HORIZON-CL5-2022-D4-02

EUROPEAN COMMISSION

European Climate, Infrastructure and Environment Executive Agency

Grant agreement no. 101123238

Smart Grid-Efficient Interactive Buildings

Deliverable D1.7

EVELIXIA platform architecture and integration Roadmap

Project acronym	EVELIXIA
Full title	Smart Grid-Efficient Interactive Buildings
Grant agreement number	101123238
Topic identifier	HORIZON-CL5-2022-D4-02-04
Call	HORIZON-CL5-2022-D4-02
Funding scheme	HORIZON Innovation Actions
Project duration	48 months (1 October 2023 – 30 September 2027)
Coordinator	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS (CERTH)
Consortium partners	CERTH, RINA-C, CEA, CIRCE, UBE, HAEE, IESRD, UNIGE, SOLVUS, R2M, EI-JKU, FHB, EEE, EG, ÖE, PINK, TUCN, DEER, TN, ENTECH, SDEF, EGC, KB, AF, Sustain, NEOGRID, MPODOSAKEIO, DHCP, HEDNO, BER, MEISA, ITG, NTTDATA, TUAS, NEOY, HES-SO
Website	https://www.evelixia-project.eu/
Cordis	https://cordis.europa.eu/project/id/101123238

Disclaimer

Copyright Message

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon Europe Framework Programme for Research and Innovation under grant agreement no 101123238. ***Disclaimer:*** *The European Commission is not responsible for any use made of the information contained herein. The content does not necessarily reflect the opinion of the European Commission.*

Deliverable D1.7

EVELIXIA platform architecture and integration Roadmap

Deliverable number	D1.7
Deliverable name	EVELIXIA platform architecture and integration Roadmap
Lead beneficiary	SOLVUS
Description	EVELIXIA End-to-End ICT architecture design, according to the user requirements (T1.1) and the UCs definition (T1.2).
WP	WP1
Related task(s)	T1.1, T1.2, T1.3, T2.4, WP3, WP4
Type	Report
Dissemination level	Public
Delivery date	30.06.2024.
Main authors	Panagiotis Markovits (SOLVUS), Antonis Karantonis (SOLVUS), Miltiadis Anastasiadis (SOLVUS), Iakovos Michailidis (ITI-CERTH), Georgios Vougiatzis (ITI-CERTH), Stavros Koltsios (ITI-CERTH), Panagiotis Klonis, Dimosthenis Ioannidis (ITI)
Contributors	Giorgos Karatzinis (ITI-CERTH), Pablo Carrasco (ITG), Eric Francois (CEA), Alberto José Moreno Pérez (FCIRCE), Paraskevi Giourka (CPERI-CERTH)

Document history

Version	Date	Changes	Author
V1 – first draft	25.01.2024		SOLVUS
V1 – reviews	12.03.2024		CERTH, CIRCE
V1 – consolidated version	8.05.2024		SOLVUS
2nd review	4.06.2024		CERTH,
V2 – second draft	18.06.2024		SOLVUS, CIRCE
3rd review	25.06.2024		NTTDATA, CERTH, IESVE
V3 – third draft	28.06.2024		SOLVUS
4th review	13.07.2024		NTTDATA
V4 – fourth draft	17.07.2024		SOLVUS
Final review	19.07.2024		CERTH
Final version	25.07.2024		SOLVUS
Final deliverable submission	30.07.2024		CERTH

ABBREVIATIONS

Abbreviation	Name
ABDT	Autonomous Building Digital Twin
ADDT	Autonomous District Digital Twin
B2G	Building to Grid
BAFF	Building Awareness and Forecasting Framework
BAUNs	Buildings as Active Utility Nodes
DCB	Data and Context Brokerage
DDT	District Digital Twin
DESCo	Distributed Energy Service Company
DNO	Distribution Network Operator
DR	Demand Response,
DSF	Demand Side Flexibility
DSM	Demand Side Management
DSO	Distribution System Operator
DSR	Demand Side Response
ESCO	Energy Service Company
G2B	Grid to Building
GUI	Graphical User Interface
IS	Innovative Solution
REST API	Representational State Transfer Application Programming Interface
SAREF	(Smart Appliances REFerence (ontology)
SCADA	Supervisory Control and Data Acquisition
SIP:	Stakeholders Interaction Platform
SRA	Scalability and Replicability Analysis

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

The architecture development process has taken into consideration the project's methodology requirements, stakeholders' needs, and the characteristics of the pilot sites to provide a holistic solution and lay the groundwork for the efficient development of EVELIXIA digital platform. An in-depth analysis of the proposed architecture framework is included from various viewpoints to ensure the validity and robustness of the proposed solution.

