



SIRCULAR

DELIVERABLE D2.1:

SIRCULAR platform WP2

Date of document

26/06/2025

Deliverable Version: D2.1, V2.0

Dissemination Level: Public

Deliverable Lead: GENEGIS GI Srl



OVERVIEW	
Project Acronym	SIRCULAR
Project Title	Sustainable and Integrated people centRIC solUtions for buiLding decARbonisation and circularity.
Project Coordinator	RINA-C
Project Duration	42 months
Deliverable Title	SIRCULAR platform
Deliverable No.	D2.1
Dissemination Level	Public
Work Package	WP2: SIRCULAR platform development
Deliverable Lead	GENEGIS GI Srl
Lead Authors	Massimiliano, Corno (GENEGIS); Josè, Otero (GENEGIS); Gianpiero, Amanzio (GENEGIS)
Main Contributor Authors	Roberta, Rini (RINA-C); Giuseppe, Turrisi (RINA-C);
Other Contributors	CERTH - HPHI - NTUA
Due date	30/06/2025
Submission date	26/06/2025
Reviewers	CERTH - ITeC



Co-funded by
the European Union

SIRCULAR has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101147412. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

Abbreviations and Acronyms	6
Background: About the SIRCULAR project	7
Executive Summary	8
1. Introduction	9
1.1 Objectives of the Deliverable	9
1.2 Structure of the Document	11
1.3 Relation to Project Documents	11
1.4 Overall Approach.....	12
2. General Architecture	13
2.1 Platform Architecture: Overview	13
2.2 Core Components and Interaction Flows.....	14
3. Cloud Infrastructure – Azure Kubernetes Service (AKS)	18
3.1 AKS configuration	19
3.2 Namespace management (Tool, Frontend, Database).....	19
3.3 Horizontal Pod Autoscaler.....	20
4. Webapp Components	21
4.1 Ingress Controller	21
4.2 Webapp Pod and Webapp Service.....	22
4.2.1 Interaction with Internal Tools	23
4.3 Persistent Storage	24
4.4 Monitoring with Azure Monitor	25
5. User Management and Security	26
5.1 Access through Azure B2C (SSO and AD credentials)	26
5.2 Secured APIs.....	27
6. Tools	29
6.1 Developed tool	30



Co-funded by
the European Union

6.1.1	VERIFY - Virtual Platform on Life Cycle Analysis	30
6.2	Tools to be developed	32
6.2.1	Geodatabase Tool	32
6.2.2	Hygrothermal Simulation Tool	33
6.2.3	Comfort Evaluation Tool	34
6.2.4	CIAT Tool	35
6.2.5	Marketplace	36
7.	DevOps	37
7.1	Docker Registry and DevOps workflows (Docker Push/Pull)	37
7.2	Update pipeline and Automatic deployment	37
7.2.1	Phase 1: Tool-Level Continuous Integration and Pre-Deployment Validation	38
7.2.2	Phase 2: Partner-Led Deployment to Shared Infrastructure	38
7.3	Versioning and environment management	39
8.	Database and Data Persistence	40
8.1	PostgreSQL with PostGIS extension	40
8.2	MySQL: usage and use cases	41
8.2.1	CIAT Tool	41
8.2.2	Comfort Evaluation Tool	41
9.	Conclusions	42
10.	References	44



Co-funded by
the European Union

Table of Tables

<i>Table 1 - Analysis per associated partner</i>	10
<i>Table 2 - System Architecture, main components</i>	15
<i>Table 3 - WEBAPP POD AND WEBAPP SERVICE COMPONENTS</i>	22

Table of Figures

<i>Figure 1 - Architectural diagram</i>	13
<i>Figure 2 - Tools Internal (to be developed) and External (developed)</i>	29



Co-funded by
the European Union

Abbreviations and Acronyms

ACRONYM	DESCRIPTION
AKS	Azure Kubernetes Service
BIM	Building Information Modelling
CERTH	Centre for Research and Technology, Hellas
CIAT	Circularity Index Assessment Tool
D	Deliverable
EU	European Union
HPA	Horizontal Pod Autoscaler
HPHI	Hellenic Passive House Institute
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaC	Infrastructure as Code
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
JWT	Json Web Token
KPI	Key Performance Indicator
KPI	Key Performance Indicator
LCA	Life Cycle Assessment
LCC	Life Cycle Cost
MRT	Mean Radiant Temperature
NTUA	National and Technical University of Athens
SSO	Single Sign-On
TLS	Transport Layer Security
WP	Work Package



Co-funded by
the European Union

SIRCULAR has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101147412. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Background: About the SIRCULAR project

SIRCULAR is coordinated by RINA-C and combines the expertise of 22 partners from six European countries, including universities, SMEs, NGOs, and industries. During the next three-and-a-half-years, SIRCULAR will transform the building sector into a circular and sustainable industry, aligned with the Built4People partnership principles.

We will test and demonstrate innovative technologies and services across four regional clusters: initially in Estonia and Spain, followed by Germany and Greece. These clusters will engage construction companies, housing companies, universities, and local administrative entities, focusing on buildings owned or occupied by vulnerable population groups, in line with the SIRCULAR just and affordable transition approach.



Co-funded by
the European Union

SIRCULAR has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101147412. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

This deliverable provides a comprehensive overview of the architectural design and technical foundation of the SIRCULAR digital platform, a key output of the Horizon Europe project aimed at fostering sustainable and circular practices in the construction sector. The platform is intended to serve as a centralized hub supporting both technical and non-technical users through integrated tools, ensuring inclusiveness, interoperability, and scalability. The goal of this deliverable is to define the platform's core infrastructure and components while enabling seamless communication between its internal architecture and the tools developed or integrated under the project.

The SIRCULAR platform adopts a modular, microservice-based architecture hosted on Microsoft Azure and orchestrated via Azure Kubernetes Service (AKS). The chosen infrastructure ensures dynamic scalability, robust performance, and secure data handling. User management is implemented through Azure B2C, enabling Single Sign-On (SSO) and granular role-based access. Each platform component is containerized and managed within dedicated namespaces to guarantee logical separation, traceability, and efficient resource allocation. The interaction between tools and services is supported through secure APIs and a common authentication layer, while persistent storage and real-time monitoring are ensured via Azure Blob Storage and Azure Monitor. Tools are integrated following standardized UX/UI principles and CI/CD pipelines, with container orchestration supporting seamless updates and deployment workflows.

The deliverable details the core components of the platform—including the web application services, database infrastructure, internal and external tools, and security mechanisms—and documents how these elements interact to form a cohesive and resilient system. Specific contributions from project partners (GENEGIS, RINA, HPHI, NTUA, and CERTH) are highlighted, particularly in relation to tool integration and architectural definition. Internal tools such as the GeoDatabase Tool, CIAT Tool, Comfort Evaluation Tool, and Hygrothermal Simulation Tool are presented alongside already developed solutions like the VERIFY platform. Their integration illustrates the platform's capacity to support comprehensive assessments of circularity, energy performance, material reuse, and comfort evaluation.

In conclusion, the deliverable defines the structural and technical underpinnings of the SIRCULAR platform and provides a stable reference architecture for the deployment of digital tools in line with project goals. The methodologies and infrastructure described will enable further development phases, including tool-specific refinements and the expansion of platform functionalities foreseen in Deliverables D1.3, T2.3, and T4.4. Moving forward, the established framework will support continued integration, scalability, and alignment with European digital standards for decarbonisation and circularity in the built environment.



Co-funded by
the European Union

1. Introduction

The SIRCULAR project will introduce an advanced digital platform designed to provide non-expert users in the construction sector with the necessary tools and resources to promote sustainable and circular building practices. Serving as a centralized hub, SIRCULAR offers a structured environment for those looking to integrate circular economy principles into their projects. Therefore, the primary objective of this deliverable is to provide a comprehensive overview of the SIRCULAR platform and to define its core components, ensuring system scalability and seamless communication between the platform and external tools.

