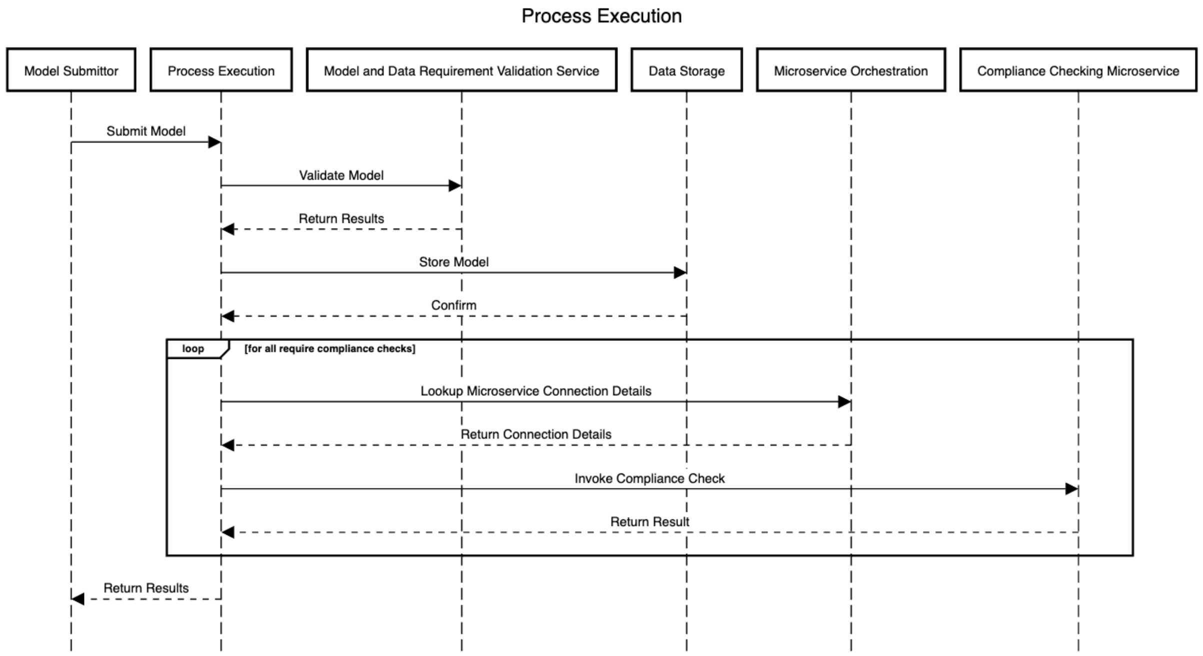

Process Execution
Model Submittor
Process Execution
Model and Data Requirement Validation Service
Data Storage
Microservice Orchestration
Compliance Checking Microservice
Submit Model
Validate Model
Return Results
Store Model
Confirm
loop
[for all require compliance checks]
Lookup Microservice Connection Details
Return Connection Details
Invoke Compliance Check
Return Result
Return Results

Figure 14. Process Execution Sequence Diagram.

5.8.4 Used Technologies

Several existing open technologies will be utilised in the development of the process engine. These are listed below:

- Business Process Modelling Notation (BPMN) for process flow descriptions.
- IFC(openCDE) and OGC standards for transmitting BIM/GIS models (showing the model, passing it on to checking services).
- BCF for serving and exchanging result findings.

5.8.5 Component Implementation

It is anticipated that this component will be developed as a modification to an existing open-source process engine. The exact open-source tool is still to be determined.

5.9 Data Storage

This section will document the Data Storage component of the ACCORD Cloud Architecture.

5.9.1 Description and Objective

The data storage component forms the central location where data that is used in the cloud-based permitting service is stored. The 'single store multiple use' concept is important here. Information should be stored only once, to create a single source of truth, and be useable and used by multiple applications. The primary use of the data storage component will be to store BIM and GIS models, along with any metadata relevant to these models.

Two methods will be utilised to provide this storage: (1) standard static file storage, and (2) semantic storage. These will both be described in the following section.

5.9.2 Structural Description

The data storage component will consist of two sub-components: (1) static file storage and (2) semantic storage.

Static File Storage: This component will store models as a single file. Access to these files will be provided via the OpenCDE API.

Semantic Data Storage: This storage component will allow retrieval of only parts of a model. i.e. 'give me all doors' or 'give me all objects related to fire safety'. This is an experimental component and will be newly developed by the ACCORD project. There is no open standard for retrieving only specific parts of a model yet. Some applications do offer such functionality, but use a proprietary language or API call, due to this lack of standardization. Thus, ACCORD must adopt or develop its own approach to solve this problem.

5.9.3 Behavioural Description

The overall behaviour of the data storage component is illustrated in *Figure* 15. This shows how the data storage component will utilise both semantic and file-based storage to answer queries.

As is shown, the primary components that will interact with the data storage components will be the compliance checking microservices. These interactions are illustrated in *Figure 18* and will be described in more detail in Section 5.11 Compliance Checking Microservice(s).

5.9.4 Used Technologies

The data storage component will utilise known standards for data presentation, data storage, and API specifications, namely:

- **IFC** for exchanging and storing BIM models.
- **RDF**: Resource Description Framework for providing a semantic representation of models.
- **SPARQL**: the standard query language and protocol for Linked Open Data on the web or for RDF triplestores. SPARQL enables users to query information from databases or any data source that can be mapped to RDF.
- **OpenAPI** and/or **GraphQL** for retrieving parts of a model based on query.
- **OpenCDE** for exchanging IFC files.
- A to be developed approach for exchanging parts of an IFC model.

- **Ontotext GraphDB**: An RDF Database for Knowledge Graphs. A W3C standards compliant RDF graph database and one of the few triple stores that can perform semantic inferencing at scale, allowing users to derive new semantic facts from existing facts.
- **Ontotext Platform**: provides GraphQL API, security and search service over Knowledge Graphs.

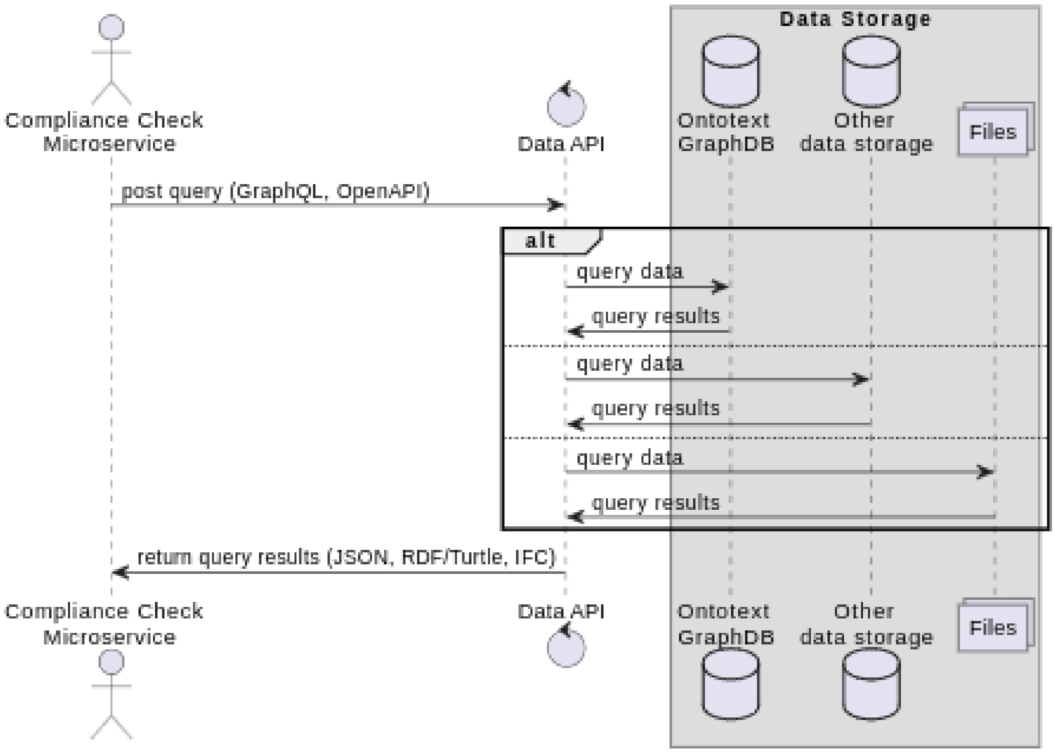


Figure 15. Sequence Diagram: Compliance Checking Microservice and Data API Interactions

5.9.5 Component Implementation

The implementation of both data storage sub-components will now be described in turn.

Static File Storage implementation: This will be implemented using Future Insight Clearly.BIM, a pre-existing software tool that can be utilised to store and serve models via the OpenCDE API.

Semantic Data Storage implementation: This component will be newly developed as part of the ACCORD project. It will be based on Ontotext GraphDB. This will be a hybrid storage mixing semantic and non-semantic (for geometry) storage of IFC models. As this is an experimental component various approaches will be trailed to enable querying of this dataset:

- Use of a domain-specific quary language
- Use of GraphQL.

5.10 Orchestrating Microservices

This section will document the orchestrating microservices component of the ACCORD Cloud Architecture.

5.10.1 Description and Objective

The function of the orchestrating microservices component is to identify, manage, and monitor the compliance checking microservices. This is required since the ACCORD Cloud Architecture will consist of multiple microservices, each of which may well have several different instances running.

This creates the need for an authoritative data source that other components of the ACCORD Cloud Architecture can utilise to identify and select a microservice, based on the compliance check that needs to be performed and retrieve the connectivity information required to utilise it.

5.10.2 Structural Description

This component will provide the following key elements of functionality:

1. Ability to register microservices and their functionality.
2. Ability to monitor microservices and their instances.
3. Provision of a querying API to allow other comments to retrieve information about microservices, their features and their connectivity information.

This will be delivered through two key sub-components:

1. **Functionality register:** A database of microservice functionality, providing a mapping between compliance checks and the microservice that can execute them.
2. **Microservice Registration and Tracking:** A component that tracks and monitors the status of microservices, their status and the required endpoint/connectivity information.

5.10.3 Behavioural Description

The behaviour of the Orchestrating Microservices component is shown in the sequence diagram in *Figure 16*.

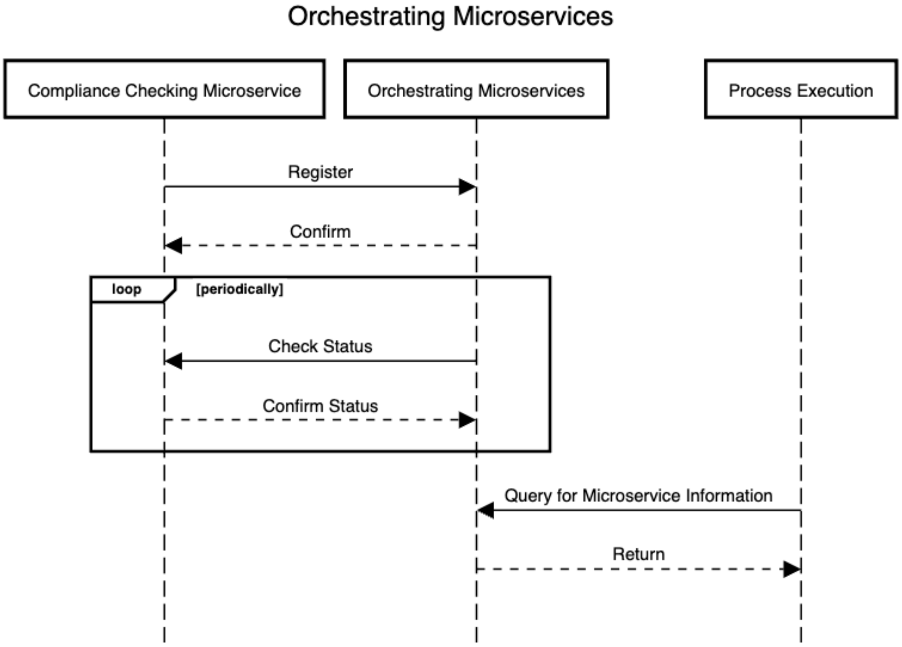

Orchestrating Microservices
Compliance Checking Microservice
Orchestrating Microservices
Process Execution
Register
Confirm
loop
[periodically]
Check Status
Confirm Status
Query for Microservice Information
Return

Figure 16. Orchestrating Microservices Sequence Diagram.

5.10.4 Used Technologies

To implement this component ACCORD will select from an existing microservices management implementation. Under consideration are:

- Netflix Eureka
- Apache Zookeeper.
- Hashicorp Consul.

One of these existing software implementations will be used to provide microservice registration and tracking element. For the functionality register, either the open source implementation selected for microservice registration and tracking will be extended, or a custom implementation will be developed.

5.10.5 Component Implementation

An existing microservice management tool will be selected and utilise to provide the Orchestrating Microservices component for the ACCORD Cloud Architecture.