Beginning with the software architecture design approach, the deliverable presents a brief overview of the most widely adopted processes. The outline of the architecture design procedure is also presented to indicate the procedure that needs to be followed to provide a holistic solution that covers all the aspects of the EVELIXIA project.

EVELIXIA's key innovation and added value, comprise of Innovation Pathways, to provide Buildings as Active Utility Nodes (BAUNs), which are:
a) energy efficient: persistent low energy use minimizes demand on grid resources and infrastructure.
b) connected: two-way communication with flexible technologies, the grid, and occupants is supported.
c) smart: analytics supported by sensors and controls are co-optimize efficiency, flexibility, and occupant preferences; and
d) flexible: flexible loads and distributed generation/storage can be used to reduce, shift, or modulate energy use.

Achieving this state of value-added services and excellence, implies designing and building an EVELIXIA integrated platform based on a robust, solid and well documented architecture which is the actual scope of this deliverable.

EVELIXIA is based on a web multi-tier architecture. Through this architecture, the system hardware and software components are organized into individual modules that can be independently updated or changed (modular system organisation). This way, the system is flexible and can adjust to dynamic changes of the operational environment as well as future technological changes of software/hardware.

Furthermore, this chapter organises the system in 3 layers, namely:

1 Application Services Layer
2 Data Management (Middleware) layer
3 Field layer

EVELIXIA' methodological approach resides on a well-established usage of a 3-layer architecture, thus creating an operational framework that provides broad capabilities of interoperability, expandability and reliable business operation. The proposed solution is based on a 3-layer architecture (field data layer, data management layer, application layer / user web interaction layer). The specific approach allows:

1. Realizing a solution that easily escalates depending on the application needs
2. Distributing operations in distinguishable levels, thus avoiding the uneven burdening of specific resources and, therefore, of the system as a whole
3. Selectively expanding the system and the subsystems without changes in their structure and architecture.

Defining the functional / structural views as well as the architectural elements specifications of EVELIXIA's holistic solution, first System Readiness Level (SRL) and the Integration Readiness Level (IRL) had been assessed to eventually compile a dynamic analysis of the platform interactions. This process defines deployment views and Unified Modelling Language (UML) for a meaningful illustration of data flows among system elements, further facilitating the development activities under WP2, 3, and 4.

The last sections of this deliverable analyse the various aspects of the platform's functionality from four distinct viewpoints: i) Functional, ii) Development, iii) Deployment, and iv) Information. The Functional View individually analyses the operational specifications of the major layers / components within the conceptual architecture framework. Each component breaks down into its sub-components, while there is an analysis of the included functionalities and communication requirements. These requirements are utilized to extract the platform's Information View, which documents all communication channels that need to be defined among the components and elaborates on how the platform's components collaborate to serve a wide set of six Business Services.

The EVELIXIA project is currently in its initial phase, with several key aspects yet to be defined. The next version of the project's platform architecture and integration roadmap deliverable is expected to provide a more detailed and better-validated version of the project's architecture framework and system specifications.

1 INTRODUCTION AND OBJECTIVES

This chapter defines the objectives, scope, context, and structure of the deliverable. It also outlines its relationship with other tasks within the project. To this end D1.4 delivers the EVELIXIA End-to-End ICT architecture design, according to the user requirements (T1.1) and the UCs definition (T1.2). key parts of the architecture include:

- Architecture design approach and Methodology
- Conceptual - Logical Architecture
- Functional view
- Development view
- Deployment view
- Use cases view - UML diagrams.

EVELIXIA relies also on architectural foundations, according to best practices of IDS-RAM (Reference Architecture Model), Data Act, and the European Energy Data Spaces; while capitalizing - upon pilot testing and further analysis - data already allowed freely within the EU (e.g., GAIA-X) [1].

Defining functional requirements of EVELIXIA's holistic solution, System Readiness Level (SRL) and Integration Readiness Level (IRL) of all IS applications / systems were assessed and in correlation with other horizontal modules and systems, a functional analysis of the dynamic aspects of the system components was performed. This process defined Unified Modelling Language (UML) diagrams for a meaningful illustration of data flows among system elements, further facilitating the development activities under WP2, 3, and 4.

The architectural description comprises information about the primary system components' identification, their interaction, and the definition of their external interfaces. Every element of the EVELIXIA framework is described in detail, covering a wide range of functional and technical specifications, in addition to giving a comprehensive overview of the EVELIXIA system design.