1.1 Objectives of the Deliverable

The objective of this deliverable is to provide a comprehensive assessment of the rationale behind the selection of the infrastructure type and its main components, with the aim of ensuring system scalability and delivering an optimal user experience for the various users accessing the platform.

The platform includes a suite of cutting-edge tools that simplify complex processes, as detailed in the following sections. With an intuitive and well-organized interface, SIRCULAR supports users in navigating the challenges of sustainable construction, making circularity a tangible and achievable objective. The initial design and testing of the platform interface are carried out by the project partners, leveraging their expertise and prior experience in the development of similar digital solutions. This first version is shaped by the technical and functional knowledge acquired through the design and implementation of user-centric platforms in related domains. Once the preliminary interface is established, it will be shared and tested with the relevant stakeholders involved in the SIRCULAR project. Their feedback will be essential to refine the user experience and ensure that the platform meets the needs and expectations of its diverse user base.

The platform's architecture will ensure seamless access to both internally and externally tools, facilitating efficient data exchange and interoperability across the platform's diverse functionalities. The tools, further elaborated in the other sections, are designed to accommodate both technical and non-technical users. Industry professionals will be able to assess the circularity of buildings, whether existing or in the design phase, while non-technical users will have the capability to perform preliminary evaluations, such as analyzing a building plan's thermal comfort or hygrothermal performance.



Co-funded by
the European Union

Table 1 - Analysis per associated partner

Partners	Specific Contribution
GENEGIS	Task leader: responsible for creating a comprehensive plan that outlines the structure, components, interactions, and design of the overall platform as well as each individual component. In addition, the task leader described the creation process of the Geo-database and its interaction within the platform environment, highlighting its role in supporting data management and integration across tools
RINA	Support has been provided to the task leader in defining the cloud provider and its main components, which contribute to the platform's performance, scalability, and security features. Additionally, the partner described how the CIAT Tool interacts with the platform, detailing its integration and the role it plays within the overall system architecture
HPHI	The partner's contribution focused on detailing how the Hydrothermal Simulation Tool operates within the platform environment
NTUA	The partner's contribution focused on detailing how the Comfort Evaluation Tool operates within the platform environment
CERTH	The partner's contribution focused on detailing how the Verify Tool interacts with the platform



Co-funded by
the European Union

1.2 Structure of the Document

This deliverable is designed to provide a clear and comprehensive overview of the architectural and technological foundations of the SIRCULAR platform. It begins by contextualising the document within the overall project framework and clarifying its objectives and methodological approach. The document then introduces the general architecture of the platform, outlining its structure, key components, and the way these interact to enable a scalable, modular, and interoperable system.

Particular attention is given to the configuration of the cloud infrastructure, which is based on Microsoft Azure and leverages Kubernetes for orchestration and deployment. The design and functionality of the platform's web application components are also addressed, including ingress management, service interactions, and mechanisms for data persistence and system monitoring.

User management and security considerations are explored, with an emphasis on identity federation through Azure B2C and the use of secure APIs to ensure controlled and traceable access to platform resources. The document also provides an overview of the tools integrated into the platform, covering both new developments and existing solutions, and describing their role within the broader system.

Furthermore, the deliverable outlines the DevOps strategy adopted to support continuous integration, automated deployment, and version control, as well as the database technologies employed to guarantee consistent and reliable data storage. The document concludes with a summary of the main elements discussed, offering a foundation for the continued development and deployment of the SIRCULAR platform.

1.3 Relation to Project Documents

This deliverable is connected to several other project deliverables, since it outlines the platform's architecture. In particular, there is a close connection with the following deliverables:

- D1.2 which develops two algorithms for evaluating the circularity and recyclability indexes within the SIRCULAR Platform.
- D1.3 establish a standardized methodology framework for the SIRCULAR digital tools, ensuring compatibility, interoperability, and effectiveness.

This document is also directly linked to the development activities carried out in Task T2.2 (SIRCULAR Platform) and Task T2.3 (Tools Development). The tools developed within T2.3 will be released on the SIRCULAR platform, following a continuous integration and deployment approach. After each release, a bug-fixing phase will be conducted to address any technical issues or inconsistencies. These releases will serve as key checkpoints for carrying out



Co-funded by
the European Union

targeted reviews on tool functionality and user experience. Such reviews will actively involve the partners and stakeholders engaged in WP4 (Monitoring and Demonstration) and WP5 (Stakeholder Engagement and Validation), who will provide feedback to refine both technical performance and usability aspects. This iterative process will support the progressive enhancement of the tools and ensure their alignment with end-user expectations and project objectives.

Moreover, as part of the platform monitoring strategy, secure and authenticated APIs will enable data exchange between WP2 and Task 4.4, allowing tools to access relevant monitoring data and performance metrics.

1.4 Overall Approach

The development of the SIRCULAR platform involves several critical steps. Initially, the cloud infrastructure for SIRCULAR was identified, focusing on scalability and compatibility with project requirements. Key components such as Azure Kubernetes Service (AKS) and Single Sign-On (SSO) as B2C were chosen to form the core infrastructure.

Secondly, functionality analysis encompassed both foundational aspects of the SIRCULAR platform and the internal and external Tools integrated via APIs. These functionalities, detailed in Deliverable D1.3 and strictly linked with T2.3, were examined to determine the data types necessary for effective tool operation. Moreover, communication methods between the SIRCULAR platform and its Tools, both internal and external, were defined to ensure seamless integration and operation.

Additionally, considerations for the Industry Foundation Classes (IFC) format and database types suitable for storing IFC file data were outlined. The use of the IFC format and the creation of a dedicated database from its attributes enable the extraction of structured information and data, which can be leveraged by the Tools to enhance usability and provide a more user-friendly experience.



Co-funded by
the European Union

2. General Architecture

2.1 Platform Architecture: Overview

The Figure 1 represents the system architecture scheme. The architecture will be based on microservices, hosted in the Microsoft Azure cloud.

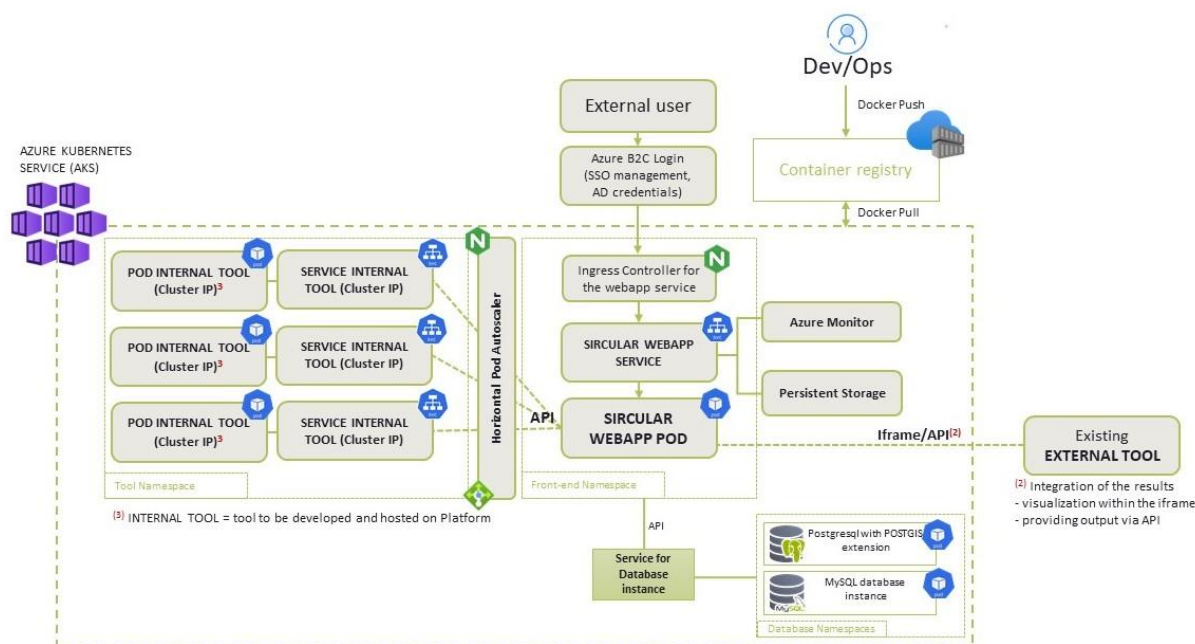


Figure 1 - Architectural diagram

The solution is designed to ensure high scalability, modularity, and interoperability, relying on a container-based deployment approach orchestrated by Azure Kubernetes Service (AKS).