5.11 Compliance Checking Microservice(s)

This section will document the compliance checking microservices that form a key element of the ACCORD Cloud Architecture.

5.11.1 Description and Objective

Within the ACCORD Cloud Architecture a compliance checking microservice is described as: a component that performs the actual concrete compliance checks within the ACCORD system.

There will be many compliance checking services within the actual developed ACCORD Cloud Architecture. The concept of this approach is to enable the ACCORD Cloud Architecture to integrate compliance checking services in a dynamic fashion allowing existing tools to be integrated without the need for extensive reworking. Furthermore, it enables the more rapid development of new services that solve a single defined problem.

This section will initially describe compliance checking services as an abstract concept, with the specific microservices that will be implemented within the ACCORD project described in 5.11.4.

5.11.2 Structural Description

A structural description of an ACCORD compliance checking microservice is shown in Figure 17.

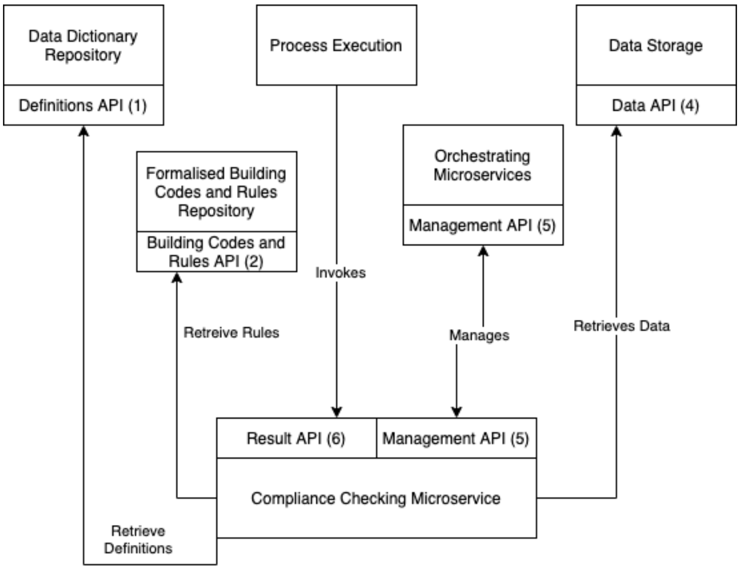


Figure 17. Compliance Checking Microservice Structure.

5.11.3 Behavioural Description

The behaviour of an ACCORD compliance checking microservice and how it interacts with other components is illustrated in a sequence diagram in Figure 18.

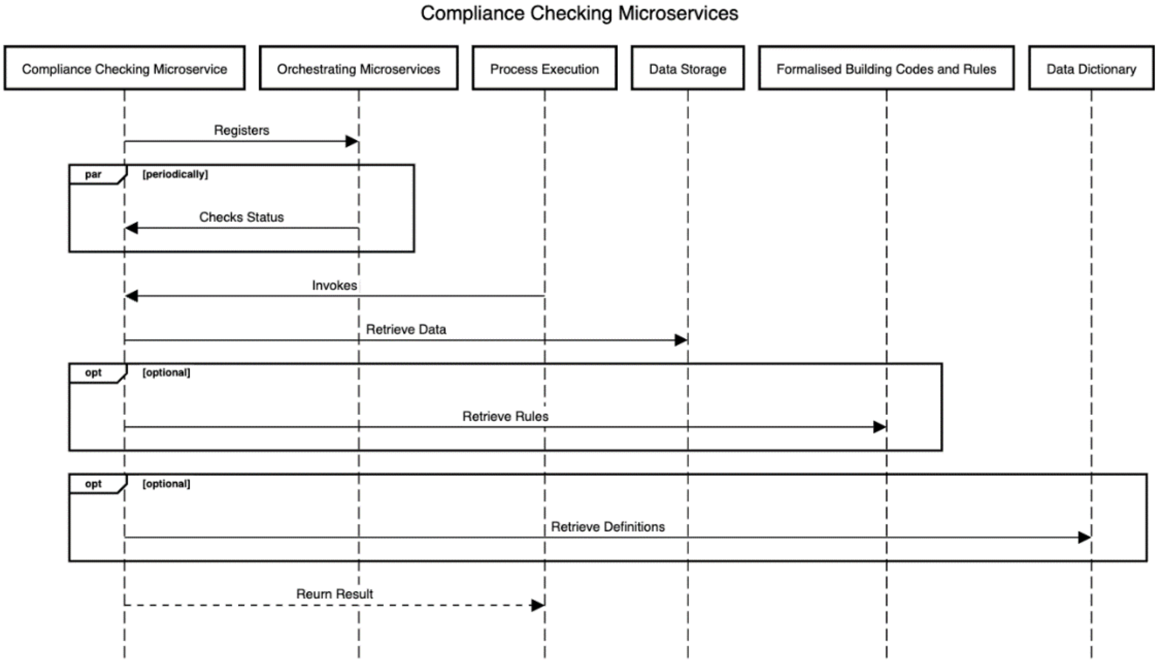


Figure 18. Compliance Checking Microservice Sequence Diagram.

5.11.4 Used Technologies

The primary technologies leveraged on will be the existing microservices to be implemented. In total

8 microservices will be implemented. Some already exist and some require full implementation. These microservices are:

- **Solibri Office:** Solibri Office is a BIM environment that support geometric checking. It provides IFC based geometric checking against rules. These rules can be either built into the Solibri software, specified using a ruleset editor, or manually developed using APIs.
- **Future Insight Clearly.BIM:** Clearly.BIM is software to view, share and query BIM models and execute IDS and compliance checks. It supports a variety of querying and rule-based functionality around building permitting, currently mostly used for Dutch, Estonian and Finnish building codes. It however offers a highly configurable check language that can be used in any context. Clearly.BIM is fully cloud based and all functionalities are available through API (both GraphQL and OpenAPI).
- **LCA Finland:** A custom developed microservice that will perform LCA and environmental emission calculation of a IFC model based on the Finnish environmental standards.
- **LCA Germany:** A custom developed microservice that will perform LCA and environmental emission calculation of a IFC model based on the German environmental standards.
- **Eurocode Compliance Checking:** A specialist tool built on open-source technology, is used to manage the structural analysis of IFC building models. This takes input from Finite Element Method (FEM), which will first enrich an IFC model with the associated response (internal forces, displacements, reactions) for each related IFC entity. This step will be performed prior to compliance checking submission. This microservices will then analyse this data to apply structural design evaluations, and typically verifications according to an appropriate design standard.
- **Urban Regulations Checking:** A microservice to computing various geometric checks required to undertake urban planning regulations checking. This will consist of two elements, (1) validation the geolocation of the building relative to the requirements specified within the town hall information system and (2) to conduct geometric checks based on the specific geolocation of the building.
- **Land use building compliance checking:** Service providing compliance checking of IFC building models according to land use requirements.
- **Type Approval building compliance checking:** Service providing compliance checking of IFC building models according to requirements for the type of approval of timber construction systems.

5.11.5 Component Implementation

Some of the compliance checking microservices within the ACCORD Cloud Architecture already exist and simply need integrating with the ACCORD APIs. Specifically, these are the Result API Data and the Management API. The list below documents the current state of each microservice and the tasks that must be conducted to integrate with the ACCORD Architecture:

- **Solibri Office:** Software tool already exists. The following tasks require completion; (a) developing of ACCORD specific geometric checks, (b) integration of geometric checks with Result API, (c) integration with Data API, (d) integration with management API.
- **Future Insight Clearly.BIM:** Software tool already exists. The following tasks require completion; (a) developing of ACCORD specific rules, (b) integration of rules with Result API, (c) integration with Data API, (d) integration with management API. If the ACCORD project results in a (proposed) standard check language, support for this standard needs to be implemented.
- **LCA Finland:** Microservice does not currently exist and will be implemented. This will require integration with: (a) Result API, (b) Data API, (c) management API.
- **LCA Germany:** Microservice does not currently exist and will be implemented. This will require integration with: (a) Result API, (b) Data API, (c) management API.

- **Eurocode Compliance Checking:** Software tool already exists. The following tasks require completion; (a) integration with Result API, (b) integration with Data API, (c) integration with management API.
- **Urban Regulations:** Microservice does not currently exist and will be implemented. This will require integration with: (a) Result API, (b) Data API, (c) management API.
- **Land Use Building Compliance Checking**: Microservice does not currently exist and will be implemented. This will require integration with: (a) Result API, (b) Data API, (c) management API.
- **Type Approval Building Compliance Checking**: Microservice does not currently exist and will be implemented. This will require integration with: (a) Result API, (b) Data API, (c) management API.

5.12 Information Services

This section will provide an overview of the information services utilised by the ACCORD framework and their associated APIs.

5.12.1 Description and Objective

Within the ACCORD Cloud Architecture an information service is defined as a source of static, or slowly changing, information, external to the ACCORD system, that is utilised by one or more microservices. Five use cases for this have been identified:

1. **Land Use OpenAPI for Features:** Retrieval of XPlanung land use models from an Open GIS-compliant OpenAPI for Features (OAF) web service. The service provides graphic and textual regulations of development plans in force and in the process of being drawn up in the Tegel-Projekt GmbH project area in three data schemas and different encodings: XPlanung (binding standard in Germany), INSPIRE Planned Land Use (PLU) and in CityGML data models and GML, JSON-FG (generated based on XPlanung- and INSPIRE PLU GML) and CityJSON encoding.
2. **Urban Data Profile Validation:** Retrieval of land use profiles for validation of urban regulations.
3. **Material Emission Database:** Retrieval of material emission characteristics from databases (such as https://co2data.fi/) by the LCA Microservices.

The motivations for treating these information services separately from the microservices that utilise them are; (a) the standard service implementation practice of separating the component that performs the process on the data, from the data source itself, (b) organisationally, many of these data sources are owned and maintained by third parties, this separating them from the ACCORD Cloud Architecture enables the maintaining of the organisational divide between the data provider and data processor.

5.12.2 Structural Description

Figure 19 illustrates the structure of two information services (as an example). As can be seen from the figure there is a simple 1:1 relationship between the external information service and the microservice within the ACCORD Cloud Architecture.

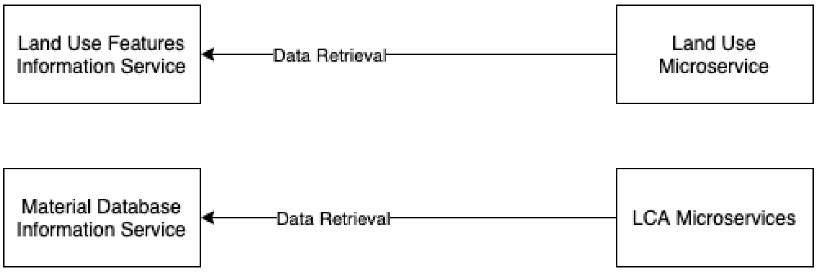

Land Use Features Information Service
Data Retrieval
Land Use Microservice
Material Database Information Service
Data Retrieval
LCA Microservices

Figure 19. Information Services Structural Diagram.

5.12.3 Behavioural Description

Figure 20 illustrates the behaviour of information services relative to the microservice that performs to required calculations. It should be noted that there is no unified Information Services API currently available, this is since each individual information service will currently have its own API and given the fact that each information service is operated by a third party outside of the project ACCORD cannot require API changes to these external data sources. However, a best practice suggested API will be proposed as part of Task 4.4.

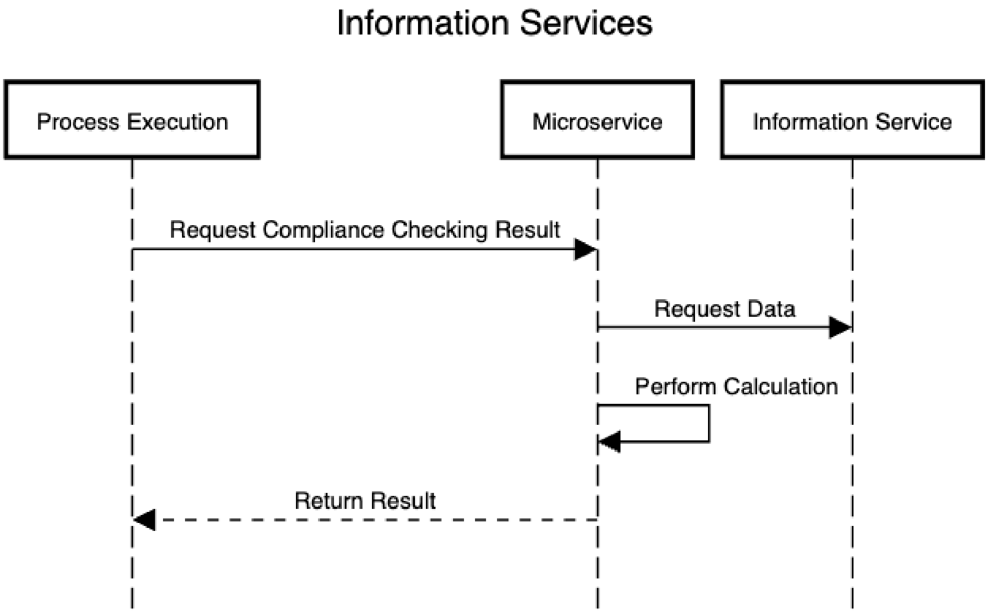


Figure 20. Information Services Sequence Diagram.