Additionally, this deliverable will create a clear roadmap for the integration of the upcoming EVELIXIA Innovation Solutions (ISs). Throughout the rest of the project, by completing tasks and collecting feedback, refinement of various aspects of the system architecture will occur.

1.1 Scope and Context

This deliverable introduces the initial version of EVELIXIA architecture. Some components of the architecture will be verified during the development of the project and validated through pilot running and within this framework it is reasonable that some of the system requirements will be updated. The second and final version of the architecture will be delivered on M30. This deliverable specifies the logical and physical architecture, a set of hardware and software requirements and describes the features of the main components. Moreover, the architecture of the system includes the interfaces of the components and the way they should interact among them. Finally, services related UML diagrams and mock ups are presented.

1.2 Interaction with other Tasks and Work Packages

Deliverable D1.7 concisely presents the EVELIXIA End-to-End ICT architecture design based on input from:

- ✓ End user requirements (T1.1),
- ✓ Use Cases definition (T1.2),
- ✓ WP2,
- ✓ WP3
- ✓ WP4

1.3 Structure of this deliverable

The deliverable is constituted by the following chapters which are interrelated and provide an overall analysis of the EVELIXIA Platform architecture.

1. Chapter Introduction and Objectives
2. Identification of Stakeholders
3. Chapter Architecture design approach and Methodology
4. Chapter Conceptual - Logical Architecture
5. Chapter Functional view
6. Chapter Development view
7. Chapter Deployment view
8. Chapter Use cases Information view - UML diagrams.
9. Chapter Conclusions
10. Chapter References
11. Annex I - IS descriptions

2 IDENTIFICATION OF STAKEHOLDERS AND END USERS

The main stakeholders have been presented in detail in D1.1 (section 1.4). Here is a list of the identified stakeholders with a brief reminder of their characteristics and responsibilities:

1. **Building owners/developers:** They are the initial investors and decision-makers in the development and management of smart buildings. They are responsible for ensuring the implementation of smart technologies.
2. **Building managers/operators**: They oversee the day-to-day operations and management of smart buildings by monitoring and controlling the building's systems.
3. **Service providers**: These are mainly companies that supply and maintain various smart technologies, such as energy management systems, security systems, and IoT devices.
4. **Government agencies/regulatory bodies:** They are responsible for establishing policies, guidelines, and standards related to smart buildings and smart cities to facilitate the adoption of smart technologies, promote sustainability, and ensure the safety and well-being of residents.
5. **Research institutions/academia**: They contribute to the development of new technologies, methodologies, and best practices for smart buildings by conducting research, providing expertise, and collaborating with other stakeholders to improve the efficiency and performance of smart buildings.
6. **Tenants/occupants:** They are the end-users of the smart building, living, or working within the premises. They need to provide feedback for improvements and contribute to energy conservation and sustainable practices.
7. **DSOs/TSOs: DSOs** play a vital role in ensuring the efficient management and distribution of resources in smart buildings acting as utility nodes. TSOs do not have a direct role in smart buildings acting as utility nodes, but they are responsible for maintaining grid stability, ensuring grid integration, and managing emergencies and can therefore indirectly impact the energy management strategies of smart buildings.
8. **Aggregators**: The role of aggregators in such systems is to optimise energy consumption and distribution, facilitate demand response programs, promote renewable energy integration, offer financial incentives, enable energy trading, and contribute to grid stability.

Figure 1 illustrates the mapping between the six EVELIXIA Services and the relevant Stakeholders:

EVELIXIA Service	Relevant Stakeholders
DSM IMPLICIT	Building Manager / Owner / Tenants
Building Investment Planning	Building Manager / Owner / Tenants
DER DISPATCH	DSO / Aggregator
P2P energy (flexibility) trading	Aggregator / DR service provider / Large Industrial customers / Pooled non-industrial customers / Tenants
Portfolio management services	Aggregator / DR service provider
Network Investment Planning	DSO

Figure 1 - Services and Stakeholders mapping

3 ARCHITECTURE DESIGN APPROACH & METHODOLOGY

This section describes the Architectural Design Approach for the EVELIXIA project. The descriptions in this report include the design principles and the architectural design process. The overall design process is divided into four phases, firstly the stakeholder's requirements and market need as well as the service scenarios, moving to the conceptual architecture that describes the workflow of the EVELIXIA project, towards the structural view of the system and finally the detailed architectural elements specification. Each phase is based on the previous one, but the design approach for the System Architecture is an iterative procedure, as new requirements and needs can be identified throughout the project. An overview of the approach is depicted in Figure 2.