Each functional element of the platform, including both internal tools and core services, is encapsulated within isolated pods and services, grouped by namespaces to facilitate maintainability and logical separation. The use of a Horizontal Pod Autoscaler ensures dynamic scaling based on workload demands, enabling optimal resource utilization.

User authentication and secure access are managed via Azure B2C, providing Single Sign-On (SSO) capabilities and integration with Azure Active Directory credentials. The architecture also includes mechanisms for persistent storage, monitoring, and interaction with external tools through APIs and iframes, allowing seamless data exchange and visualization within the platform environment.

This high-level structure provides the foundation for the various components and interaction flows, which will be detailed in the following.

As part of the platform monitoring strategy, WP2 will provide Task 4.4 with secure and authenticated APIs to access data needed for continuous improvement activities, such as BIM information to enrich sensor data. At the same time, from Task 4.4, connectivity APIs will be provided to feed the tools with monitoring data. These APIs will also expose relevant tool-level metrics and specific KPIs, allowing external systems, e.g. visualisation, to retrieve the necessary information. This configuration enables synchronization between WP2 tools and T4.4 monitoring, allowing the monitoring plan to be implemented efficiently across all demonstration sites.

2.2 Core Components and Interaction Flows

In the SIRCULAR Platform users access through Azure B2C Login, which serves as the sole authorized entry point for authentication. Once successfully authenticated, the Ingress Controller handles routing and presents the appropriate web interface to the user. At this stage, the user is considered to be within the secured application environment and can begin interacting with the various internal tools provided by the system. All interactions are orchestrated through the core backend component known as the SIRCULAR Webapp Service. This service also integrates with key Azure resources such as Azure Monitor (Used for continuous logging and performance monitoring to detect potential issues) and persistent storage, which enables secure file storage and retrieval as part of the backend's functionality.

The SIRCULAR Webapp Service communicates directly with the SIRCULAR Webapp Pod, which is responsible for coordinating interactions with the platform's internal tools. This pod manages calls to various internal services by routing requests to the appropriate tool pods within namespace of the tools. These interactions are dynamically scaled using the Horizontal Pod Autoscaler (HPA). The HPA continuously monitors key performance metrics, such as CPU and memory usage, and automatically adjusts the number of active pod instances based on the current load, ensuring both optimal performance and efficient resource usage.

Through this mechanism, the SIRCULAR Webapp Pod connects users to the individual containers that host each internal tool. Inside these containers, users can access the full range of services provided by each tool. To ensure a seamless and cohesive user experience, all internal tools are required to implement a frontend that closely follows a shared design system and interaction model. This consistency across tools ensures that the UX/UI remains intuitive and familiar, regardless of which tool the user is interacting with. Any communication between an internal tool container and the core application must be conducted via secure internal APIs, which are exposed specifically to handle various interaction scenarios across the system.

The backend core of the application also includes a set of services dedicated to database interactions. Two primary database systems are integrated into the architecture, each serving distinct purposes. The first is a MySQL database, used for general-purpose data storage across



Co-funded by
the European Union

the application and its internal tools. This database handles standard transactional and structured data required for core platform functionality.

The second is a PostgreSQL database enhanced with the PostGIS extension, which allows for advanced geospatial data management. This geospatial database is specifically intended to store and process information extracted from BIM (Building Information Modeling) models, enabling the platform to handle location-aware data and spatial analysis tasks efficiently.

In addition to its internal components, the platform is also capable of integrating external tools hosted outside the main AKS infrastructure. This is achieved by embedding these tools directly into the platform's frontend using iframes. This approach enables users to access and interact with external functionalities within the unified SIRCULAR user interface.

The following Table 2 has been created to provide a summarized overview of the main components that constitute the platform architecture. For each component, a brief description is provided to clarify its role within the system and to support users in understanding its specific function and contribution to the overall infrastructure. In the subsequent paragraphs, each component will be examined in detail to illustrate its functionality and interaction.

Table 2 - System Architecture, main components

COMPONENT	DESCRIPTION
Azure Kubernetes Service (AKS)	A managed service from Microsoft Azure that simplifies the deployment, management, and operations of Kubernetes. In this architecture, AKS serves as the fundamental platform that hosts all application components in containers, automatically managing aspects such as scalability, load balancing, and fault tolerance.
External User	Consumers of the web service who connect via browsers or other client applications
Azure B2C Login	Azure B2C (Business-to-Consumer) is an identity management service that allows users to securely access the application. It manages SSO (Single Sign-On) credentials, Active Directory integration, and other security features for access control



Co-funded by
the European Union

DevOps	Represents both the team and practices used to integrate software development (Dev) with IT operations (Ops), ensuring a continuous flow of updates and improvements to the application
Container Registry	It is a centralized repository where different versions of the application are stored in container format. The DevOps team uploads new versions here (Docker Push) and the infrastructure retrieves them for execution (Docker Pull)
Ingress Controller	Functions as an entry point for all requests directed to the application. It routes HTTP/HTTPS traffic to the appropriate services within the Kubernetes cluster, also managing features such as load balancing and TLS termination
SIRCULAR webapp service	It is the service definition that exposes the SIRCULAR application within the cluster. It acts as an intermediary between the Ingress Controller and the application pods, ensuring that requests are directed to available pods
SIRCULAR webapp POD	A pod is the basic unit in Kubernetes and represents a running instance of the web application. Multiple pods can run simultaneously to handle high workloads or ensure high availability
POD internal tool (Cluster IP)	These internal tools are developed and hosted by RINA + PARTNERS, as indicated in the diagram. They are accessible only within the cluster and provide support services for the management and operation of the main application
Service internal tool (Cluster IP)	These services allow different parts of the application to communicate with internal tools through IP addresses internal to the cluster (Cluster IP), without exposing them externally
Horizontal Pod Autoscaler	Monitors resource utilization and automatically increases or decreases the number of running pods to maintain optimal performance even with variable loads
Tools to be developed	The architectural diagram indicates that internal (to be developed) tools are planned to be developed in the future to integrate new functionalities into the system
Developed tool	Already existing tool (developed) that will be integrate with the application. This tool provides:



Co-funded by
the European Union

	<ul style="list-style-type: none"> • Integration of results • Visualization within the iframe • Output via API
Service for Database Instance	This component coordinates connections to databases, ensuring that the application can access them reliably and securely
PostgreSQL with POSTGIS extension	PostgreSQL is a relational database management system, while POSTGIS is an extension that adds support for geographic objects, allowing location queries and spatial analysis
MySQL database instance	MySQL is another relational database management system used in the architecture, probably for specific functionalities or for compatibility reasons with certain parts of the application
Persistent Storage	Ensures that data remains available even if containers restart. It is used to store files, configurations, and other data that must persist over time
Azure Monitor	Provides monitoring, logging, and alerting capabilities to ensure that the application functions correctly. It helps identify and resolve issues, as well as optimize performance



Co-funded by
the European Union

3. Cloud Infrastructure – Azure Kubernetes Service (AKS)

The SIRCULAR platform has been designed to operate on Microsoft Azure, one of the leading cloud platforms globally, recognized for its high level of security, scalability, and integration with DevOps tools. The infrastructure that will be used is provided by RINA-C. Azure provides a complete ecosystem for the management of containerized infrastructures, offering managed services, such as Azure Kubernetes Service (AKS), that simplify the deployment, management, and monitoring of applications in cloud-native environments. Key features of Azure include:

- Automatic and on-demand scaling of computational resources.
- Native integration with CI/CD tools, such as GitHub Actions and Azure DevOps.
- Managed services for monitoring (Azure Monitor), access control (Azure Active Directory), and secrets management (Azure Key Vault).
- Compliance with international security standards, including ISO 27001, GDPR, and CSA STAR.
- Extensive global network of data centers, which guarantees high availability and low-latency performance.