5.12.4 Used Technologies

Current APIs that exist in this space and, for pragmatic reasons, ACCORD must integrate with currently utilise the following approaches:

- RESTful architectural style.

- Any authentication/authorisation requirements are handled using the OAUTH2 protocol and JSON Web Tokens.

However, as part of the ACCORD development work, as mentioned previously, a best practice suggested API will be proposed. This propose API will utilise the OpenAPI standard will be utilised to document the structure of the API.

5.12.5 Component Implementation

The proposed API related to the information services will be implemented using a standard software engineering approach. Firstly, the API will be designed and documented using an OpenAPI specification.

The practical integration with existing APIs will be conducted by firstly analysing the existing APIs that are available and producing an implementation that allows each individual microservice to retrieve data from the existing API.

5.13 APIs

This section will describe the APIs utilised by the ACCORD Cloud Architecture.

5.13.1 Description and Objective

The APIs utilised by the ACCORD Cloud Architecture enable the communication between components. These APIs are a mixture of the adoption of the use of existing standardised APIs and the development of custom APIs where no standard API exists.

The 7 APIs utilised within he ACCORD Cloud Architecture are:

1. **Definitions API:** The definitions API existing to provide an interface where other components can retrieve data dictionary from the data dictionary repository.
2. **Building Codes and Rules API:** The Building Codes and Rules API exists to provide an interface between components of the ACCORD framework and the Formalized Building Codes and Rules component and the IDS repository component. Thus, the purpose of this API is to provide a consistent abstract interface for retrieving and uploading digitised construction regulations to the GraphDB used by the Formalized Building Codes and Rules component. It also supports the retrieval of IDS files from the IDS repository.
3. **Information Services APIs**: This API represents the collection of APIs utilised by the information services described in Section 5.12 Information Services. This allows the Compliance Checking microservices to retrieve information, as required, from these information services.
4. **Data API**: The objective of Data API is to provide data to be used in Compliance Checking Microservices upon request. Data API provides interfaces to put various data into the Data Storage and makes these data available to Compliance Checking Microservices, so they can receive the data from the Data Storage in a particular format.
5. **Management API**: This API provides the interface between the orchestrating microservices component and the compliance checking microservices. Allow the exchange of management and connectivity information between these components.
6. **Results API:** The Result API will provide the interface to enable the exchange of the results of compliance checking processes between the ACCORD Cloud Architecture, and an individual compliance checking microservices. This will allow the process execution

component to trigger a given compliance check on a compliance checking microservice and then subsequently receive the results of that compliance check.

7. **Reconciliation API**: This API allows components to query the Reconciliation Service.

Only the APIs 2,4 and 6 will be developed within the project. APIs 1,3,5 and 7 will utilise existing standardised APIs.

5.13.2 Behavioural Description

This section will describe the interactions between other components of the ACCORD Cloud Architecture and the ACCORD APIs. The components that will interact with each API is shown in *Figure 21*.

Most of the APIs exhibit a simple request and response pattern. However, the Result API, exhibits a different pattern. The Result API acts as the interface to enable the exchange of the results of compliance checking processes between the ACCORD Cloud Architecture, and an individual compliance checking microservices. These interactions, and the components with which this API will interact is illustrated in *Figure 22*. This includes two separate use cases; (1) where a result can be provided immediately to the compliance checking process and (2) where a result will take some time to compute, and the calling component must retrieve the result later once the process is complete.

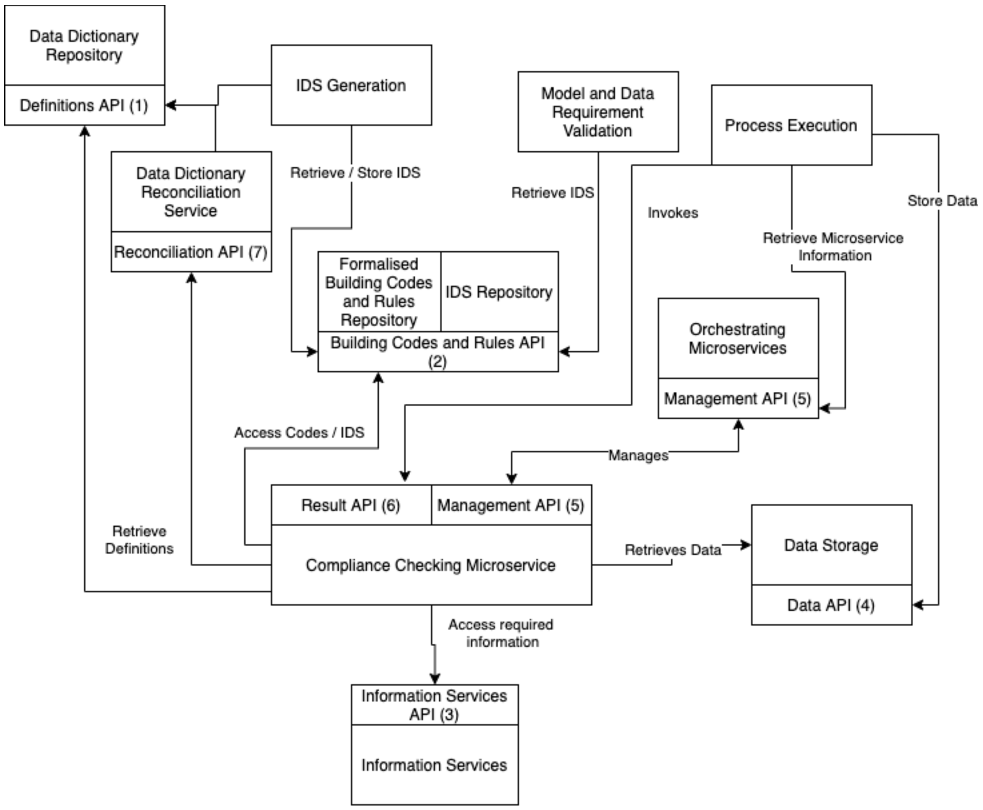

Data Dictionary Repository
Definitions API (1)
IDS Generation
Model and Data Requirement Validation
Process Execution
Data Dictionary Reconciliation Service
Reconciliation API (7)
Retrieve / Store IDS
Retrieve IDS
Invokes
Store Data
Retrieve Microservice Information
Formalised Building Codes and Rules Repository
IDS Repository
Building Codes and Rules API (2)
Orchestrating Microservices
Management API (5)
Access Codes / IDS
Manages
Result API (6)
Management API (5)
Retrieve Definitions
Compliance Checking Microservice
Retrieves Data
Data Storage
Data API (4)
Access required information
Information Services API (3)
Information Services

Figure 21. Result API Interactions.

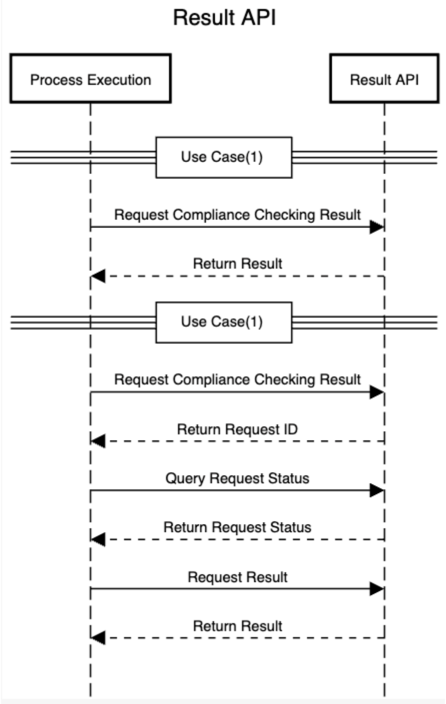


Figure 22. Result API Sequence Diagram.

5.13.3 Used Technologies

This section will document the existing standardised APIs that are being utilised for the 4 APIs (1,3,5 and 7) that are adopting existing APIs:

- **Definitions API:** This API will utilise the existing buildingSMART data dictionary API.
- **Information Services APIs**: This will utilise the APIs from the specific information services considered.
- **Management API**: This will utilise the existing API from the selected open source microservice management software selected.
- **Reconciliation API**: This will utilise the existing W3C Reconciliation Service API.

5.13.4 Component Implementation

This section will document the high-level implementation details for the three APIs that are being developed within the project (2,4 and 6). For each of these a common set of implementation principles will be followed.

1. The APIs will be implemented using the RESTful architectural style.
2. The OpenAPI standard will be utilised to document the structure of the API, which will be agreed across the project team.
3. Any authentication/authorisation requirements will be handled using the OAUTH2 protocol and JSON Web Tokens.
4. For the **Building Codes and Rules API**: Will utilise YAML, IDS and JSON.
5. For the **Data API:** Will utilise the following existing technologies as part of its implementation; (1) GraphQL, (2) IFC, (3) RDF/Turtle, (4) JSON and (5) OpenCDE.

6. For the **Results API**: Will integrate technologies such as the BIM collaboration format (BCF) including the BCF API developed by buildingSMART international.

5.14 Demo-Specific Alignment

This section will document the mapping of the ACCORD Cloud Architecture components to the ACCORD demo use cases. It will showcase what elements of the ACCORD cloud architecture will be deployed at each demo and in what context.

To organise this, there will be two modes of deployment of a given component: (1) Normal, where a component will be used to deliver the demo and (2) experimental, where a more experimental component developed by the ACCORD project will be tested as well, to validate its functionality and usability.

Table 16 describes this mapping. The following numbering represents the mappings in the table, with demo cases illustrated as columns and the components as rows. In the table, blue indicated a components will be used as a normal component within the demo, orange will indicate a component being used in an experimental context.

Demo Use Cases:

- DE - Germany
    1. Land Use Permitting
    2. Environmental Compliance
    3. Type Approval for Timber Construction Systems
- ES - Spain - Urban Regulations
- UK - Steel Modular Housing
- FI - Finland
    1. Population Information System
    2. Accessibility
    3. CO2
    4. Operational Safety
- EE - Estonia
    1. Accessibility
    2. Fire Safety
    3. Education (School/ Kindergarden).

ACCORD Cloud Architecture Components:

1. Rule Formalization
2. Data Dictionary
    a. Data Dictionary Repository
    b. Data Dictionary Reconciliation Service
3. Formalized Building Codes and Rules Repository
4. Information Requirements
    a. IDS Repository
    b. IDS Generation Tool
5. Cloud-based Building Permit Services
6. Model- and Data Requirement Validation
7. Process Execution
8. Data Storage
    a. File based data storage.
    b. Semantic Data Storage
9. Orchestrating Microservices
10. Compliance Checking Microservices

    (1) Solibri Office
    (2) Future Insight Clearly.BIM
    (3) LCA Finland
    (4) LCA Germany
    (5) Eurocode Compliance Checking
    (6) Urban Regulations Checking
    (7) Land Use Building Compliance Checking.
    (8) Type Approval Building Compliance Checking.

11. Information Services
    (1) Land Use OpenAPI for Features
    (2) Urban Data Profile Validation
    (3) Material Emission Database.

	DE 1	DE 2	DE 3	ES	UK	FI 1	FI 2	FI 3	FI 4	EE 1	EE 2	EE 3
No	**ACCORD Cloud Architecture Components**											
1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
2a	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
2b		🟦						🟦				
3	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
4a	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
4b	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
5	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
6	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
7	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
8a	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
8b	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
9	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
10	**Compliance Checking Microservices**											
(1)							▒		▒	▒	▒	▒
(2)							▒		▒	▒	▒	▒
(3)								▒				
(4)		▒										
(5)					▒							
(6)				▒								
(7)	▒		▒									
(8)			▒									
11	**Information Services**											
(1)	▒											
(2)				▒								
(3)		▒						▒				
12	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Table 16. Aligning ACCORD Cloud Architecture components to demo use cases.

5.15 Conclusion

This section has documented 12 components and 7 APIs of the ACCORD Cloud Architecture and aligned those components to demo uses cases. The components are:

1. **Rule Formalization Tool –** The tool used by regulatory professionals to formalize building codes and rules into an executable format. Not specified in this deliverable as it is delivered

in WP2.