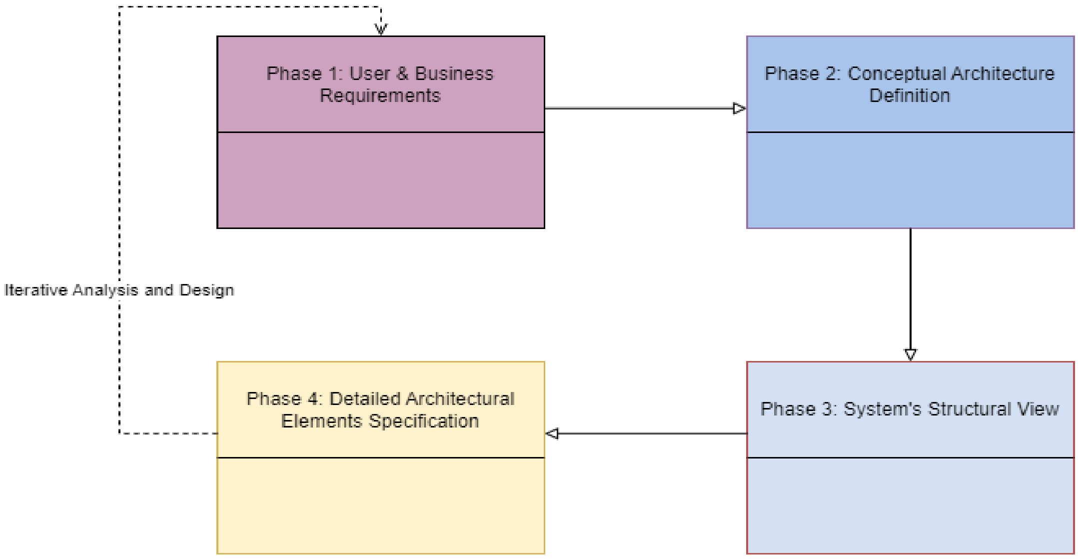


Figure 2 - Design Approach for System Architecture

All the steps and procedures will be introduced for the definition of the EVELIXIA architecture as well as the design methodology that has been adopted. Throughout the whole documentation of this report, basic principles are followed. Thus, the documentation of the first technical guideline of the overall EVELIXIA framework is presented alongside a preliminary understanding of the dependencies, input/output, and specifications of the individual architecture components.

In this report, the technical partners present all the materials and knowledge gathered over the basic tools, solutions and functionalities expected at a sub-component level, creating the complete representation of the entire architecture design. This aspect will enable the further development of the integration of the EVELIXIA Innovation Solutions.

3.1 Used methodology

EVELIXIA' methodological approach resides on a well-established usage of a 3-tier architecture (field layer, data management layer, services layer), thus creating an operational framework that provides broad capabilities of interoperability, expandability and reliable business operation.

The specific approach allows:

1. Realizing a solution that easily escalates depending on the application needs
2. Distributing operations in distinguishable levels, thus avoiding the uneven burdening of specific resources and, therefore, of the system as a whole
3. Selectively expanding the system and the subsystems without changes in their structure and architecture.

The operational model of four logical levels that is applied distinguishes the information system's operations as follows:

1. The data tier is responsible for all commercial systems to be purchased/ used for the project in all pilot sites, the data storage inherent to these systems and the APIs to be created to link the field data layer with the data management layer.
2. The data management layer tier links modules, tools, and processes.
3. The application tier integrates the business logic, in other words, the business rules, which are administered in every application function.
4. The presentation tier is responsible for presenting the data and providing the user interface. (This tier will be part of Tier 2, thus constituting layer 2 of the EVELIXIA Architecture).

To identify the main aspects of successful application implementation, we should point out each key factor that can improve the functionality and, therefore, the effectiveness of the EVELIXIA application. These seven factors that can – ultimately – lead to success is:

Purpose

Solution owners who have a solid understanding of what they expect to get out of their system will be able to work backwards to determine how the application should be managed, which functionality should be included, what messages should be communicated to internal and external users, and really everything involved in developing and running a tech product. The purpose of an application sets the tone for all the activities, and it gives a context that should be used to help in decision-making.

Usability

While design and appearance are also important for the user experience, usability is the top priority. The usability needs of an application will partially depend on the nature of the application.

User Focus

Since software applications are designed to serve users, their needs should be the primary focus during development.