The use of Azure therefore allows the SIRCULAR platform to benefit from a reliable, modular and European-standard compliant infrastructure, facilitating distributed deployment and collaboration between the various project partners.

The core component of the infrastructure is the Azure Kubernetes Service (AKS) cluster, which hosts the platform's microservices organized in specific namespaces (tool, frontend, database) and supported by automatic scaling mechanisms and continuous monitoring. The following sections describe in detail the configuration of the AKS, the management of namespaces and the implementation of the Horizontal Pod Autoscaler (HPA).



Co-funded by
the European Union

3.1 AKS configuration

The SIRCULAR platform will be hosted, at RINA-C infrastructure, within an Azure Kubernetes Service (AKS) cluster, chosen for its scalability, flexibility and native integration with Azure services. AKS provides a highly available environment, automated node management, and suitable for the deployment of containerized microservices. While AKS offers high scalability and flexibility, we acknowledge that dynamic scaling can lead to increased operational costs if not properly managed. For this reason, cost control and resource optimization will be a key consideration throughout the deployment and maintenance phases. During the platform's evolution, various strategies for cost monitoring and optimization will be evaluated, including the use of Azure-native tools for budgeting, alerts, and usage analytics.

The cluster will be configured to support multiple namespaces, each logically isolated to facilitate the management of the architecture components. The nodes will be managed through automatic scaling groups, which guarantee correct resource allocation based on the workload. The use of standardized Helm Charts will allow the automation of the deployment and update processes of the platform components.

The infrastructure will be protected through Kubernetes network policies (Network Policies) and access to the cluster will be regulated through Azure Active Directory (Azure AD), allowing granular role-based control (RBAC – Role-Based Access Control).

3.2 Namespace management (Tool, Frontend, Database)

The management of namespaces in the AKS cluster of the SIRCULAR platform will be structured to ensure logical separation, security and traceability between the different components of the system.

The main namespaces defined are:

- **Tools:** will host all the pods related to the internal tools of the platform (e.g., Geodatabase Tool, Hygrothermal Tool). Each tool will be containerized and managed independently but orchestrated by the core backend.
- **Frontend:** will contain the pods responsible for the web user interface and related services, including the components of the SIRCULAR Webapp Service and the Ingress Controller for routing.
- **Database:** includes the services related to relational databases (MySQL, PostgreSQL with PostGIS extension). Each database will be exposed only internally to the cluster through services, ensuring exclusive access by authorized components.



Co-funded by
the European Union

The segmentation in namespaces will facilitate the management of security policies, resources (CPU, memory), and will allow effective isolation between independent modules, promoting scalability and maintenance of the system.

3.3 Horizontal Pod Autoscaler

To ensure resource efficiency and automatic scaling of microservices, the SIRCULAR platform will adopt the Horizontal Pod Autoscaler (HPA) as a key component of the AKS infrastructure. The HPA will monitor real-time performance metrics such as CPU and memory usage of active pods and will automatically adjust the number of replicas for each service based on the current load.

This feature is particularly useful for managing the elasticity of containerized tools, hosted in the tool namespace, which can experience peaks in usage when processing BIM models, simulations or LCA analyses. In environments with high traffic variability, the HPA will ensure dynamic adjustment of resources without manual intervention, helping to maintain optimal performance and service continuity.

The HPA configuration will include:

- **Target metrics:** CPU usage thresholds (e.g. 70%) or custom thresholds via Azure Monitor.
- **Scale limits:** minimum and maximum number of replicas for each deployment.
- **Scaling strategies:** decision algorithms based on observation and stabilization periods, to avoid excessively frequent variations.

Integration with Azure Monitor will also allow a detailed view of HPA activity, supporting debugging and fine-tuning of scaling policies for each microservice.



Co-funded by
the European Union

4. Webapp Components

4.1 Ingress Controller

An Ingress Controller is one of the fundamental components of a Kubernetes environment. In the SIRCULAR platform its primary role will be to manage inbound traffic to internal cluster services, acting as a single-entry point for all HTTP and HTTPS requests from external users.

In this context the Ingress Controller is expected to:

- **route incoming requests** to the appropriate microservices, based on predefined rules that map specific URLs to Kubernetes services.
- **handle TLS certificates** to ensure secure HTTPS communication, potentially integrating with certificate management services such as Azure Key Vault or Let's Encrypt.
- **provide load balancing** across multiple instances of a service, improving performance and overall system availability.
- **enforce basic security policies**, such as request filtering, gateway-level authentication, and traffic throttling or IP-based access restrictions.

The implementation will likely rely on either **NGINX** or **Azure Application Gateway**, both widely used and fully compatible with AKS. Ingress rules will be managed through Infrastructure as Code (IaC) practices, allowing teams to version, validate, and roll back configurations as needed during development and deployment.

By adopting this architecture, the SIRCULAR platform will ensure centralized, secure, and flexible access to its web-based services. The Ingress Controller will play a crucial role in enabling interoperability between tools and delivering a streamlined user experience.



Co-funded by
the European Union

4.2 Webapp Pod and Webapp Service

The Webapp Pod and Webapp Service will represent the central access point of the SIRCULAR platform, responsible for managing all interactions between the end users and the tools' frontend and backend components.

The Webapp Pod will contain a web application built using Angular Framework and served through a lightweight web server container. It will expose a responsive and intuitive user interface that allows users to authenticate, upload files (e.g., IFC models), access to each specific internal and external tool's interface. Angular was selected for the development of the SIRCULAR web application due to its robust structure, long-term support by Google, and suitability for building modular, scalable enterprise-level applications.

The Webapp Service will expose the Webapp Pod internally and externally, depending on access requirements. Internally, it will allow seamless communication with backend services and tools hosted in other namespaces. Externally, it will connect to the Ingress Controller, which will route incoming requests based on URL paths and security policies.

The following Table 3 has been created to provide a summarized overview of the main components that constitute the Webapp and Pod.

Table 3 - WEBAPP POD AND WEBAPP SERVICE COMPONENTS

WEBAPP POD AND WEBAPP SERVICE COMPONENTS	DESCRIPTIONS
POD	Is the smallest and basic unit of execution in Kubernetes. It can contain one or more containers that share the same network namespace, the same storage and have the same lifecycle
WEB APP POD	A Webapp Pod is a Pod that runs a web application
SERVICE	A Service in Kubernetes is a resource that exposes a set of Pods through an immutable endpoint. In this way, pods can die and be recreated even with different IPs, but remain reachable
WEB APP SERVICE	A Webapp Service exposes the Webapp Pod (or a group of them, as in a Deployment) inside the cluster (or outside, depending on the type of Service)



Co-funded by
the European Union

4.2.1 Interaction with Internal Tools

A key function of the Webapp component will be to mediate and manage interactions with the internal tools deployed in the tool namespace. This interaction will follow a structured pattern:

- When the user accesses the SIRCULAR platform, he/she will have to authenticate using the Webapp via the Azure B2C service.
- Once the authentication is complete, the Webapp will receive a JWT token from the B2C and will send it to the Webapp Service (backend) which will validate it.
- If the token is confirmed valid, the user will be authorized to access the tools (internal and external).
- Each tool will have its own backend with which its Frontend will communicate.

This orchestration model ensures a clean separation of concerns between the user interface and computational services, while maintaining strong security boundaries using Kubernetes network policies and role-based access controls.

Furthermore, the Webapp Service will act as a proxy for accessing already existing external tools (CERTH's tool) embedded via iFrames. In these cases, session data and user tokens will be passed securely to ensure a consistent user experience across the platform.

Overall, the Webapp Pod and Service will be critical for enabling a unified, secure, and user-friendly access point to the wide range of functionalities offered by the SIRCULAR platform.