2. **Data Dictionary** – This component comprises subcomponents 2a and 2b:
    a. **Data Dictionary Repository** – A repository of data dictionaries used in the compliance checking process.
    b. **Data Dictionary Reconciliation Service** – A service designed to perform reconciliation and matching on data dictionary stored within the data dictionary repository.
3. **Formalized Building Codes and Rules Repository** – A repository of formalized building codes and rules.
4. **Information Requirements –** This component comprises subcomponents 4a and 4b:
    a. **IDS Repository** – A repository of IDS files used in the compliance checking process.
    b. **IDS Generation Tool** – An experimental tool designed to generate IDS files from the building codes and rules stored in the building codes and rules repository.
5. **Cloud-based Building Permit Services** - This component is specified by its subcomponents number 6, 7, 8 and 9.
6. **Model and Data Requirement Validation** – The component responsible for validating models submitting to the ACCORD cloud architecture.
7. **Process Execution** – The central coordinating component of the ACCORD cloud architecture which will execute and monitor the overall process of compliance checking.
8. **Data Storage** – The central data-storage component that will provide a repository of semantic and static data that is used in cloud-based building permit services.
9. **Orchestrating Microservices** – The component that will identify, monitor and manage and compliance checking microservices.
10. **Compliance Checking Microservices** – The set of 8 microservices that will perform the work of compliance checking.
11. **Information Services** – The set of 3 information services that the compliance checking microservices will utilise to retrieve information necessary for compliance checking.
12. **APIs** - A set of 7 APIs either re-used or to be created for components of the ACCORD cloud architecture to communicate.

6 Conclusions

This deliverable has documented results of ACCORD's Tasks 4.1 and 4.2 presenting both the ACCORD Technical Requirements and the developed ACCORD Cloud Architecture.

Specifically, the outputs of the work are:

- A list of 134 technical requirements which will form the basis for ACCORD solution developments and technical implementations.
- The developed ACCORD Cloud Architecture comprising 12 components and subcomponents, and 7 identified APIs that will be developed.
- The definition of a set of 8 compliance checking microservices and 3 information services will form part of the overall ACCORD compliance checking approach.
- The defined demo-specific alignment, integrating the cloud architecture, consortium partners' and existing (micro-)services according to the specific needs of the demonstrator projects.

In the upcoming tasks of ACCORD WP4, these outputs will be key in guiding the implementing and integration of the ACCORD Cloud Architecture and the testing, validation in ACCORD demonstrators and quality assurance of developed solutions.

References

Atef, M. and Zulkernine, M., 2010. "Architectural design decisions for achieving reliable software systems". In: Architecting Critical Systems: First International Symposium, ISARCS 2010, Prague, Czech Republic, June 23-25, Proceedings. Springer Berlin Heidelberg.

Breivold, H. P., Crnkovic, I. and Magnus Larsson, 2012. "A systematic review of software architecture evolution research." Information and Software Technology, 54.1, p. 16-40.

Bruegge, B. and Dutoit, A. H., 2009. “Object-Oriented Software Engineering: Using UML, Patterns and Java”. 3rd Edition, Publisher: Prentice Hall, Upper Saddle River, New York.

Hofmeister, C., Nord, R. and Soni, D., 2000. Applied Software Architecture. Addison-Wesley Professional.

Linaker, J., et al., 2015. “Guidelines for conducting surveys in software engineering”, v.1.1. Lund University, Volume 50.

Strnad, C. F. and Schöning, H., 2023. “Datenplattformen, Datenräume und (Daten-)Ökosysteme - Einordnung und strategische Aspekte”. In: Weber, B., 2023. “Data Governance, Nachhaltige Geschäftsmodelle und Technologien im europäischen Rechtsrahmen”, Springer.

Zowghi, D. and Coulin, C., 2005. "Requirements elicitation: A survey of techniques, approaches, and tools”. In: Engineering and Managing Software Requirements, p. 19-46.

Appendices

Appendix 1. ACCORD Framework User Requirements (D1.2)

The table below outlines the complete list of user requirements elicited for the ACCORD Framework being reported in D1.2. With regard to the origin of each requirement, V represents requirements from the ACCORD vision, H for ACCORD high-level requirement, and the abbreviations G, S, U, F and E indicate a singular or additional demo-country of origin, that is Germany, Spain, UK, Finland and Estonia.

No.	User Requirement	Origin
1	Provide a platform to provide digitised building permitting processes.	V, S, F, U, E, G
2	Provide automated compliance checking.	V, S, F, U, E, G
3	Support both Geospatial and BIM(IFC) data input.	V, H, F, S, E, G
4	Provide sufficient customisation ability such that formally specified processes are generic enough to be scaled to the European level but flexible enough to allow the specific nature of each nation’s permitting processes to be considered.	V, H, S, F, U, E, G
5	Provide an intuitive method to allow regulation experts to digitise building codes/regulations and embed rules within them, without the need to write code.	V, H, S, F, U, E, G
6	Provide the ability to store a database of rules.	V, S, F, U, E, G
7	Provide the ability to extract the information requirements for digital building permitting and compliance processes and represent these as a standardised data schema using BuildingSMART standards (i.e., IDS).	V, H, S, F, U, E, G
8	Should be a dynamic system with the ability to add and removable modules.	V
9	Should support integration of data dictionaries to enable mappings between regulatory terms and data schemas.	V, H, F, S, E, U, G
10	Should be able to leverage emerging Artificial Intelligence techniques, such as semantic deep learning Natural Language Processing (NLP).	V, G
11	Should provide a set of microservices, with tools and solutions for digital permitting and automated compliance checking of buildings.	V, H, U, S, F, E, G
12	Should provide open standardized application programming interfaces and make use of open standards where applicable (i.e., from OGC, buildingSMART etc..)	V, H, G

No	User Requirement	Origin
13	The digitised format of building/codes regulations should be independent of any specific building modelling format.	H, G
14	Should support the use of classification systems	H, F, G
15	Should provide the formalization of concepts from building codes/regulations in a semantic form.	H
16	Provide the ability to pre-check for compliance prior to formal submission.	H, G
17	Provide the ability to link building permitting processes, applicable legislation and building data standards and provide audit abilities to track decisions.	H, G
18	Provide open access to limited data about building permitting assessments.	H, G
19	Provide a standardised submission process.	H, G
20	Should support and enable direct communication between the submitter and regulator.	H, F, S, E, G
21	Should provide suitable security models to differentiate between users to enable selection of an appropriate user to assess a given regulation.	H, F, S, E, G
22	Provide the generation of human readable and machine-readable (BCF) reporting based on submissions.	H, U, F, S, E, G
23	Should enable collection of suitable evidence to complement assessments	H
24	Should retain the ability for manual human input.	H, S, G
25	The ability to share compliance results with other relevant users.	U, F, S, E, G
26	Support the ability to perform model validation – checking that it contains the required information to perform compliance checking.	U, F, S, E, G
27	Support the ability to perform model verification - checking that a submission complies with relevant modelling schema.	U, F, S, E, G
28	Support integration of a Finite Element Analysis compliance checking microservice	U
29	Support providing reporting of results of model verification and validation.	U, F, S, G
30	Support upload of model files in an appropriate format	U, F, S, G
31	Provide a compliance checking microservice that can extract data from national level databases and check against it.	F, S, E, G

No	User Requirement	Origin
32	Provide ability to select the regulations against which a submission is to be checked.	F, S, G
33	Provide archival of submitted models	F, G
34	Provide integration with a microservice to check building CO2 compliance	F, E, G
35	Provide notification when building permitting is completed	S, E, G
36	Provide ability to export submitted model	S
37	Provide visualisation of BIM models	S, E, G
38	Be able to produce appropriate licenses and certificates	S, E
39	Be able to automatically determine the regulations to be checked against based on building criteria.	S
40	Be able to extract building and spatial information from IFC-file and visualize these results	E
41	Extract and check against building environmental data from national Digital Twin if IFC file is georeferenced.	E
42	Allow for the configuration of varying requirements for BIM models (modelling guidelines and Level of Information Needs) required for differing submission stages and building permit types.	G
43	Provide ability to generate documentation of BIM model requirements for differing submission stages and building permit types and to adapt to locally differing sets of predefined requirements.	G
44	Be able to extract building and spatial information from CityGML and IFC-files and check against requirements provided in standardised data format (XPlanXML in German context/ INSPIRE PLU in European context).	G
45	Be able to extract required information for the formal building permit application from IFC-files and convert it into XBau standard (XBauXML files).	G
46	Be able to support XBauXML files as an input format.	G
47	Provide integration with a microservice to provide Lifecycle Assessment for Green Building Certification.	G
48	Provide integration with a microservice to provide checking of timber construction systems.	G

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components

The following “Technical Requirements to ACCORD Cloud Architecture Components” list resulted from the Technical Requirements Elication Phases 1 and 2. The list also includes the technical requirements collected in WP1 (reported in ACCORD Framework User Requirements in D1.2). The elicitation phases, steps and elicitation criteria are destribed in detail in section 2 of this document.