Navigation

SW application navigation affects both usability and accessibility. When developing a new SW application or working on a redesign, navigation should be a primary concern. Most SW applications today use common navigational techniques that are expected by the average user. Typically, the SW application will have a primary navigation menu that will link to the most important pages on the SW application, and other links will be added to the body of the page wherever appropriate. Sitemaps, SW application-wide searches and FAQ pages are all very common, and visitors will look for them when they do not know where else to turn. In addition to providing visitors with an easy way to move through the site, designers often use navigation to create more visually appealing SW applications. Although navigation provides designers with a great opportunity to improve the look of the SW application, attractiveness should not come at the expense of usability and accessibility.

Appearance

The appearance of an interface will be a determining factor in the user experience and the intuitiveness of its offered services.

Clarity

One of the goals during the development of an application should be to achieve clarity that will show users what the SW application offers and in what manner/depth of configuration. Clarity is partly achieved through simplicity, thus one of the reasons for using a minimalistic design is to assist in the level of clarity.

Accessibility

Accessibility and usability are often considered identical because they partly overlap, but they are two distinct issues. An accessible SW application aims to be equally compatible with all supported operating systems/internet browsers. The difficulty with accessibility is the number of factors that are outside of the development team's control.

3.2 Architectural Design Process

As mentioned before, the Design Process consists of four phases. Each phase feeds the next one, but the whole process is an iterative procedure, as new stakeholder's requirements or market needs may occur throughout the project.

3.2.1 Phase 1: User & Business Requirements Definition

The first phase of the architectural design contains the elicitation of EVELIXIA's pilot sites specifications and Use-Cases definition for cross-vector applications and novel market positioning (T1.3). Based on the context of the above-mentioned Task and in cooperation with WP2, WP3 and WP4 a detailed User and Business Requirements description of the overall EVELIXIA Suite and the specifications for each of its key components, modules and their functionalities are delivered.

3.2.2 Phase 2: Conceptual Architecture Definition

Throughout the second phase, a high-level conceptual view of the overall system is defined. The major components are identified along with their high-level requirements, providing the concept behind the implementation of the system. EVELIXIA will organize its core components into the proposed conceptual / technical architecture. Conceptual architecture is described in detail in Chapter 4.

3.2.3 Phase 3: System's Functional / Structural View

The designing approach of the architecture follows the standard IEEE 42010 "Systems and software engineering — Architecture description". This procedure is suitable for the architectural description of software intensive systems. Thus, throughout the third phase, the system is further analysed, following a static and dynamic view analysis, based on three architecture viewpoints and on Use cases. In particular:

- Within the **Static View**, a detailed analysis of the system architecture in distinct components is presented, which defines their interconnection from a functional view (Chapter **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**). Also, a detailed deployment view of the system along with its deployment is presented (Chapter **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**).
- Within the **Dynamic View**, based on the EVELIXIA Services that have been identified (Chapter **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**), the dynamic behaviour of the system is defined. More specifically, the dynamic information flow between the Innovation Solutions (ISs) and associate components is presented (Chapter 8), enhanced with UML based descriptive diagrams.

3.2.4 Phase 4: Detailed Architectural Elements Specification

In this fourth and final step, all the key elements of EVELIXIA suite will be described in detail to provide a deep view on the system and to ensure interoperability between the components. Moreover, an analysis of the components along with their modules and their sub-modules is performed in Chapter **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**. More specifically, at the Functional View there is a detailed presentation of the internals of the components, the interfaces, the data types and as well as their components diagrams with their inputs and outputs.

3.3 Design principles

The architecture of the platform is open, flexible, modular and secure. In that way, suppliers, vendors, users, and other potential actors of the project could capitalise on the distinct functionalities of the system. The architecture is as technology independent as possible and includes generic solutions when key technologies (open source or commercial) are available. The following principles have been specified to ensure that the implemented architecture promotes cost efficiency, extendibility and modularity:

- **<u>Separation of concerns</u>**: The system is divided into distinct features, and each one addresses different concerns. The primary goal is to minimize interaction points, increase cohesion, and low coupling.

- ➢ **Single Responsibility**: Each module of the system handles a specific task and is responsible for a single functionality or even aggregation of cohesive functionality.
- ➢ **Least Knowledge**: Each architectural component has limited knowledge about other components.
- ➢ **Don't repeat yourself**: According to this principle, a specific functionality or intent is implemented in only one component.
- ➢ **Minimize upfront design**: In the first steps of the architecture implementation, only the necessary functionalities and methods are designed, because the design of the architecture is possible to change throughout the time.
- ➢ **Scalability**: The system will be designed to handle many users and high volumes of data. This will be achieved using scalable technologies and architectures, such as cloud computing and microservices.
- ➢ **Security**: The system will ensure the security of user data. This will be achieved using secure protocols, encryption, and blockchain enabled technology.
- ➢ **Interoperability**: The system will be able to interact with various external systems and platforms. This will be achieved using standard protocols and APIs.
- ➢ **User Experience**: The system will provide highly personalized experiences for relevant stakeholders. This will be achieved using user-centric design and personalization technologies.