Co-funded by
the European Union

4.3 Persistent Storage

Persistent Storage in a Kubernetes environment refers to storage volumes that retain data independently of the lifecycle of the containers using them. This ensures that data remains available even if the associated components are stopped, restarted, or redeployed.

Data persistence within the SIRCULAR platform will be also handled through **Azure Blob Storage**, a scalable and secure object storage solution provided by Microsoft Azure. This service is well-suited for storing unstructured data such as uploaded BIM files (e.g., IFC), simulation outputs, reports, tool-generated datasets, and intermediate artifacts exchanged between components.

Azure Blob Storage offers several advantages that align with the platform's architectural goals:

- **High scalability**, allowing the system to store and retrieve large volumes of files without performance degradation.
- **Integration with Azure Active Directory (AAD)** and role-based access control (RBAC), enabling secure and granular permissions for different services and users.
- **Lifecycle management** policies, which will be leveraged to automate data retention and cost optimization by moving infrequently accessed files to lower-cost storage tiers.
- **Geo-redundant storage (GRS)** for disaster recovery and high availability across regions.

By relying on Azure Blob Storage, the SIRCULAR platform ensures a reliable and performant foundation for data persistence, while also providing the flexibility needed to support a broad range of user interactions and tool functionalities.



4.4 Monitoring with Azure Monitor

To ensure system reliability, performance, and observability, the SIRCULAR platform will adopt **Azure Monitor** as its primary solution for monitoring infrastructure, applications, and services deployed within the Azure Kubernetes Service (AKS) environment.

Azure Monitor provides a comprehensive set of features for collecting, analyzing, and acting on telemetry from both Kubernetes-native components and application-level services. Its integration with AKS allows platform administrators and developers to maintain real-time visibility into the health and behavior of the deployed microservices and tools.

The following capabilities will be enabled as part of the monitoring strategy:

- **Container Insights:** Automatically collects CPU, memory, and network metrics from Kubernetes nodes and pods. This will allow teams to visualize resource usage patterns, detect bottlenecks, and tune resource allocation policies.
- **Log Analytics Workspace:** All logs, including application logs, system events, and custom diagnostics from internal tools, will be aggregated in a centralized workspace. These logs can be queried using Kusto Query Language (KQL) for troubleshooting and auditing purposes.
- **Alerting Rules:** Azure Monitor supports the definition of custom alerts based on thresholds or anomalous behavior (e.g., CPU usage > 90% for 5 minutes, repeated failed API calls). Alerts can trigger notifications via email, Teams, or webhook integrations.
- **Dashboarding and Visualization:** Custom dashboards will be configured to display real-time KPIs, including pod health, request latency, error rates, and tool-specific metrics. These will be used by operators and developers to monitor performance and user experience.



Co-funded by
the European Union

5. User Management and Security

5.1 Access through Azure B2C (SSO and AD credentials)

User authentication within the SIRCULAR platform is handled through Azure Active Directory B2C (Azure B2C), which provides a secure, scalable, and standards-based identity management solution. The authentication flow has already been designed in the SIRCULAR platform's frontend and backend architecture.

Upon successful authentication, Azure B2C will issue a JWT (JSON Web Token), which includes user claims, roles, and an access token. This token will be securely stored on the client side and attached to all subsequent HTTP requests sent to the platform's backend.

The backend, developed using SpringBoot framework, will act as a resource server and is responsible for validating the access tokens received from the frontend. This is achieved by:

1. Verifying the token signature using the public keys provided by Azure B2C's OpenID Connect metadata endpoint.
2. Checking the audience (aud) and issuer (iss) fields to ensure the token was issued for the intended API and tenant.
3. Extracting user roles and claims from the token to enforce role-based access control at the API level.

This architecture will ensure that only authenticated users with valid tokens can access protected resources or interact with internal tools through the frontend and backend services. Role-based policies may also be used to restrict access to specific API endpoints or platform functionalities (e.g., tool access, file upload, admin views).

The adoption of Azure B2C provides the platform with the following benefits:

- Centralized identity management with customizable user flows
- Strong security and compliance with OAuth 2.0 / OpenID Connect standards
- Scalable integration with internal and external user bases
- Seamless **Single Sign-On (SSO)** across the SIRCULAR platform and external tools (if supported).

This secure and modular approach to authentication lays the foundation for managing diverse user roles and permissions across the SIRCULAR ecosystem while ensuring a smooth and user-friendly login experience.



Co-funded by
the European Union

5.2 Secured APIs

As part of the planned architecture for the **SIRCULAR Platform**, secure APIs will play a central role in enabling communication between internal components, user-facing services, and third-party tools.

5.2.1. Authentication and Authorization Strategy

All API access is intended to be governed by OAuth 2.0, providing a secure and standardized method for access control. After authenticating via **Azure B2C**, users will receive access tokens (In JWT format) that will be required for any API interaction. Backend services will be responsible for validating these tokens and enforcing user permissions accordingly.

5.2.2. External Tool Integration and Session Propagation

For external tools that reside outside the core AKS infrastructure, the platform will support a session handoff mechanism. The idea is to redirect users to external tools along with secure tokens that can be verified through dedicated validation endpoints exposed by the main platform. This will allow external systems to recognize existing user sessions and avoid redundant logins, effectively supporting a single sign-on (SSO)-like experience.

Once connected, external tools will continue communicating with the platform via authenticated API calls. These calls will likely be used for data synchronization, session verification, or triggering backend workflows.

5.2.3. Cluster-Scoped Internal APIs

Internally, tools deployed within the Kubernetes cluster will communicate via private APIs using ClusterIP services. These APIs will not be exposed outside the cluster. Access will be controlled using Kubernetes network policies and service identities to restrict traffic between namespaces and services based on predefined rules.

5.2.4. API Structure and Role Separation

The API ecosystem is expected to be modular, with clear separation between:

- System APIs, intended for authenticated users accessing the frontend web interface.
- Internal APIs, dedicated to service-to-service communication within the cluster.
- Partner APIs, exposed to trusted external systems under strict access controls and with usage contracts in place.



Co-funded by
the European Union

All endpoints will implement proper request validation, error handling, and rate limiting from the start, laying the groundwork for reliable production readiness.

5.2.5. Transport Security

All APIs will be served over HTTPS, with TLS certificates managed either through Azure or an integrated certificate manager via the Ingress Controller.



Co-funded by
the European Union

SIRCULAR has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101147412. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

6. Tools

The SIRCULAR platform is designed as a modular system composed of multiple specialized tools, each addressing specific functionalities within the broader application ecosystem. These tools are developed independently but are integrated through shared APIs and a common authentication layer. This section outlines the purpose, scope, and interaction model of each tool, highlighting their contribution to the overall platform experience.

Currently, the tools are divided into two main groups (Figure 2): those already developed and those planned for development during the project. So, the following paragraphs provide a general overview of the tools that will operate within the SIRCULAR platform infrastructure. Each tool is briefly described in terms of its intended functionality and role within the system. For a comprehensive and detailed analysis of the tools, readers are referred to Deliverable D1.3, where this topic is addressed extensively.