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components - page 1 of 5

TR Elicitation 2.2 ACCORD Cloud Architecture Components Total no. of TRs assigned to components: 134			TR Elicitation 2.1 ACCORD Framework User Requirements Total no. of elicitated TRs: 41		TR Elicitation 1 ACCORD Framework Components Total no. of elicitated TRs: 93		Technical Requriements Elicitation Criteria													Change log			Alignement with Country- and Use Case-specific Requirements				
Ref. No.	Component No.	Name	Ref. No.	Name	Component No.	Name	Type	Category	Potential Source	Direct/ Indirect Source	Priority	Existence	Requirement Specificity	Responsible Task(s)	Short Description	Description	Rationale	Comments & Links to documentation	Responsible Partner/ Person	TR Elicitation 1	TR Elicitation 2.1	TR Elicitation 2.2	GE	FI	EE	UK	ES
	0 - Generic ACCORD Cloud Architecture				0 - Generic ACCORD Framework																						
	Not assigned.		4	ACCORD Framework User Requirements	0	ACCORD Framework										Provide sufficient customisation ability such that formally specified processes are generic enough to be scaled to the European level but flexible enough to allow the specific nature of each nation's permitting processes to be considered.			FhG / Katja Breitenfelder (Elicitation)		Not elicitated as Technical Requirement.	Not assigned to ACCORD Cloud Architecture.					
1	0	ACCORD Cloud Architecture	1	ACCORD Framework User Requirements	0	ACCORD Framework	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide a platform allowing to excecute digitised building permitting processes.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Changed description. Added TR elicitation criteria.						
2	0	ACCORD Cloud Architecture			0	ACCORD Framework	Functional	Functional Suitability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T1.3, T4.2, T4.3, T4.4	The system must allow to store, process, analyse and verify regulations for construction, rehabilitation and demolition works.	The system must apply the ACCORD semantic framework, answer its user requirements and store, process, analyse and verify regulations for construction, rehabilitation and demolition works.	The ACCORD semantic framework was defiend along with its user requirements specification.	Refer to the Proposal Part B-Page 30.	ITeC / Mercè Morilla	Changed responsible tasks and specified description.							
3	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Interoperability	End-user	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4, T4.5		The systems' services must follow best practices of deployment agnostic federation.	Potential sources: End-users and Software Developers. Microservice architecture interoperability relies one the implementation of agnostic and reliable integrations as much as possible using de-facto industry standards in data exchange. It will enable open ecosystems and avoid vendor lock-in.		OGC / Piotr Zaborowski	Changed type, added potential source, changed responsible tasks and specified rationale.							
4	0	ACCORD Cloud Architecture			0	ACCORD Framework	Functional	Interoperability	Software Architects	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must provide each component and microservice the ability to communicate with other components using well known web standards.	Well known web standards for communication allow easy connection between microservices. An ecosystem of microservices being loosely coupled and easily to be exchanged. In contrary, proprietary connections reduce the ease of coupling and changing.		FUI / Rick Makkinga	Added responsible tasks, specified rationale.							
5	0	ACCORD Cloud Architecture	12	ACCORD Framework User Requirements	0	ACCORD Framework	Functional	Interoperability	Software Architects	Indirect	MEDIUM	Existing	Overall Requirement	T4.2, T4.3, T4.4	The system should provide open standardized APIs and make use of open standards where applicable.	The system should provide open standardized application programming interfaces and make use of open standards where applicable (i.g. from OGC, buildingSMART etc.).			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria.		x				
6	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Interoperability	Software Architects	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4	The system must facilitate the availiablity of functionalities of each component and microservice through a documented API.	The system must facilitate the availiablity of functionalities of each component and microservice through a documented API, e.g. REST or GraphQL.	A documented API allows others to easily review what functionalities a microservice offers.		FUI / Rick Makkinga	Changed type, added responsible tasks.							
7	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Interoperability	Software Architects	Indirect	HIGH	Existing	Overall Requirement	T2.5, T4.2, T4.3, T4.4	The system must allow data exchange using open data standards being essential for cross systems exchange.	The system must allow data exchange using open data standards, e.g. IFC for BIM files using Open CDE API, BCF for checking results, CityGML/3D tiles for 3D city models or to-be-defined standards for defining rule-based checks.	Open standards are essential for cross systems exchange and prevent vendor lock-in.		FUI / Rick Makkinga	Changed type, added reaponsible task, specified description.							
8	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Interoperability	Software Architects	Direct	HIGH	Existing	Overall Requirement	T2.5, T4.2, T4.3, T4.4		The system must make use of existing BuildingSMART standards whereever possible.	To comply with current standardisation practices in the built environment.		CU / Thomas Beach	Changed type, added potential source, changed existence and responsible task.							
9	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Usability	Building Permit User	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should allow applicants to submit their building permit application and BIM models without difficulties.	Ensure an easy submission of BIM models.		HAM / Xinxin Duan	Changed category and potential source, added responsible tasks.							
10	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Usability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system's component must be available online and not require local installation on end user devices.	BIM models can be 'heavy' and in practice, especially the non technical users tend to have a device that cannot handle such heavy models. Smart online solutions that can solve that.		FUI / Rick Makkinga	Changed type, category, existence, responsible tasks, specified description and rationale.							
11	0	ACCORD Cloud Architecture	8	ACCORD Framework User Requirements	0	ACCORD Framework	Non-Functional	Usability	Local Authority	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should be dynamic with the ability to add and remove modules.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Changed description, added TR elicitation criteria.						
12	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Security	Security Compliance Engineer	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3		The system must use well-known standards for authentication of users.	OpenID and SAML are well known standars for authentication.		FUI / Rick Makkinga	Changed type and potential source, added responsible tasks, specified description.							
13	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Security	Security Compliance Engineer	Indirect	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must provide security measures to prevent "registration" or rogue services.	Users could try to compromise the system by registering a rogue microservice.		CU / Thomas Beach	Changed type, added potential source.							
14	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Security	Security Compliance Engineer	Indirect	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must provide appropriate security to allow access of building permitting results.	Different users should be able to see different levels of granularity for compliance checking results.		CU / Thomas Beach	Changed type, added potential source.							
15	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Security	Security Compliance Engineer	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system access services must use or implement control and clearance.	Access to resources has to be provided in accordance to predefined data access control and security clearance requirements.		OGC / Piotr Zaborowski	Changed type, category, potential source, responsible tasks, specified description, added rationale.							
16	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Security	Security Compliance Engineer	Indirect	MEDIUM	Existing	Overall Requirement	T4.2		The systems' components shall follow security best practices like OWASP, ISO 27001 according to local regulations and industry standards.	The system prototype shall assure it is possible to apply security concerns that would be required for the operational system, even if the prototype does not implement all of them.	It exists for standalone applications, while not neccesarily for APIs/ plugins being developed in the project.	OGC / Piotr Zaborowski	Changed potential source.							
17	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Compliance	Security Compliance Engineer	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must implement personal data protection regulations.	They are required for the system in order to be operational.		OGC / Piotr Zaborowski	Changed type, category, potential source, responsible tasks and specified description.							
18	0	ACCORD Cloud Architecture			0	ACCORD Framework	Functional	Compliance	System Customer	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must identify licensing approval processes based on applicable rules.	To answer WP objectives 4 and 5.	Refer to the Proposal Part B, page 30.	ITeC / Mercè Morilla	Changed category, potential source and responsible tasks.							
19	0	ACCORD Cloud Architecture			0	ACCORD Framework	Non-Functional	Compliance	System Customer	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must use reliable data being compliant with predefined rules according to user roles (e.g. proof of license, provenance, version, ownership).	The system must use reliable data being compliant with predefined rules according to user roles. The data must have license, provenance, version, ownership and vocabularies available as well as applied accuracy-/ quality measures.	Data exchange between public- and private data providers and receivers requires the definition of roles and responsibilities and the compliance with predefined rules. The demanded data reliability to be used for decision support in regulation processes requries to proof data ownership and data provenience.		OGC / Piotr Zaborowski	Changed type, category, potential source, priority, responsible tasks and specified description and rationale.							
	1 - Rule Formalization				1 - ACCORD Rule Formalization Approach 1a Rule Formalization Process																						
	Not assigend.		15	ACCORD Framework User Requirements	1a	Rule Formalization Process							Overall Requirement			Should provide the formalization of concepts from building codes/regulations in a semantic form.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added requirement specificity.	Refers to ACCORD Framework component 1a. Not assigned to ACCORD Cloud Architecture.					
	Not assigend.				1a	Rule Formalization Process	Functional	Interoperability	Software Architects	Direct	HIGH	Novel	Overall Requirement	T2.2, T2.3		The binding "Rule-Data" must have an abstraction layer in the middle. This layer should facilitate the use of data representation variants, and reduce the differences to insulate the higher-level rule components from such differences. Some of the differences will be accounted for by the rule creator (eg dedicates classes vs not). Others should not be the concern of the rule creator, or at any rate not for *every* rule to be authored: IFC version, RDF representation, RDF/binary split.	We don't know which IFC version(s) ACCORD should work with. Revit supports IFC 4.1 (but maybe not 4.3), whereas ArchiCAD is stuck in 2.3 or some such. Furthermore, the same building may be expressed in IFC in different ways: (1) With dedicated classes, e.g. IfcRamp (IsAccessible=true), (2) With CSG geometry that does't use dedicated classes but hopefully binds to appropriate bSdd to decribe that's a "ramp" (or we'll make IDS requirements that it must), (3) With BREP geometry (in fact IfcOpenShell converts to BREP because it uses OpenCascades for visualization).		ONTO / Vladimir Alexiev, Nataliya Keberle	Added responsible tasks, specified rationale.		Not assigned to ACCORD Cloud Architecture.					
					1b Data Dictionaries																						
																						Separated as ACCORD Cloud Architecture component number 2.					
					1c AEC3PO Ontology																						
	Not assigend.				1c	AEC3PO Ontology	Functional	Functional Suitability	Software Developers	Direct	MEDIUM	Novel	Overall Requirement	T2.2		The ontology shall respect meet high quality criteria. OWL2DL profile, consistent, good naming practices, metadata, good axiomatisation, good documentation, published according to best practices, and others. Potential sources are application developers and other users of the ontology.	This will foster its use.		IMT / Maxime Lefrançois	Added potential source.		Not assigned to ACCORD Cloud Architecture.					
					1d Domain Specific Language																						
	Not assigend.		13	ACCORD Framework User Requirements	1d	Domain Specific Language							Overall Requirement			The digitised format of building/codes regulations should be independent of any specific building modelling format.			FhG / Katja Breitenfelder (Elicitation)		Added requirement specificity.	Refers to ACCORD Framework component number 1d. Not assigned to ACCORD Cloud Architecture.	x				

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components - page 2 of 5

TR Elicitation 2.2 ACCORD Cloud Architecture Components Total no. of TRs assigned to components: 134			TR Elicitation 2.1 ACCORD Framework User Requirements Total no. of elicitated TRs: 41		TR Elicitation 1 ACCORD Framework Components Total no. of elicitated TRs: 93		Technical Requriements Elicitation Criteria													Change log			Alignement with Country- and Use Case-specific Requirements				
Ref. No.	Component No.	Name	Ref. No.	Name	Component No.	Name	Type	Category	Potential Source	Direct/ Indirect Source	Priority	Existence	Requirement Specificity	Responsible Task(s)	Short Description	Description	Rationale	Comments & Links to documentation	Responsible Partner/ Person	TR Elicitation 1	TR Elicitation 2.1	TR Elicitation 2.2	GE	FI	EE	UK	ES
					2 - Formalized Building Codes and Rules																						
					(1a+2) Rule Formalization Tool																						
20	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	System Customer	Direct	HIGH	Novel	Overall Requirement	T2.5		The system must provide a rule configurator allowing the user to evaluate rules and record provenance of the contribution (who/when said so).			ONTO / Vladimir Alexiev, Nataliya Keberle	Added potential source, specified description.							
21	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Building Permit User	Direct	HIGH	Novel	Overall Requirement	T2.5		The system must provide a rule tool allowing rule authors to bind rules to data (IFC, bsDD, various data representations).			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible task, specified description.							
22	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Building Permit User	Direct	MEDIUM	Novel	Overall Requirement	T2.3, 2.5		The system should provide a rule configurator allowing to select rule calculators depending of the data representation.			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible task, specified description.							
23	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Building Permit User	Direct	LOW	Novel	Overall Requirement	T2.5		The system should provide a rule tool assisting rule authors with auto- completions, in-place documentation and lookup into IFC- and bSdd-definitions.			ONTO / Vladimir Alexiev, Nataliya Keberle	Merged requirement. Changed responsible tasks, specified description.							
24	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T2.5	The system must provide a rule tool converting rules from YAML- to RDF-format and allowing rule authors to edit and explicate rules.	The system must provide a rule tool converting rules from building compliance rule language (YAML) to RDF and allowing rule authors to edit and explicate the rules.			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed potential source, responsible tasks and specified description.							
25	1	Rule Formalization			1a + 1b + (1a+2)	• Rule Formalization Approach • Data Dictionnaries • Rule Formalization Tool	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T2.4, T2.5	The system must provide NLP approaches for rule extraction from regulations (explicating clause structures using RASE or logical connectives).	The system must provide NLP with a function to break regulations into subclauses, down to atomic level, and explicates clause structure (using RASE and/or logical connectives).			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed potential source and responsible tasks.							
26	1	Rule Formalization			(1a+2)	(1a+2)	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T2.5	The system must provide a rule configurator allowing to select from a list of available rule calculators to execute the same rule.	The system must provide a rule configurator allowing to select from a list of available rule calculators to execute the same rule, but at least one rule calculator should be provided for each calculation.			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible tasks and specified description.							
27	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Developers	Indirect	HIGH	Novel	Overall Requirement	T2.5, T4.5	The system must be able to detect conflicting rules and provide a strategy to select the rule to be applied.	The system must be able to detect conflicting rules and provide a strategy to select the rule to be applied in such case (filter the conflicting rules based on the country or some specific conditions, or follow a pessimistic approach that prioritizes the worst case).	This will allow the users to solve issues on contradicting rules by themselves.		IMT / Maxime Lefrançois	Changed type, category and responsible tasks, specified description and rationale.							
28	1	Rule Formalization Tool	10	ACCORD Framework User Requirements	(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Developers	Direct	MEDIUM	Novel	Overall Requirement	T2.4, T2.5	The system should be able to leverage emerging AI techniques, such as semantic deep learning or NLP.	The system should be able to leverage emerging Artificial Intelligence techniques, such as semantic deep learning or Natural Language Processing (NLP).			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria.		x				
29	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Developers	Direct	LOW	Novel	Overall Requirement	T2.3, T2.5		The system may optionally have NLP finding entities, properties and measures (e.g. 2000 mm).			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed potential source.							
30	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Architects	Direct	HIGH	Novel	Overall Requirement	T2.5		The system must provide a rule formalization tool allowing regulation experts (e.g. municipality technician) to define checking rules for regulations.	This way, the expert user will be able to create a set of rules for checking one or more regulations.	Refer to the Proposal Part B, page 31.	FUNITEC / Gonçal Costa	Specified description.							
31	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Functional Suitability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T2.3, T2.5		The system must integrate the results of NLP processing of building regulations and codes together with other inputs to be used in the rule formalization tool.	Allowing the user (building code expert, e.g. municipality technician) to create rules without being an expert in the use of computer languages or programming.	Refer to the Proposal Part B, page 31.	FUNITEC / Gonçal Costa	Changed responsible tasks, specified rationale.							
32	1	Rule Formalization			1a + (1a+2)	• Rule Formalisation Process • Rule Formalisation Tool	Functional	Interoperability	End-user	Direct	HIGH	Novel	Overall Requirement	T2.3, T2.5	The system must enable mapping between abstract concepts in regulatory documents and concrete concepts in BIM data file format.	The system must enable mapping between abstract concepts in regulatory documents and concrete concepts in BIM data file format. E.g. data may be stored in a different place in different version of IFC model.	To allow automated lookup of data from a BIM model without regulation authors needing to have knowledge of IFC data structures.		CU / Thomas Beach	Added potential source and changed responsible tasks.							
33	1	Rule Formalization	5	ACCORD Framework User Requirements	(1a+2)	Rule Formalization Tool	Non-Functional	Usability	End-user	Indirect	HIGH	Novel	Overall Requirement	T2.3, T2.5	The system must provide an intuitive method to allow experts to digitise regulations and embed rules within them, without the need to write code.	The system must provide an intuitive method allowing regulation experts to digitise building codes/regulations and embed rules within them, without the need to write code.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria.						
34	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Localization	Software Developers	Indirect	HIGH	Novel	Overall Requirement	T2.3, T2.5, T4.5		The system must be able to classify rules by country.	This will help to apply the right set of rules according to the corresponding country where the building is located.		IMT / Maxime Lefrançois	Changed type.							
35	1	Rule Formalization			(1a+2)	Rule Formalization Tool	Functional	Localization	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T2.3, T2.5, T4.5		The system must provide country dependency between rules in the database and regulation documents.	To ensure NLP process preserves each country´s individual regulations.	Refer to the ACCORD Proposal - Part B, pages 11, 12, and 19.	JU / He Tan, Maria Hedblom	Changed category and responsible tasks.							
	2 - Data Dictionnaries																										
																						Separated component referring to ACCORD Framework component number 1b.					
	3 - Rule Repository and Provision																										
36	3	Rule Repository and Provision	6	ACCORD Framework User Requirements	No corresponding component.		Functional	Functional Suitability	System Customer	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide the ability to store a database of rules.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.	New ACCORD Cloud Architecture component.					
	4 - Information Requirements				Information Requirements																						
37	4	Information Requirements	27	ACCORD Framework User Requirements		Information Requirements	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must support the ability to perform model verification.	The system must support the ability to perform model verification - checking that a submission complies with relevant modelling schema.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
38	4	Information Requirements	7	ACCORD Framework User Requirements		Information Requirements	Functional	Functional Suitability	Software Developers	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must provide the ability to extract information requirements and to represent these using bS standards.	The system must provide the ability to extract information requirements for digital building permit- and compliance processes and to represent these as a standardised data schema using BuildingSMART standards (i.g. IDS).			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
39	4	Information Requirements	42	ACCORD Framework User Requirements		Information Requirements	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system should allow for the configuration of varying BIM requirements for differing submission stages and building permit types.	The system should allow for the configuration of varying requirements for BIM models (modelling guidelines and Level of Information Needs) required for differing submission stages and building permit types.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
40	4	Information Requirements	43	ACCORD Framework User Requirements		Information Requirements	Functional	Localization	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system should allow to generate BIM requirements for differing submission stages and permit types and to adapt to locally differing requirements.	The system should provide the ability to generate documentation of BIM (model) requirements for differing submission stages and building permit types and to adapt to locally differing sets of predefined requirements.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
	5 - Cloud-based Building Permit Services				3 - Cloud-based Building Permit Services																						
41	5	Cloud-based Building Permit Services	17	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must allow to link permit processes, applicable legislation and data standards and provide audit abilities to track decisions.	The system must provide the ability to link building permit processes, applicable legislation and building data standards, and provide audit abilities to track decisions.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
42	5	Cloud-based Building Permit Services	18	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide open access to limited data about building permit assessments.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
43	5	Cloud-based Building Permit Services	35	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide notification when building permitting is completed.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x		x		x
44	5	Cloud-based Building Permit Services	20	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system should enable direct communication between the submitter and building permit authority.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x	x		x
45	5	Cloud-based Building Permit Services	23	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should enable the collection of suitable evidence to complement assessments.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
46	10	Compliance Checking Microservices	46	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Others	Direct	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must be able to support XBauXML files as data input format.	The system must be able to support XBauXML files as data input format in the German context.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description. Potential source: Standardization boby.		x				
47	5	Cloud-based Building Permit Services	19	ACCORD Framework User Requirements	3	Cloud-based building Permit Services	Functional	Compliance	Local Authority	Indirect	HIGH	Existing	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide a standardised submission process.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
	6 - Model & Data Requirement Validation				3a Model & Data Requirement Validation																						
48	6	Model & Data Requirement Validation			3a	Model & Data Requirement Validation	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Existing	Overall Requirement	T4.2, T4.3, T4.4	The system should provide IDS checking services.	The system should provide IDS checking services that are able to process any IDS that meets the buildingSMART IDS specification.	IDS checking should stick to the standard and not have any custom requirements.		FUI / Rick Makkinga	Changed category and responsible tasks.							
49	6	Model & Data Requirement Validation	26	ACCORD Framework User Requirements	3a	Model & Data Requirement Validation	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must support information requirements validation of BIM models.	The system must support the ability to perform model validation – checking that it contains the required information to perform compliance checking.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
50	6	Model & Data Requirement Validation	16	ACCORD Framework User Requirements	3a	Model & Data Requirement Validation	Functional	Functional Suitability	Building Permit User	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide the ability to pre-check application data for compliance prior to formal submission.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
51	6	Model & Data Requirement Validation			3a	Model & Data Requirement Validation	Functional	Functional Suitability	Software Architects	Direct	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system should provide a gatekeeping functionality allowing to detect invalid and incomplete application data.	The system should provide a gatekeeping functionality to allow detection of invalid data format, while accepting that often partially incomplete submissions should be allowed.	To prevent submission of invalid or incomplete data sets.		CU / Thomas Beach	Changed category, potential source and responsible tasks.							