3.3.1 Atomic Design Principles and Methodology for UI-UX

As web design evolves, the industry is recognizing the importance of developing comprehensive design systems rather than merely assembling individual web pages. Much discussion has focused on creating these design systems, particularly on foundational elements such as colour, typography, grids, texture, and related components. While this approach is crucial, it is important to note that these design aspects are inherently subjective. The proposed method analyses interface components to construct design systems more systematically [2].

In the search for inspiration, parallels were found in chemistry. The concept is that all matter (solid, liquid, gas, simple, complex, etc.) is made up of atoms. These atomic units combine to form molecules, which then combine into more complex organisms, ultimately creating all matter in the universe. Similarly, interfaces consist of smaller components, allowing for the deconstruction of entire interfaces into basic building blocks that can be built up from there. This is the core idea of atomic design [2].

Atomic design is a methodology for developing design systems, comprising five distinct levels:

- Atoms
- Molecules
- Organisms
- Templates
- Pages

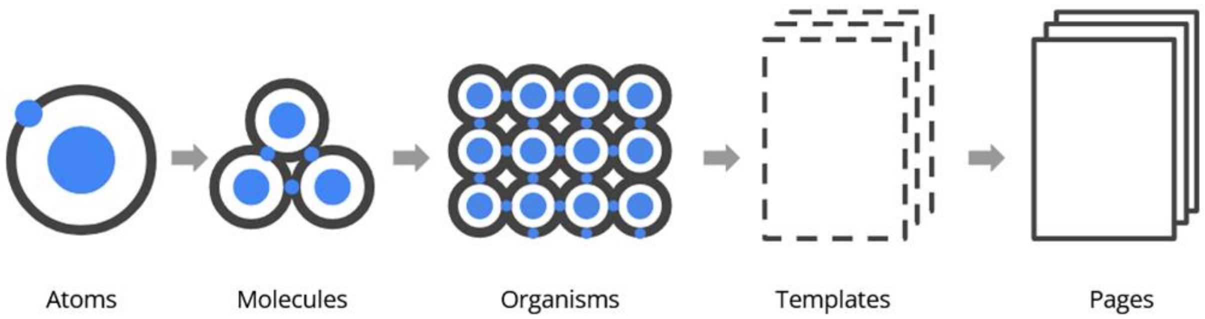


Figure 3 - Atomic design components [2]

Atoms

Atoms are the fundamental components of matter. In the context of web interfaces, atoms are represented by basic HTML tags like form labels, inputs, or buttons. They can also encompass more abstract elements such as colour palettes, fonts, and less visible features like animations. Similar to natural atoms, they are quite abstract and not particularly useful by themselves. However, they serve as a valuable reference in a pattern library, allowing you to see all your global styles at a glance.

Molecules

Molecules are clusters of atoms bonded together, forming the smallest fundamental units of a compound. In design systems, these molecules acquire distinct properties and form the backbone of our design. For instance, a form label, input, or button might not be very useful on their own, but when combined into a form, they become functional together. Constructing molecules from atoms promotes a "do one thing and do it well" mindset. Although molecules can be complex, they are typically simple combinations of atoms designed for reuse.

Organisms

Molecules provide us with the building blocks that we can combine to create organisms. Organisms are collections of molecules joined together to form relatively complex and distinct sections of an interface, making the design more concrete. While someone might not be very interested in the molecules of a design system, organisms reveal the final interface taking shape. Organisms can include similar or different types of molecules. For instance, a masthead organism might include various components like a logo, primary navigation, search form, and a list of social media channels. In contrast, a "product grid" organism might consist of the same molecule repeated, each containing a product image, title, and price. Building from molecules to organisms promotes the creation of standalone, portable, and reusable components.

Templates

At the template stage, the analogy to chemistry is set aside in favour of language that resonates more with clients and aligns with the final output. Templates are primarily composed of groups of organisms assembled to form pages. This is where the design begins to take shape, and layout becomes visible in action. Templates are highly concrete and provide context to the previously abstract molecules and organisms. This stage is where clients start to see the final design come together. Following this methodology, templates begin as HTML wireframes and gradually increase in fidelity until they become the final deliverable.