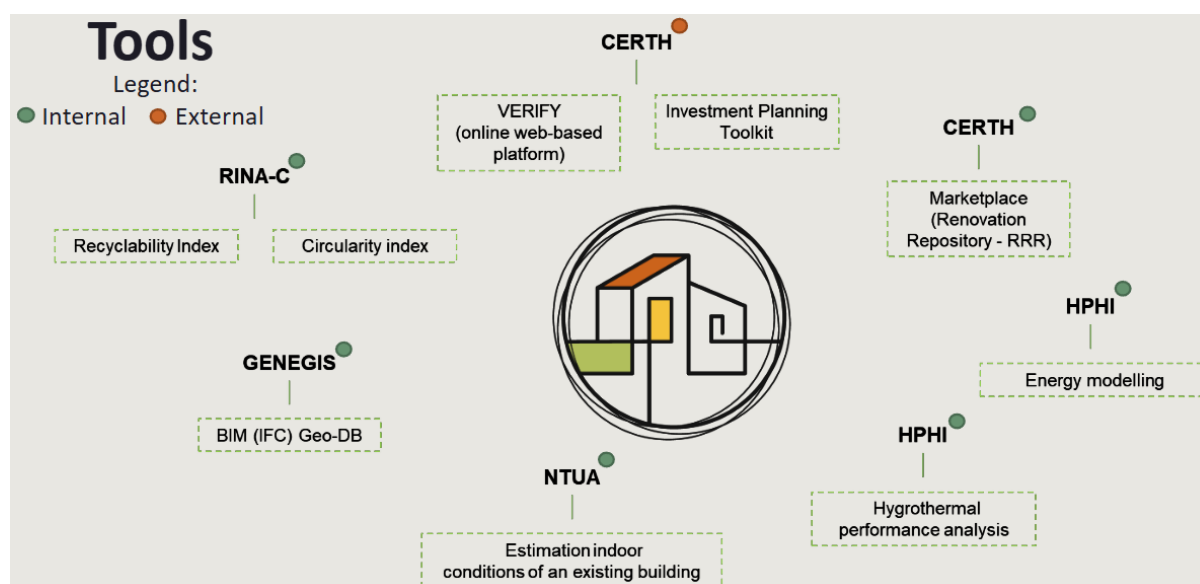


Figure 2 - Tools Internal (to be developed) and External (developed)



Co-funded by
the European Union

6.1 Developed tool

6.1.1 VERIFY - Virtual Platform on Life Cycle Analysis

The VERIFY tool [1], a platform developed by the Centre for Research & Technology Hellas (CERTH), is an advanced digital tool designed to perform comprehensive environmental Life Cycle Assessment (LCA) and economic Life Cycle Costing (LCC) analyses across the full life cycle of buildings and energy systems. Serving both individual buildings and larger-scale urban districts, VERIFY supports data-driven decision-making by integrating real-time data and dynamic performance modeling. VERIFY tool due to its integration with Internet of Things (IoT) technologies enables real-time data collection from active and passive energy systems within buildings or districts. This data collection is processed and translated into dynamic performance indicators, allowing users to monitor, compare, and optimize various design or operational alternatives.

The platform is aligned to international standards (e.g. ISO 14040, ISO 14044, ISO 15686-5) and to the EU Level(s) framework, ensuring its methodological robustness and credibility. Moreover, VERIFY supports BIM-based workflows and can operate with Building Information Modeling (BIM) inputs to enable a fully automated BIM2LCA2LCC process; a process which is fundamental for the SIRCULAR platform functionality and will be developed analytically in SIRCULAR Task 4.2. A core component of the VERIFY platform is its interoperability with INTEMA, CERTH's in-house dynamic modeling and simulation tool. INTEMA supplies VERIFY with accurate energy consumption and storage profiles through high-fidelity simulations or real-time sensor data. This enables the use of white-box (i.e., physics-based model), black-box (i.e., data-driven model), and grey-box (i.e., hybrid model) modeling approaches to forecast performance, optimize system behavior, and conduct techno-economic evaluations. The tools are linked via open APIs, ensuring smooth integration with third-party software and external databases. Moreover, its architecture allows communication with Emissions Trading Systems (ETS) to obtain live carbon pricing and emissions factors, enhancing the temporal accuracy of environmental impact estimations.

Another VERIFY's key feature is the Multi-Criteria Decision Analysis (MCDA), which utilizes trade-offs between energy use, CO₂ emissions, capital and operational expenditures over the system's lifecycle (i.e., typically up to 50 years). The tool is further equipped with automated data validation, cleansing, and interpolation routines, ensuring the integrity of datasets sourced from heterogeneous sensor networks or third-party repositories. This makes VERIFY not only a valuable asset for strategic energy planning and renovation investment analysis, but also a robust computational framework for model-predictive control (MPC) and digital twin development in smart building and smart grid contexts.



Co-funded by
the European Union

VERIFY is an already existing external tool that will be integrated into the SIRCULAR Platform via an iFrame. When users access VERIFY through the SIRCULAR Platform, the tool will open within the same interface, maintaining a seamless user experience. Authentication will be handled directly within VERIFY, offering users to log in through the Project Platform's Single Sign-On (SSO) system (e.g., "Sign in with [Platform Name]").

Beyond authentication, VERIFY plays a key role in enabling BIM-based workflows within the SIRCULAR Platform. It supports fully automated BIM-to-LCA-to-LCC (BIM2LCA2LCC) processes, which are fundamental to the platform's overall functionality. External users connected to the platform can upload IFC files directly through the VERIFY interface. The tool automatically extracts relevant data from the IFC file and allows users to enrich the building model by adding additional components or technologies, such as active systems, through VERIFY's user interface. The extracted and enriched information is then structured according to the project's data models and transmitted to the SIRCULAR Dataspace via API. This structured data becomes accessible to other tools within the ecosystem, both internal and external, enabling integrated, cross-functional use of the information.

Overall, the VERIFY tool stands out as a highly interoperable and scalable solution for life cycle analysis. Its ability to integrate real-time data, support scenario-based evaluation, and deliver robust sustainability metrics makes it a valuable tool for policy makers, engineers, urban planners, and researchers aiming to drive the transition toward a low-carbon, circular built environment.



Co-funded by
the European Union

6.2 Tools to be developed

6.2.1 Geodatabase Tool

GeneGIS will be responsible for developing the GeoDatabase Tool, a component of the SIRCULAR platform that enables users to upload IFC (Industry Foundation Classes) files from supported BIM providers, along with a set of related metadata. One of the most critical pieces of metadata is the geolocation of the model, as spatial referencing is fundamental to the tool's functionality since the models should be geographically traceable.

The tool's backend will be built primarily using Python and Java. Python will be responsible for parsing and validating IFC files, including verifying key attributes such as ThermalConductivity, which is essential for downstream tools that perform simulations or analytical modeling. Moreover, we may consider implementing validation and control mechanisms during file ingestion to prevent duplication and improve data consistency. Java will handle the internal service logic, including API endpoint management and interactions with the database. Additionally, a series of internal endpoints will be developed to facilitate the reception and management of data extracted from the BIM models. The tool will strictly enforce compliance with the IFC 4x3 standard. If a user attempts to upload a file that does not meet the defined standards, a clear and informative error message will be displayed, explaining the reason for the rejection.

On the front-end side, the tool interface will be developed using Vue.js, ensuring a responsive and user-friendly experience for uploading models and entering metadata. The design will adhere to the UX/UI standards established for the SIRCULAR platform, maintaining consistency across tools and providing a seamless experience for end users.

The tool will follow a clear Git-based development workflow, using separate branches for development and production to ensure stability and traceability. Containerization will be implemented using Docker, with the build process packaging all the GeoDB services and dependencies into an isolated container, ready for deployment within the AKS environment. This approach ensures compatibility with the platform's CI/CD and deployment pipelines while supporting future scalability.



Co-funded by
the European Union

6.2.2 Hygrothermal Simulation Tool

The Hygrothermal Simulation Tool being developed by the Hellenic Passive House Institute (HPHI) within the SIRCULAR project is designed to evaluate how material assemblies in buildings perform under varying thermal and moisture conditions. The tool's primary function is to assess the risks of condensation and mold growth by analyzing internal surface temperatures and moisture transfer, using the simplified Glaser method and following international standards such as ISO 13788:2012.

Rather than functioning as a high-resolution dynamic simulation engine, this tool aims to provide reliable, standardized assessments through a more accessible interface. It focuses on monthly steady-state calculations of moisture and temperature profiles within building assemblies, which helps determine their vulnerability to interstitial condensation and mold issues. The tool requires inputs such as material properties, internal and external climate conditions, and surface resistances. Despite relying on simplified calculations, the tool is valuable for early-stage design or quick checks by non-expert users. It allows for evaluating different construction components and verifying their compliance with hygrothermal performance criteria.