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components - page 3 of 5

TR Elicitation 2.2 ACCORD Cloud Architecture Components Total no. of TRs assigned to components: 134			TR Elicitation 2.1 ACCORD Framework User Requirements Total no. of elicitated TRs: 41		TR Elicitation 1 ACCORD Framework Components Total no. of elicitated TRs: 93		Technical Requriements Elicitation Criteria													Change log			Alignement with Country- and Use Case-specific Requirements				
Ref. No.	Component No.	Name	Ref. No.	Name	Component No.	Name	Type	Category	Potential Source	Direct/ Indirect Source	Priority	Existence	Requirement Specificity	Responsible Task(s)	Short Description	Description	Rationale	Comments & Links to documentation	Responsible Partner/ Person	TR Elicitation 1	TR Elicitation 2.1	TR Elicitation 2.2	GE	FI	EE	UK	ES
	7 - Process Execution				3b Process Execution																						
52	7	Process Execution			3b	Process Execution	Functional	Performance efficiency	Software Architects	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The processing services should run asynchronously for long operations.	Potentialy validation poses a demanding computing task. Asynchronous execution allows for reliable results, persistance and lower communication.		OGC / Piotr Zaborowski	Changed responsible tasks, specified description and rationale.							
53	7	Process Execution			3b	Process Execution	Non-Functional	Performance efficiency	Software Architects	Direct	LOW	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system shall allow to optimize the execution of automated compliance checking based on computation expense (time/cost) of a given logical operation.	Sometimes regulations could have multiple paths to a true or false. It should be possible to try to find the most computationally efficient route to find an answer to reduce checking time/cost.		CU / Thomas Beach	Changed type and responsible tasks, added potential source, specified description.							
54	7	Process Execution			3b	Process Execution	Functional	Reliability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The processing services must provide persistent activity and results logs with output links as relevant.	Potentialy validation poses a demanding computing task that shall leave the log and results offline.		OGC / Piotr Zaborowski	Changed responsible tasks, specified rationale.							
	8 - Data Storage				3c Data Storage																						
55	8	Data Storage	30	ACCORD Framework User Requirements	3c	Data Storage	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system should support the upload of BIM model files in an appropriate format.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x		x	x
56	8	Data Storage	33	ACCORD Framework User Requirements	3c	Data Storage	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system should provide archival of submitted BIM models.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x			
	9 - Orchestrating Micorservices				3d Orchestrating Microservices																						
57	9	Orchestrating Microservices			3d	Orchestrating Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide configurable components and services allowing to adjust core functionalities aacording to local regulations.	To avoid hardcoded rulesets and context data, it shall be possible to select and apply only relevant rules and data to the workflow, that is compliance checking of apllication data against regulations for particular municipalities. It can be the conbination of the national and local regulations.		OGC / Piotr Zaborowski	Changed category and responsible tasks, specified description and rationale.							
	10 - Compliance Checking Microservices Generic Requirements				4 - Compliance Checking Microservices																						
58	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Building Permit User	Direct	LOW	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system shall provide several reusable calculation components.			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible tasks.							
59	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4	The system must allow for manual assessment when automated checking is not available, and for a manual override of results by qualified users.	The system must allow for human input where no automated result is available and provide ability for human override of automated results (assuming correctly qualified user).	Many requirements are not automatically checkable - we should allow manual assessment to still deliver complete results.		CU / Thomas Beach	Added potential source and changed responsible tasks.							
60	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	End-user	Direct	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system should provide a description on the implementation of checks: "simple" using a declarative language (e.g. SPARQL) or "complex" by invoking specialized calculations.	The system should be able to describe how checks are implemented: "simple": implemented in a declarative language like BIM SPARQL; "complex": implemented by invoking specialized calculations (but the purpose and input/output of these routines is semantically described).			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible tasks.							
61	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Software Developers	Direct	MEDIUM	Existing	Overall Requirement	T4.2, T4.3, T4.4	The system should allow to use proprietary definitions for BIM compliance checking.	The system should provide the ability for BIM checks using a proprietary check definition (as there is no standard available).	There is no standard available that can be used. Once a standard for defining BIM checks is available, an additional requirement can be added that this standard is supported. However, it's expected that this standard will not cover every possible BIM check, so proprietary/custom BIM checks will still be present.		FUI / Rick Makkinga	Changed category, potential source, type of source and responsible tasks, specified description.							
	Solibri Office																										
	Future Insight Clearly.BIM																										
62	10	Compliance Checking Microservices	40	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system should be able to extract building- and spatial information from IFC-files and visualize the results.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.						
63	10	Compliance Checking Microservices	41	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system should be able to extract information from IFC-files and check it against building environmental data from national Digital Twin.	The system should be able to extract information from IFC-files and check it against building environmental data from national Digital Twin if IFC-files are georeferenced.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.				x		
	LCA Finland																										
	LCA Germany																										
64	10	Compliance Checking Microservices	47	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	System Customer	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must provide a microservice supporting LCA for Green Building Certification.	The system must provide a microservice supporting Lifecycle Assessments for Green Building Certification.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				
	Eurocode Compliance Checking																										
65	10	Compliance Checking Microservices	28	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must support the integration of a Finite Element Analysis compliance checking microservice.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.					x	
66	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Software Developers	Direct	HIGH	Existing	Country-specific Requirement	T4.2, T4.4		The system must link BIM data inputs with finite an element analysis tool to facilitate automated checks.	So that the BIM model and the tools can communicate seamlessly and generate credible analysis or checks.		BCU / Personell unspecified	Changed category, requirements specificity, responsible tasks and specified description.						x	
67	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Interoperability	Software Developers	Direct	MEDIUM	Existing	Country-specific Requirement	T4.3, T4.4		The system should use open data formats for delivering results of calculation software used for finite element analysis.	So that anyone can look at the data without restrictions and so that vendor lock-in is avoided.	Open data formats: 1. MED/Salome: https://docs.salome-platform.org/latest/dev/MEDCoupling/developer/med-file.html, 2. VTK: https://docs.vtk.org/en/latest/design_documents/VTKFileFormats.html, 3. XDMF: https://xdmf.org/index.php/XDMF_Model_and_Format	AEE / Ioannis P. Christovasilis	Changed requirements specifity and responsible tasks.						x	
	Urban Regulations																										
68	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must provide automated compliance checking of urban regulations using BIM models and GIS as data input.	The system's checking of compliance with urban regulations must be automated. The permitting process will be applicable to both public and private buildings during the design and construction phases, using BIM and GIS as data input.	So the process is automatically compliant with the regulations of the City Hall of Malgrat de Mar (AMM).	Refer to the Proposal Part B-Page 35.	ITeC / Mercè Morilla FUNITEC / Gonçal Costa	Merged requirement. changed responsible tasks.							x
69	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must allow to integrate GIS- and BIM data to carry out checking against urban regulations using one unified model.	So that the resulting model enables carrying out the automatic rule checking process for the Malgrat de Mar City Council (AMM).	Refer to the Proposal Part B-Page 35.	FUNITEC / Gonçal Costa	Changed responsible tasks, specified description.							x
70	10	Compliance Checking Microservices	39	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must be able to automatically determine the regulations to be checked against based on building criteria.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.						x
	Land Use Building Compliance Checking																										
71	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must allow local authorities to check submitted BIM models against land use requirements provided in XPlanGML format.	To integrate WMS service provided by HAM and offering land use requirements in XPlanGML format into the ACCORD system.		HAM / Xinxin Duan FhG / Katja Breitenfelder	Merged requirement. Changed category, added responsible tasks and specified description and rationale.			x				
72	10	Compliance Checking Microservices			4	Compliance Checking Microservices	Functional	Compliance	Software Developers	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system's spatial checking rules definitions should use estabilished vocabularies and query languages (like GEOSPARQL).	These well-known standards are resulting from the legacy of experiments and experience in formal expressing geo-related conditions and discrimination.		OGC / Piotr Zaborowski	Changed potential source, responsible tasks, specified description and rationale. Spatial checking rules aren't tackeld in T2.3.			x				
73	10	Compliance Checking Microservices	44	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Others	Direct	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must be able to extract building- and spatial information from IFC- (and CityGML)-files and check against requirements provided in XPlanXML format.	The system must be able to extract building- and spatial information from IFC-files (and CityGML, if applicable) and check against requirements provided in the standardised data format XPlanXML in the German context (and INSPIRE PLU in European context).			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description. Potential source: Standardization boby.		x				
74	10	Compliance Checking Microservices	45	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Others	Direct	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system should be able to extract information from IFC-files being required for formal building permit applications and convert it into XBauXML format.	The system should be able to extract information from IFC-files being required for formal building permit applications and convert it into the standardised data format XBauXML in the German context.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description. Potential source: Standardization boby.		x				
	Type Approval Building Compliance Service																										
75	10	Compliance Checking Microservices	48	ACCORD Framework User Requirements	4	Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must provide a microservice supporting compliance checks of timber construction systems and facilitating the type approval of buildings.	The system must provide a microservice supporting compliance checks of timber construction systems.	To facilitate the type approval of buildings.		FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description and rationale.		x				