Pages

Pages are specific instances of templates. At this stage, placeholder content is replaced with real representative content to accurately depict what users will ultimately see. Pages represent the highest level of fidelity, and because they are the most tangible, they are where most of the process participants focus their time and where most reviews are centered. The page stage is crucial for testing the effectiveness of the design system. Viewing everything in context allows for modifications to molecules, organisms, and templates to better address the real-world application of the design.

Pages are also where template variations are tested. For example, it is important to demonstrate how a headline with 40 characters looks compared to one with 340 characters, or to show the difference between a shopping cart with one item versus one with 10 items and a discount code applied. These specific instances help refine and enhance the design system by providing valuable feedback for iterative improvements.

Why Atomic Design

In many ways, this is how the design process has been approached all along, even if it wasn't consciously framed in these specific terms. Atomic design offers a clear methodology for developing design systems. By presenting the steps in this manner, clients and team members can better understand and appreciate the concept of design systems.

This approach also allows for a transition from abstract concepts to concrete implementations. As a result, it helps create systems that are consistent and scalable while also showcasing elements in their final context. By focusing on assembling components rather than deconstructing them, a well-organized system is crafted from the start, rather than selecting patterns in an ad-hoc manner.

Usability

The adage "A picture is worth a thousand words" suggests that some complex concepts can be more effectively communicated through images rather than text. People, including children, can process large amounts of information quickly through visual means. However, it's crucial to ensure that the data visualization is both usable and useful for the user.

The main goal of data visualization is to convey quantitative information clearly and effectively, which requires simplicity. The visualization should only include the data necessary for the specific situation. Understanding how people interpret visual images is also important. Key principles to consider include:

- Proximity: Items placed close together are perceived as related, while those further apart are seen as less connected.
- Similarity: Objects that look alike are perceived as belonging to the same group. Similarities in colour, shape, size, and orientation can imply a relationship.
- Closure: People tend to complete incomplete shapes based on familiarity.
- Continuation: Objects perceived as moving in a certain direction are seen as continuing in that direction.
- Figure/ground: Elements are viewed as either figures (main focus) or background. High contrast between foreground and background enhances clarity.

When designing data visualizations, it's essential to use components familiar to users. Understanding the context and users' workflows helps in selecting appropriate visualizations. Familiar symbols, icons, and colours should be used consistently.

Colours in visualizations should be used cautiously. While they can enhance the appearance of a graphic, they should not be distracting. It's best to use a limited colour palette and ensure colours are accessible to those with colour blindness. Tools like Coblis and Vischeck can simulate colour blindness effects to test visibility.

The software platform should be user-friendly, saving time and being intuitive for all users. Ideally, interactions should be self-explanatory and similar to commonly used tools like smartphones. The system should:

- Provide clear, multilingual explanations that are understandable to users of varying technical proficiency, including children and educators.
- Feature minimal, self-explanatory buttons, similar to the “Apps” button on a smartphone.
- Use illustrative, clear, and abstract images where possible, minimizing numerical classifications.
- Present a simple, well-structured interface with essential information upfront, while allowing for deeper exploration through sub-datasets to access more detailed information about the EVELIXIA framework.

3.4 Bottom-Up and Top-Down Processes

First, the preliminary six (6) Business Services (BS) and Technical Cases were documented to accurately capture and depict the necessary aspects of the EVELIXIA architecture. This version presents the refined content based on inputs from technical work packages and partners’ Innovation Solutions. The architecture definition process in the first version involved a bottom-up and top-down approach, exploring technologies and identifying core functionalities. Partners provided templates for describing the Innovative Solutions (presented in Annex 10.1) and initial plans for EVELIXIA components, while a set of SIX (6) Business Services and Technical Cases was drafted. These allowed the top-down phase to be initiated towards more clearly defining the functionalities required for meeting the project’s objectives.

3.4.1 Bottom-up Process

This phase aimed to collect and categorize technologies and software components introduced by individual partners of the EVELIXIA project. Partners were asked to provide information on relevant tools and updates using templates, and their expertise was utilized in the process. The task also involved identifying and defining Business Services and Technical Cases.

3.4.2 Top-down Process

During this phase, the focus was on defining individual and integrated functionalities to meet project objectives and address Business Services needs and composition based on EVELIXIA Innovative Solutions. This phase helped to define core functionalities, component interaction, and sequence diagrams for each technical case.