The core logic and computational formulas underlying the tool's functionality will be implemented within the backend of the application. On the front-end side, the user interface will be developed using a framework (ex. Vue.js, React), offering a responsive and user-friendly experience for interacting with the tool. The interface design will comply with the UX/UI guidelines defined for the SIRCULAR platform.

From a development perspective, the tool will follow a Git-based workflow, with separate branches for development and production to guarantee controlled releases. Containerization will be achieved using Docker, packaging the application along with all required services and dependencies into an isolated container environment. This container will be fully compatible with the Azure Kubernetes Service (AKS) infrastructure and integrated into the platform's CI/CD pipelines, supporting efficient deployment, maintainability, and future scalability.



Co-funded by
the European Union

6.2.3 Comfort Evaluation Tool

The **Comfort Evaluation Tool** is a full-stack web application designed to facilitate and visualize **building comfort evaluations** based on energy simulation data.

At its core, the platform integrates:

- A **PostgreSQL** database that stores detailed simulation data.
- A **RESTful API**, built with **Django**, which provides structured access to this data for both frontend consumption and external services.

On the client side, a **Single Page Application (SPA)** will be developed using **Vue.js**. Through this interface, users will be able to:

- Create and manage building simulations.
- View and analyze simulation results through key performance indicators (**KPIs**) and visual summaries.

The interface design will comply with the UX/UI guidelines defined for the SIRCULAR platform. To ensure consistent deployment and environment parity, the entire application stack will be containerized using Docker, ensuring full compatibility with the Azure Kubernetes Service (AKS) infrastructure. A CI/CD system will also be implemented to support efficient deployment, maintainability, and future scalability.



Co-funded by
the European Union

6.2.4 CIAT Tool

RINA-C will be responsible for developing the CIAT Tool. The CIAT tool will implement the methodology for evaluating recyclability and circularity indexes, developed starting from T1.2. Its primary function is to provide users with an analytical framework, allowing them to assess the sustainability and environmental impact of buildings from the initial design phase to end-of-life demolition and material recovery. This methodology aims to promote responsible construction practices that align with the principles of circular economy, ensuring that materials and resources are efficiently reused, repurposed, or recycled rather than wasted.

Integration into the SIRCULAR platform is achieved through the use of APIs. These APIs enable data exchange, facilitating dynamic interactions between the CIAT module and the SIRCULAR platform.

The back-end will be developed in Java, using SpringBoot Framework, while the front-end will be developed in Angular. They will adhere to a well-defined Git-based development workflow, employing two distinct branches to maintain stability and traceability: develop and master. Both back-end the front-end will be deployed into the SIRCULAR Azure Kubernetes using Docker for containerization.



Co-funded by
the European Union

6.2.5 Marketplace

The SIRCULAR Marketplace is designed to facilitate sustainable construction practices by automating the generation of material and product databases, and mapping recycling facilities. It gathers data from public repositories, such as Environmental Product Declarations (EPD), to create a structured repository prioritising environmental impact and material circularity. This database is automatically populated using algorithms that classify materials based on attributes such as recyclability, energy efficiency, and lifecycle performance. Additionally, the platform integrates a geo-database derived from BIM models, linking material properties to spatial data. The geo-database supports the identification of recycling plants based on their processing capabilities, proximity to construction sites, and sustainability certifications, offering an efficient resource for material reuse and waste management. Furthermore, apart from allowing users to purchase or sell materials/products, the Marketplace will be able to provide personalised recommendations for optimised and eco-friendly purchases by making use of CIAT tool outputs.

From a technical point of view, the SIRCULAR Marketplace consists of the back-end and the front-end UI modules. For the former, the Marketplace is utilising the MySQL DBMS for storing the data. However, it can be adapted to utilise different DBMS such as PostgreSQL or MongoDB. Regarding the UI, the Marketplace platform is utilising Angular 15 along with SCSS, Angular Material 15, and PrimeNG for UI components, towards providing a user-centric and simple-to-use responsive web-based UI.

From an integration point of view, the Marketplace will interface with the geo-database through a RESTful API, enabling seamless communication between material listings and recycling plant locations. A map-based interface provides a visualization of recycling points, making it easy for users to assess material availability in their area. Moreover, the integration of Single Sign-On (SSO) for secure user login ensures users can access the Marketplace with unified authentication. Additionally, the Marketplace will interact with the CIAT tool via RESTful APIs, in order to provide information about products and materials from the repository and receive CIAT tools lifecycle assessment results for each product.



Co-funded by
the European Union

7. DevOps

7.1 Docker Registry and DevOps workflows (Docker Push/Pull)

It is a centralized repository where different versions of the application are stored in container format. The DevOps team uploads new versions here (Docker Push) and the infrastructure retrieves them for execution (Docker Pull).

To support modular development and deployment of the SIRCULAR platform components, a centralized Docker Registry will be used to store and distribute container images for all internal tools, services, and the general platform web application. The platform will rely on Azure Container Registry (ACR) as its main registry, offering seamless integration with Azure Kubernetes Service (AKS) and secure access controls.

Each platform component (whether a tool, frontend module, or backend service) will be containerized using Docker. These container images will be tagged, versioned, and pushed to ACR using standardized DevOps workflows, once the image is successfully built and passes all checks using static analysis via SonarQube.

7.2 Update pipeline and Automatic deployment

This section outlines the envisioned mechanisms for streamlining the build, validation, and deployment processes of the platform's internal tools. Given that each tool is developed and maintained independently by different partners, the update pipeline must support decentralized development while ensuring centralized control over deployment and quality standards.

To streamline development and ensure consistent deployments across environments, the platform should adopt a two-phase continuous deployment strategy, especially for internally developed tools hosted independently by partners.

The pipeline that will define process of building and publishing images will be detailed in the following of the project.



Co-funded by
the European Union

7.2.1 Phase 1: Tool-Level Continuous Integration and Pre-Deployment Validation

Each internal tool should be maintained in its own Git-based repository. Upon every code update (e.g., a push to a release or main branch), an automated CI process should:

- Build and package the application into a Docker container image.
- Run static code analysis using SonarQube and producing a quality report.
- Store the resulting Docker image and the SonarQube report locally or in a staging registry.
- Notify the hosting partner (RINA) or trigger some automated validation step, where the report is reviewed to ensure it meets predefined quality thresholds.

Only after the report is reviewed and approved, the container image should be allowed to proceed to the deployment phase (i.e., be pushed to the ACR and made eligible for deployment to the AKS cluster).

7.2.2 Phase 2: Partner-Led Deployment to Shared Infrastructure

Once the internal tool's Docker image and SonarQube report have been generated and the report has passed the required quality checks through manual validation, the deployment process proceeds under the responsibility of the hosting partner.

This phase includes:

- Pulling the validated Docker image from the staging registry or directly from a provided artifact (after quality approval).
- Pushing the image into the shared Azure Container Registry (ACR) used by the central platform.
- Deploying the image into the appropriate namespace within the Azure Kubernetes Service (AKS) cluster, using standardized Helm charts or deployment manifests.
- Registering or updating service routes within the platform's internal service mesh, so that the new or updated tool is discoverable and accessible by the SIRCULAR WebApp Service.

The hosting partner (RINA) remains in charge of this controlled deployment step to ensure traceability, infrastructure integrity, and adherence to internal policies.

7.3 Versioning and environment management

SIRCULAR will adopt a structured approach for versioning to ensure consistent and reliable deployments across the software development lifecycle.

7.4.1 Source Code Versioning

All application and tool code should be maintained in **Git-based repositories**, with each development team or partner free to adopt a branching strategy that suits their workflow, as tools will be developed independently. Releases must be tagged using **semantic versioning** (e.g., v1.0.0) to track changes and enable reliable rollbacks when necessary. The versioning convention should follow the standard format: the **first number** indicates major releases, the **second** represents minor updates, and the **third** is reserved for patches or bug fixes.