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components - page 4 of 5

TR Elicitation 2.2 ACCORD Cloud Architecture Components Total no. of TRs assigned to components: 134			TR Elicitation 2.1 ACCORD Framework User Requirements Total no. of elicitated TRs: 41		TR Elicitation 1 ACCORD Framework Components Total no. of elicitated TRs: 93		Technical Requriements Elicitation Criteria													Change log			Alignement with Country- and Use Case-specific Requirements				
Ref. No.	Component No.	Name	Ref. No.	Name	Component No.	Name	Type	Category	Potential Source	Direct/ Indirect Source	Priority	Existence	Requirement Specificity	Responsible Task(s)	Short Description	Description	Rationale	Comments & Links to documentation	Responsible Partner/ Person	TR Elicitation 1	TR Elicitation 2.1	TR Elicitation 2.2	GE	FI	EE	UK	ES
	11 - Information Services				Information Services																						
																						New ACCORD Cloud Architecture component.					
	12 - API(s)				5 - API(s)																						
76	12	API(s)			5	API(s)	Functional	Interoperability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide a generic API framework to allow third parties to create new services that can interact with core components.	To allow for the growth of the system and to allow third parties to develop new components.		CU / Thomas Beach	Added potential source, changed responsible tasks.							
77	12	API(s)			5	API(s)	Functional	Interoperability	Software Developers	Direct	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must provide a Data Accessor Configurator in order to be able to access certain data from BIM models via external APIs.			BCU / Personell unspecified	Type, category and direct Source added, changed responsible tasks.							
78	12	API(s)			5	API(s)	Functional	Interoperability	Software Developers	Direct	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must facilitate the communication between system components through standardized APIs.	For example, compliance checking services all implement the same API that can be used by the Cloud-based Building Permit Service.		SOL / Pasi Paasiala	Changed responsible tasks, specified description and rationale.							
	Multiple ACCORD Cloud Architecture Components				Multiple ACCORD Framework Components																						
	Not assigend.				1b + 1c	• Data Dictionaries • AEC3PO Ontology	Functional	Interoperability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T2.2, T2.3		The developed ontology must use terms that are meaningful for people that understand building code, i.e. the description of the each term maps that to IFC and possibly other representations.			SOL / Pasi Paasiala	Added existence and responsible tasks.		Not assigned to ACCORD Cloud Architecture.					
	Not assigend.				1c + 1d	• Data Dictionaries • AEC3PO Ontology	Functional	Interoperability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T2.2, T2.3		The rule language must use terms of the ontology.			SOL / Pasi Paasiala	Changed existence, changed responsible tasks.		Not assigned to ACCORD Cloud Architecture.					
79	1 + 2 + 3 + 4	• Rule Formalization • Data Dictionaries • Rule Repository and Provision			(1a+2) + 1b + 3c	• Rule Formalization Tool • Data Dictionaries • Data Storage	Functional	Reliability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide versioning to all assets (models, rules, dictionaries).	Versioning is required to control changes in the environment and the model and be able to backtrack/document decisions.		OGC / Piotr Zaborowski	Changed potential source, responsible tasks and specified description.							
80	1 + 2 + 11	• Rule Formalization Tool • Data Dictionaries	14	ACCORD Framework User Requirements	(1a+2) + 1b + Information	• Rule Formalization Tool • Data Dictionaries	Functional	Interoperability	Software Developers	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should support the use of classification systems.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x			
81	1 + 2 + 12	• Rule Formalization Tool • Data Dictionaries • API(s)	9	ACCORD Framework User Requirements	(1a+2) + 1b + 5	• Rule Formalization Tool • Data Dictionaries • API(s)	Functional	Interoperability	Software Developers	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should support the integration of data dictionaries to enable mappings between regulatory terms and data schemas.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
82	1 + 5 + 10	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices			(1a+2) + 3 + 4	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Direct	HIGH	Novel	Overall Requirement	T2.5		The system must allow the visualisation of regulation documents in a human readable form.	To enable permitting officer and design teams to view the regulations against which they must comply.		CU / Thomas Beach	Added potential source, changed responsible tasks.							
83	1 + 5 + 10 + 12	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices			(1a+2) + 3 + 4	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must allow marking rules with subjective judgement as "isSubjective" (requires human judgement).			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed source to direct and added responsible tasks.							
84	1 + 5 + 10 + 12	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			(1a+2) + 3 + 4 + 5	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Functional Suitability	Software Architects	Direct	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must allow for the execution of high level logic behind building permit processes and regulation documents.	We need to be able to execute checks at granular level - even though each individual unit of execution may well be made of many decisions.		CU / Thomas Beach	Added potential source and responsible tasks.							
85	1 + 5 + 10 + 12	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			(1a+2) + 3 + 4 + 5	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Interoperability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide vocabularies that are both human and machine readible.	Human readibility of vocabularies allow for edits and manual reviews, machine readable ones are required for unambigious interpretations and release automation potential.		OGC / Piotr Zaborowski	Changed responsible tasks.							
86	1 + 5 + 10 + 12	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			(1a+2) + 3 + 4 + 5	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Usability	Software Developers	Indirect	MEDIUM	Novel	Overall Requirement	T2.5, T4.3, T4.4, T4.5		The system must add and display tags to the rules.	The tags will help classifying the rules (ex. dimension rule, fire safty rule, etc.). They also help finding redundant or contradictory rules, if any.		IMT / Maxime Lefrançois	Changed type.							
87	1 + 5 + 10 + 12	• Rule Formalization • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			(1a+2) + 3 + 4 + 5	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Usability	Local Authority	Direct	MEDIUM	Novel	Overall Requirement	T2.3, T2.4, T2.5, T4.3, T4.4		The system should allow for maintenance of links between results, and maintenance of regulatory clauses even across document version changes.	To allow for changing document versions - a common problem with construction regulations.		CU / Thomas Beach	Added potential source, changed responsible tasks.							
88	1 + 5 + 10	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices			(1a+2) + 3 + 4	• Rule Formalization Tool • Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Software Developers	Direct	LOW	Novel	Overall Requirement	T2.5, T4.3, T4.4		The system shall provide a plugin for an integrated development environment (IDE) allowing to edit and run the rules.	This will help the development of rules.		Personell unspecified.	References to multiple ACCORD components possible, to be further specified. Description: Text changes.							
89	1 + 3 + 8 + 12	• Rule Formalization Tool • Rule Repository and Provision • Data Storage • API(s)			(1a+2) + 3c + 5	• Rule Formalization Tool • Data Storage • API(s)	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Overall Requirement	T2.5, T4.3, T4.4	The rule formalization tool should link the regulation's document version to the rules being created. The system should allow updating the rule database accordingly, that is providing a data storage for archiving checking results including links to relevant rules and regulation document versions.	The rule formalization tool should link information on the regulation's document version to rules being created. The system should allow updating the rule database according to changes in regulation documents. It should provide building permit authorities with a data storage for archiving checking results that are linked to outdated rules and regulation document versions.	To keep the consistency between regulation documents and rules in the database due to governmental or other changes.	Refer to the ACCORD Proposal Part B, page 24.	FhG / Katja Breitenfelder JU / He Tan, Maria Hedblom	Merged requirement. Changed category, potential source and responsible tasks. Merged descriptions.							
90	1 + 10	• Rule Formalization • Compliance Checking Microservices			(1a+2) + 4	• Rule Formalization Tool • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Direct	HIGH	Novel	Overall Requirement	T2.3, T2.5, T4.3, T4.4		The system must be able to elaborate regulation clauses into checks.			ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible tasks.							
91	1 + 10 + 12	• Rule Formalization Tool • Compliance Checking Mircroservices • API(s)			(1a+2) + 4 + 5	• Rule Formalization Tool • Compliance Checking Mircroservices • API(s)	Functional	Functional Suitability	Software Developers	Indirect	HIGH	Novel	Overall Requirement	T2.5, T4.3, T4.4		The system must link or show the corresponding rule(s) that led to a given result.	This will help to justify the system results.		IMT / Maxime Lefrançois	Changed type, category, potential source and added responsible tasks.							
92	1 + 10 + 12	• Rule Formalization Tool • Compliance Checking Mircroservices • API(s)			(1a+2) + 4 + 5	• Rule Formalization Tool • Compliance Checking Mircroservices • API(s)	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Novel	Overall Requirement	T2.5, T4.3, T4.4		The system should allow users to select the set of rules to be used for the checking.	This will help to check the building compliance according to a set of specific rules instead of running all applicable rules.		IMT / Maxime Lefrançois	Changed type and category, added responsible tasks, added rationale.							
93	1 + 10 + 12	• Rule Formalization Tool • Compliance Checking Microservices • API(s)			(1a+2) + 4 + 5	• Rule Formalization Tool • Compliance Checking Microservices • API(s)	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4	The system must allow to visualize the orginal regulation text with checking rules and -results.	The system must allow for the lookup and visualisation of the text of the original regulation clause with any given result.	To aid designers in solving issues it is important to associate the result with the textual clause that generated the result.		CU / Thomas Beach	Added potential source.							
94	1 + 12	• Rule Formalization Tool • API(s)			(1a+2) + 5	• Rule Formalization Tool • API(s)	Functional	Functional Suitability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T2.5, T4.2, T4.3, T4.4	The rule formalization tool must be able to communicate with the rule database.	The system's rule formalization tool must be able to communicate with the database where the rules will be stored.	This way, the microservices / applications can access the database to obtain the set of rules necessary to perform the ACC according to the selected regulation.	Refer to the Proposal Part B, page 31.	FUNITEC / Gonçal Costa	Changed responsible tasks.							
95	4 + 5 + 10 +12	• Information Requirements • Cloud-based Building Permit Services • Compliance Checking Microservice • API(s)			Information Requirements + 3 + 4 + 5	• Information Requirements • Cloud-based Building Permit Services • Compliance Checking Microservice • API(s)	Non-Functional	Performance efficiency	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system's services must use resources reasonably by narrowing down operations, that is only using full spartial or building models to execute checks when necessary.	The system's services must use resources reasonably. In particular, it must narrow down operations in order not to execute full (spatial) scans, and transmit the whole building models when not neccesary.	Economy of technical exploitation shall be taken into account as well as user station capacity and user experience on the web. Level of Development (LOD) specifications and tailing mechanism can help here.		OGC / Piotr Zaborowski	Changed type, category, responsible tasks, specified description and rationale.							
96	4 + 6	• Information Requirements • Model- & Data Requirement Validation			Information Requirements	• Information Requirements • Model- & Data Requirement Validation	Functional	Functional Suitability	Software Developers	Direct	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4		The system must provide and facilitate BIM-model validation.	Validate the BIM models against requirements (e.g. IDS).		BCU / Personell unspecified	Changed categroy, added responsible task and specified							
97	5 + 8 + 10 + 12	• Cloud-based Building Permit Services • Data Storage • Compliance Checking Microservice			3 + 3c + 4 + 5	• Cloud-based Building Permit Services • Data Storage • Compliance Checking Microservice	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide a central repository for checking results.	Compliance checking results should be lodged centrally for achival purposes.		CU / Thomas Beach	Added potencial source, changed source and responsible tasks.							
98	5 + 8 + 10 + 12	• Cloud-based Building Permit Services • Data Storage • Compliance Checking Microservices	3	ACCORD Framework User Requirements	3 + 3c + 4 + 5	• Cloud-based Building Permit Services • Data Storage • Compliance Checking Microservice	Functional	Functional Suitability	End-user	Indirect	HIGH	Existing	Overall Requirement	T4.3, T4.4		The system must support both Geospatial and BIM(IFC) data input.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
99	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	System Customer	Indirect	MEDIUM	Existing	Overall Requirement	T4.3, T4.4		The system should make use of (IFC to) RDF tools.	Facilitate data generation in semantic web format.		BCU / Personell unspecified	Added type, category, indirect source, existamce and responsible tasks.							
100	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	2	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide automated compliance checking.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
101	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	37	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	HIGH	Existing	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide visualisation of BIM models.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x		x		x
102	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	11	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system should provide a set of microservices, with tools and solutions for digital permit- and automated compliance checking of buildings.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
103	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Novel	Overall Requirement	T4.3, T4.4		The system should be able to bind checks to data.	Not specified.		ONTO / Vladimir Alexiev, Nataliya Keberle	Changed priority, type of source and responsible tasks.							
104	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Direct	MEDIUM	Novel	Overall Requirement	T4.2, T4.3		The system should provide cognitive services specific to building requirements, so that	So that we can demonstrate the benefits of assessing structural integrity using BIM based	Refer to the ACCORD Proposal, Part B, page 35.	BCU / Franco Cheung	Changed responsible tasks.							
105	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	24	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Existing	Country-specific Requirement	T4.2, T4.3, T4.4		The system should retain the ability for manual human input.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x				x
106	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	End-user	Indirect	MEDIUM	Existing	Country-specific Requirement	T4.3, T4.4	The system should allow to visualise and select information from GIS models.	The visualisation and building context information selection from GIS models shall support various levels of details or tilling.	City models can be of significant size. Loading all data is not always possible. Smart, context depending selection is required both, for checking models against regulations and for visualisation.	It exists for standalone applications, while not neccesarily for APIs being used.	OGC / Piotr Zaborowski	Changed category, requirement specifity, responsible tasks and specified description and rationale.			x				
107	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	21	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Non-Functional	Security	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system should provide security models allowing to differentiate between users and their access to checking services and/or results.	The system should provide suitable security models allowing to differentiate between users and to enable assess for appropriate users to regulations and checking results.			FhG / Katja Breitenfelder (Elicitation)				x	x	x		x
108	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must allow the user to view and manage building permit process status.	Potencial sources: Builing Permit User (design teams to view the processes they should follow) and Local Authority.		CU / Thomas Beach	Added potential source, specified description and rationale.							

Appendix 2. Technical Requirements to ACCORD Cloud Architecture Components - page 5 of 5

TR Elicitation 2.2 ACCORD Cloud Architecture Components Total no. of TRs assigned to components: 134			TR Elicitation 2.1 ACCORD Framework User Requirements Total no. of elicitated TRs: 41		TR Elicitation 1 ACCORD Framework Components Total no. of elicitated TRs: 93		Technical Requriements Elicitation Criteria													Change log			Alignement with Country- and Use Case-specific Requirements				
Ref. No.	Component No.	Name	Ref. No.	Name	Component No.	Name	Type	Category	Potential Source	Direct/ Indirect Source	Priority	Existence	Requirement Specificity	Responsible Task(s)	Short Description	Description	Rationale	Comments & Links to documentation	Responsible Partner/ Person	TR Elicitation 1	TR Elicitation 2.1	TR Elicitation 2.2	GE	FI	EE	UK	ES
	Multiple ACCORD Cloud Architecture Components				Multiple ACCORD Framework Components																						
109	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must allow the user to explore and visualise the results of building permit checks.	To enable all users to visualise the results of building permitting in a way that is suitable for their use. I.e. design team to solve faults, permitting officers to assess etc. Potencial sources: Building Permit User and Local Authority.		CU / Thomas Beach	Added potential source, specified description and rationale.							
110	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.3, T4.4	The system must allow experts to intervene and to give feedback to checking result.	The system must allow to the building permit compliance expert to intervene and to give feedback to every result.	To improve the system's functionality and accuracy.		IMT / Maxime Lefrançois	Changed category, potential source, responsible tasks, specified description and							
111	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices	38	ACCORD Framework User Requirements	3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.3, T4.4		The system should be able to produce appropriate licenses and certificates.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.				x		x
112	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Building Permit User	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.3, T4.4	The system should allow to view land use plans (originated from the XPlanGML format) before starting the planning.	The system should provide the user with the ability to view land use plans (originated from the XPlanGML format) before starting the planning.	Applicants should be able to check their application data against regulations stated in the land use plan.		HAM / Xinxin Duan	Merged requirement (same author). Changed potential source, added responsible tasks and specified description and rationale.			x				
113	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must be able to plan the order and execution of rule checkings.	Example: "x AND y UNLESS z, in which case a OR b".		ONTO / Vladimir Alexiev, Nataliya Keberle	Changed responsible tasks, specified description and rationale.							
114	5 + 10	• Cloud-based Building Permit Services • Compliance Checking Microservices			3 + 4	• Cloud-based Building Permit Services • Compliance Checking Microservices	Functional	Functional Suitability	Software Developers	Direct	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system should provide semantic BIM enrichment (before compliance checking).	The system should provide semantic BIM enrichment, so that BIM models comply with information requirements, before the start of compliance checking.			BCU / Personell unspecified	Changed category and responsible tasks, specified description and rationale.							
115	5 + 10 + 11	• Cloud-based Building Permit Services • Compliance Checking Services • Information Services			3 + 4 + Information Services	• Cloud-based Building Permit Services • Compliance Checking Services • Information Services	Non-Functional	Scalability	Software Architects	Indirect	MEDIUM	Novel	Overall Requirement	T4.3, T4.4		The system should provide services that are scalable, preferably containerised and clusterable.	The solution's potential to include operational environments and scalability must be evaluated.		OGC / Piotr Zaborowski	Changed category, responsible tasks, specified rationale.							
116	5 + 10 + 11	• Cloud-based Building Permit Services • Compliance Checking Services • Information Services			3 + 4 + Information Services	• Cloud-based Building Permit Services • Compliance Checking Services • Information Services	Functional	Localization	End-user	Indirect	MEDIUM	Existing	Overall Requirement	T4.3, T4.4	The system's services should support multiple languages.	The system's services should be able to add support for multiple languages (it's not yet required to actually have language sets for every language).	For integration in local legal systems, it's often required that the local native language can be used. Therefor, services should allow for translations.		FUI / Rick Makkinga	Changed category and responsible tasks.							
117	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	22	ACCORD Framework User Requirements	3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Services • API(s)	Functional	Functional Suitability	End-user	Indirect	HIGH	Existing	Overall Requirement	T4.2, T4.3, T4.4	The system must provide the generation of human- and machine-readable (BCF-based) reporting.	The system must provide the generation of human- and machine-readable (BCF-based) reporting based on submissions.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
118	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	25	ACCORD Framework User Requirements	3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Services • API(s)	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide the ability to share compliance checking results with other relevant users.			FhG / Katja Breitenfelder (Elicitation)		Applies to all demo countries. Added TR elicitation criteria, specified description.						
119	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	32	ACCORD Framework User Requirements	3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Services • API(s)	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide the ability to select regulations against which a submission is to be checked.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x			x
120	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Functional Suitability	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must allow to document and use checking results for communicating further steps with the applicant.	The system must provide the ability to document and to use checking results for communicating further steps with the building permit applicant.	Results must be communicated in an understandable way.		HAM / Xinxin Duan	Changed responsible tasks.							
121	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Interoperability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The Cloud-based Building Permit Services must provide user-based authentication: using identifiers when passing tasks to checking services and when returning back results.	The compliance checking microservices do not do user-based authentication. Cloud-based Building Permit Service uses an identifier when passing a checking task to the checking service and when checking service is ready with the results, the results are returned back to the permitting service with the identifier.			SOL / Pasi Paasiala	Added existence, changed responsible task and specified description.							
122	5 + 10 + 12	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)			3 + 4 + 5	• Cloud-based Building Permit Services • Compliance Checking Microservices • API(s)	Functional	Interoperability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide interoperability between the rule database and microservices.	So that the microservices that require the application of the rules from this database can connect properly.	Refer to the ACCORD Proposal Part B, page 31 and 45.	FUNITEC / Gonçal Costa	Changed responsible tasks.							
123	6 + 12	• Model and Data Requirement Validation • API(s)	29	ACCORD Framework User Requirements	3a + 5	• Model and Data Requirement Validation • API(s)	Functional	Functional Suitability	Building Permit User	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4	The system must provide reporting results of model verification and validation.	The system must support the provision of reporting results of model verification and validation.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x		x	x
124	7 + 9	• Process Execution • Orchestrating Microservices			3b + 3d	• Process Execution • Orchestrating Microservices	Non-Functional	Security	Software Architects	Indirect	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system processing service definitions shall separate access, control, execution and input/ output data definitions.	For high Technology Readiness Levels (TLR), these separations are matter of security and operation management.		ONTO / Vladimir Alexiev, Nataliya Keberle	Changed type, catregory and responsible taks, specified rationale.							
125	8 + 10 + 12	• Data Storage • Compliance Checking Microservice • API(s)			3c + 4 + 5	• Data Storage • Compliance Checking Microservice • API(s)	Functional	Performance efficiency	Internal Technical Analyst	Direct	MEDIUM	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system should facilitate the simultaneous execution of compliance checking services upon (parts of) IFC models.	The system should facilitate that multiple microcervices communicate the simultaneous process of compliance checking using parts of the IFC model.	This will improve required data ascquisition.		BCU / Personell unspecified	Changed responsible tasks, specified description.							
126	8 + 12	• Data Storage • API(s)	36	ACCORD Framework User Requirements	3c + 5	• Data Storage • API(s)	Functional	Functional Suitability	Local Authority	Indirect	MEDIUM	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system should provide the ability to export submitted (BIM) models.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.						x
127	10 + 11	• Compliance Checking Microservices • Information Services			4 + Information Services	• Compliance Checking Microservices • Information Services	Functional	Interoperability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4	The system must make use of linked data modelling approaches, that is use URI/ URLs for referencing definitions and instances.	The system's data must be modeled as linked data in the sense that URI/ URLs shall be used when referencing definitions and instances.	For the sake of complete information, unambiguity of interpretations and data exchange efficiency, Linked Data approach shall be used whenever possible.		OGC / Piotr Zaborowski	Changed responsible tasks.							
128	10 + 11	• Compliance Checking Microservices • Information Services			4 + Information Services	• Compliance Checking Microservices • Information Services	Functional	Interoperability	Local Authority	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must read specific attribute values from OGC services (originated from INSPIRE PLU data).	The system must provide the local authority the ability to compare the attribut value from INSPIRE PLU data which is served by OGC services with values from the submitted BIM model.	INSPIRE Planned Land Use (PLU) Data Model can be found here: https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html. The data itself would also be served through OGC services, but in a different data structure as per the INSPIRE Directive.	HAM / Xinxin Duan	Changed requirement specifity and responsible tasks.			x				
129	10 + 11 + 12	• Compliance Checking Microservices • Information Services • API(s)	31	ACCORD Framework User Requirements	4 + Information Services + 5	• Compliance Checking Microservices • Information Services • API(s)	Functional	Functional Suitability	Software Developers	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system's compliance checking microservices must allow to extract data from national databases and/or check against it.			FhG / Katja Breitenfelder (Elicitation)				x	x	x		x
130	10 + 11 + 12	• Compliance Checking Microservices, • Information Services, • API(s).			4 + Information Services + 5	• Compliance Checking Microservices • Information Services • API(s)	Functional	Localization	Local Authority	Indirect	HIGH	Novel	Overall Requirement	T4.2, T4.3, T4.4		The system must provide support to various Coordinate Reference Systems (CRSes) supporting spatial data integration from external sources.	Various countries have their own CRS for GIS data, lack of support blocks potential reusability.		OGC / Piotr Zaborowski	Specified description, changed responsible tasks.							
131	10 + 12	• Compliance Checking Microservices • API(s)	34	ACCORD Framework User Requirements	4 + 5	• Compliance Checking Microservices • API(s).	Functional	Functional Suitability	System Customer	Indirect	HIGH	Novel	Country-specific Requirement	T4.2, T4.3, T4.4		The system must provide integration with a microservice to check building CO2 compliance.			FhG / Katja Breitenfelder (Elicitation)		Added TR elicitation criteria, specified description.		x	x	x		
132	11 + 12	• Information Services • API(s)			Information Services + 5	• Information Services • API(s)	Functional	Functional Suitability	Software Developers	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must allow to match services to logical statements within the digitised regulation.	A given compliance check may well require information from several different services - this should be supported.		CU / Thomas Beach	Added potential source, specified rationale.							
133	11 + 12	• Information Services • API(s)			Information Services + 5	• Information Services • API(s)	Functional	Interoperability	Software Architects	Indirect	HIGH	Novel	Overall Requirement	T4.3, T4.4	The system must define integration and interoperability pathways for geospatial- and BIM data regarding data modelling, semantics and interface design using open source OGC and bS standards.	The system must be able to define an integration and interoperability pathway between geospatial models and BIM models approaches with respect to data modelling, semantics, and interface design using open source standards defined by the OGC and bSI respectively.	In order to the ideas will be exchanged with other projects funded under the same call in a working group that will be set up by the EU BIM Task Group. This task will develop best practices that describe how geospatial and BIM models can be used jointly to reduce integration costs and improve build quality along with the entire design, build, finance, maintain the lifecycle of buildings.	Refer to the ACCORD Proposal Part B, pages 36 and 37.	ITeC / Mercè Morilla	Changed responsible tasks.							
134	11 + 12	• Information Services • API(s)			Information Services + 5	• Information Services • API(s)	Functional	Functional Suitability	Software Architects	Direct	HIGH	Novel	Overall Requirement	T4.3, T4.4		The system must allow for the "registration" of services and component capabilities (e.g. what can each component do).	The system should be dynamic in that a new service can be added without need to excessive reconfiguraiton of other components.		CU / Thomas Beach	Applied potential source.							