7.4.2 Container Versioning

Every Docker image produced should be tagged with a unique version identifier. These images will be pushed to the **Azure Container Registry (ACR)** and deployed through CI/CD pipelines, ensuring that each environment runs a specific and traceable version of the code.



Co-funded by
the European Union

8. Database and Data Persistence

8.1 PostgreSQL with PostGIS extension

The platform will work with a **PostgreSQL** database, enhanced with the **PostGIS extension**, to serve as the core **geospatial database**. This setup is intended to support the storage and querying of spatial data, mainly extracted from **IFC models**. During the upload process, users will be able to define geolocation metadata through the user interface, allowing precise positioning of BIM model data within geographic space.

This database will be primarily used by the tool known as **GeoDatabase**, which will expose its content. Other tools deployed inside platform will also be able to query geospatial data through APIs exposed by the GeoDB tool, serving as an abstraction layer that centralizes access and ensures controlled interaction with the geospatial database.

Benefits of Using PostGIS

PostGIS significantly enhances PostgreSQL spatial data management. The key benefits from its use include:

- **Native support for geometric and geographic data types**, such as points, lines, polygons, and multi-geometries.
- **Advanced spatial functions** to perform operations like distance calculations, spatial joins, containment checks, buffering, intersection, and more.
- **Indexing with GiST (Generalized Search Trees)**, which greatly improves the performance of spatial queries.
- **Integration with GIS tools and libraries**, such as QGIS, GeoServer, and Leaflet, allowing seamless visualization and analysis.
- **Coordinate system support**, including transformations between different SRIDs (spatial reference systems).



Co-funded by
the European Union

8.2 MySQL: usage and use cases

8.2.1 CIAT Tool

The CIAT tool will use a series of static parameters (Disassemblability - connection type, Disassemblability - connection accessibility, Disassemblability - Crossings, Disassemblability - Form containment) that will be stored in specific tables in the MySQL database. The opportunity to save the results from the execution of the tool in the same database will also be evaluated.

8.2.2 Comfort Evaluation Tool

From the side of the Comfort Evaluation Tool the bellow data are needed to get retrieved from the database, in order to run a simulation:

Parameter	Type	Description	Possible Values
1. building_shape	Categorical	Shape of the building	square, rectangular
2. room_area	Real Number	Area of the room	
3. exposure_to_wind	Categorical	Degree of building exposure to wind	low, high
4. insulation_level	Boolean	Whether the building is insulated	true, false
5. windows_glazing	Categorical	Type of window glazing	double, single
6. add_insulation	Boolean	Whether insulation is added during renovation	true, false
7. add_windows	Boolean	Whether windows are added during renovation	true, false
8. add_shading	Boolean	Whether shading is added during renovation	true, false
9. add_ventilation	Boolean	Whether ventilation is added during renovation	true, false



Co-funded by
the European Union

9. Conclusions

This deliverable has presented a comprehensive overview of the architecture, infrastructure, and foundational components that underpin the SIRCULAR digital platform. Starting from the rationale behind the selection of the cloud provider and its associated technologies, the document has described the key elements required to ensure a scalable, secure, and interoperable environment capable of supporting a diverse set of tools and user profiles.

Through detailed analysis, the report has illustrated how the platform leverages Microsoft Azure and Kubernetes to manage deployment, container orchestration, and service scalability. It has also outlined the configuration of web application components, data persistence strategies, and monitoring mechanisms that collectively guarantee high performance and reliability.

User management and access control mechanisms, based on Azure B2C and secure API integration, have been defined to ensure a seamless and secure user experience across all tools and services. The document has also provided a structured view of the internal and external tools integrated into the platform, as well as the workflows supporting their continuous deployment, version control, and interaction with shared data services.

The contributions from project partners have been essential in shaping the architecture and integrating specific tools, such as the CIAT Tool, the Hygrothermal Simulation Tool, the Comfort Evaluation Tool, and the Verify Tool. These tools not only extend the platform's capabilities but also reflect the collaborative effort required to ensure a cohesive and robust system.

Finally, this deliverable sets the stage for future developments, particularly those described in D1.3 and the ongoing activities in WP2 and WP4. The described architecture provides a stable foundation [2, 3] upon which further functionalities will be developed, ensuring that the SIRCULAR platform continues to evolve in alignment with the project's goals of promoting circularity, sustainability, and inclusiveness in the construction sector. Moreover, architecture and implementation strategy presented in this deliverable are based on technical choices and best practices derived from both project-internal documents and relevant industry standards, ensuring a robust and well-justified design. Where applicable, design decisions have also been informed by existing frameworks and reference architectures such as ISO standards and widely adopted cloud-native technologies (e.g., Kubernetes orchestration patterns), thus reinforcing the reliability and applicability of the proposed solutions.

As the platform involves multiple tools and partners, data governance, data ownership, and data sharing will be managed in alignment with the principles and agreements defined within



Co-funded by
the European Union

the project framework. These aspects will be further addressed and detailed under WP6, ensuring that all activities comply with applicable regulations and stakeholder expectations.



**Co-funded by
the European Union**

SIRCULAR has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101147412. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

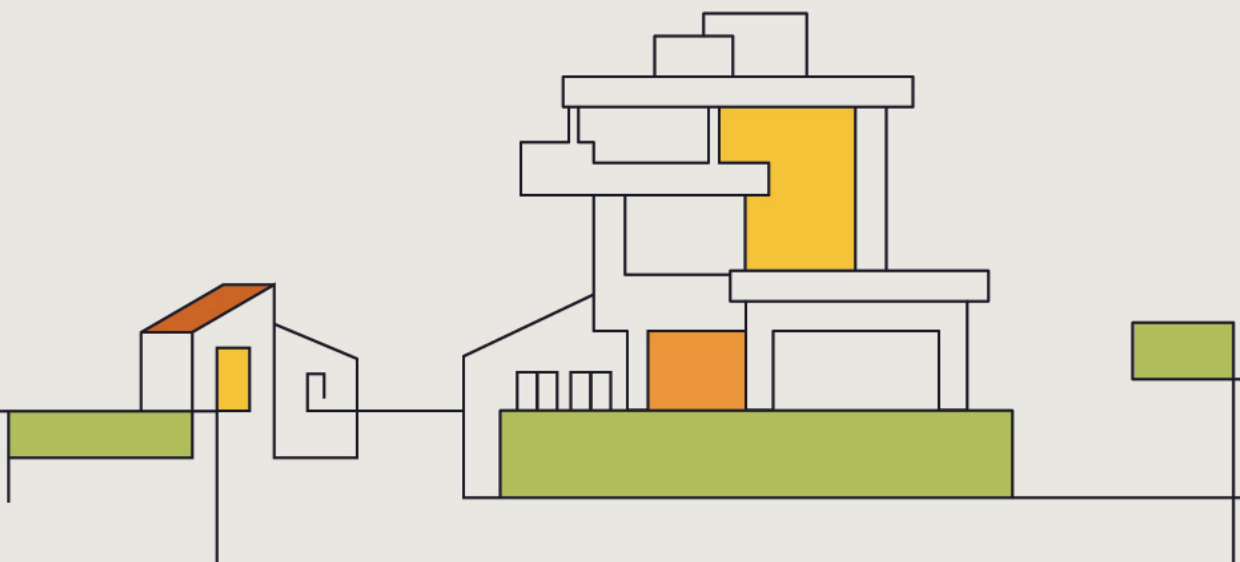
10. References

- [1] Centre for Research and Technology Hellas (CERTH) e Chemical Process & Energy Resources Institute (CPERI). Online: <https://verify.cperi.certh.gr/>.
- [2] Carbon a Azure Kubernetes Service Case Study. Online: <http://www.carbonrmp.com/>
- [3] SIEMEN HEALTHINEERS an Azure Kubernetes Service and Kubernetes Case Study. Online: <https://iamhimy.medium.com/azure-kubernetes-service-aks-industry-use-case-1298d86682b2>



Co-funded by
the European Union

www.sircularproject.eu



@SircularProject



@SIRCULAR Project