4 CONCEPTUAL - LOGICAL ARCHITECTURE

This chapter describes the conceptual architecture of the EVELIXIA system, providing a high-level view of the technical architecture of the system. EVELIXIA is based on a web multi-tier architecture. Through this architecture, the system hardware and software components are organized into individual modules that can be independently updated or changed (modular system organisation). This way, the system is flexible and can adjust to dynamic changes of the operational environment as well as future technological changes of software/hardware.

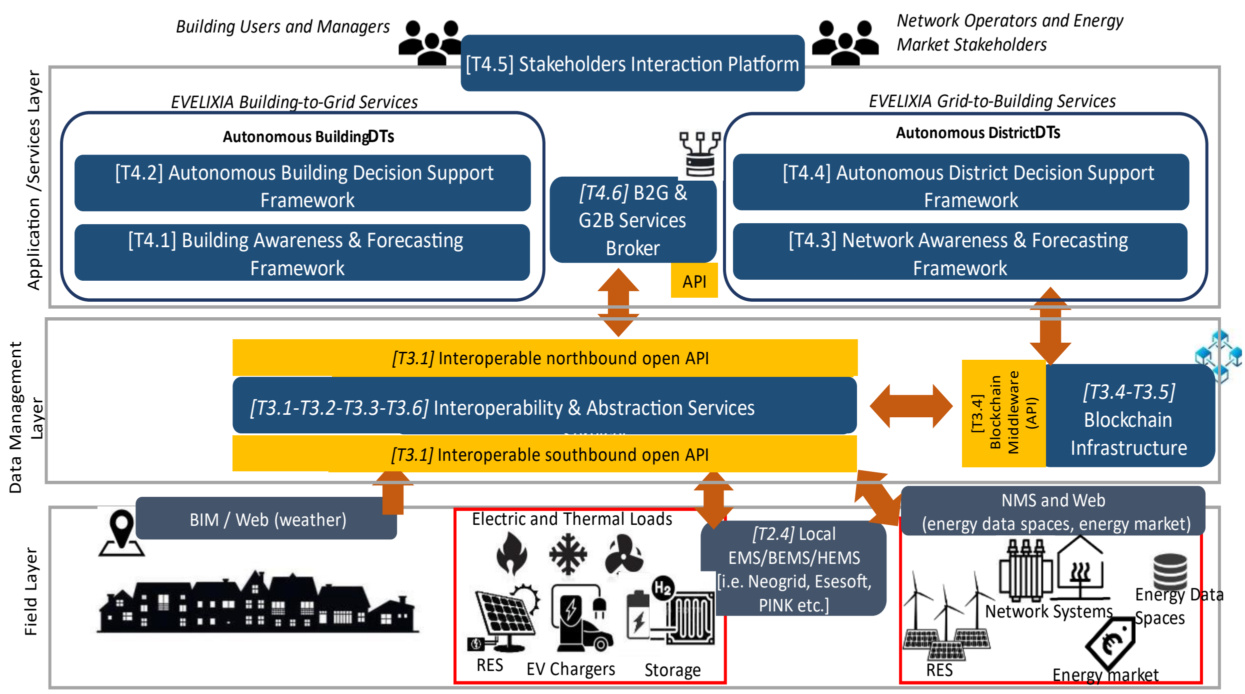


Figure 4 – Overall EVELIXIA Conceptual Architecture

Furthermore, this chapter organises the system in four layers, namely:

- ➢ User Web Interaction Layer,
- ➢ Application Services Layer,
- ➢ Data Management (Middleware) Layer
- ➢ Field Layer

❖ **User Web Interaction Layer**: It contains all the necessary modules needed for the interaction / communication between users / stakeholders and the system as well as the communication among the system's modules.

- **Application Services Layer**: This layer provides the envisioned services and the Innovative Solutions (IS) which will assist users to make decision during the project operation, as well as to improve their everyday life.
- **Data Management Layer**: EVELIXIA's middleware layer will serve as the communication gateway, responsible to effectively bridging the gaps between the highly elaborated services foreseen within EVELIXIA's service layer and the field layer. Moreover, dedicated identity management and security mechanisms will be coupled to the platform to ensure authorized and highly encrypted Southbound (field), and Northbound (services) communication based on Distributed Ledger technologies (BLOCKCHAIN). Finally, dedicated security and transparency BLOCKCHAIN mechanisms will be developed to allow energy smart contracting schemes between buildings and interconnected grids, both at the building and district level. The layer provides the Middleware that constitutes the glue of the system.
- **Field Data Layer**: This layer contains all the physical devices to be deployed, which provides the necessary information from the physical world to the EVELIXIA framework.

The conceptually logical architecture of the EVELIXIA platform, organized in four layers as mentioned above, is depicted in the Figure 5.

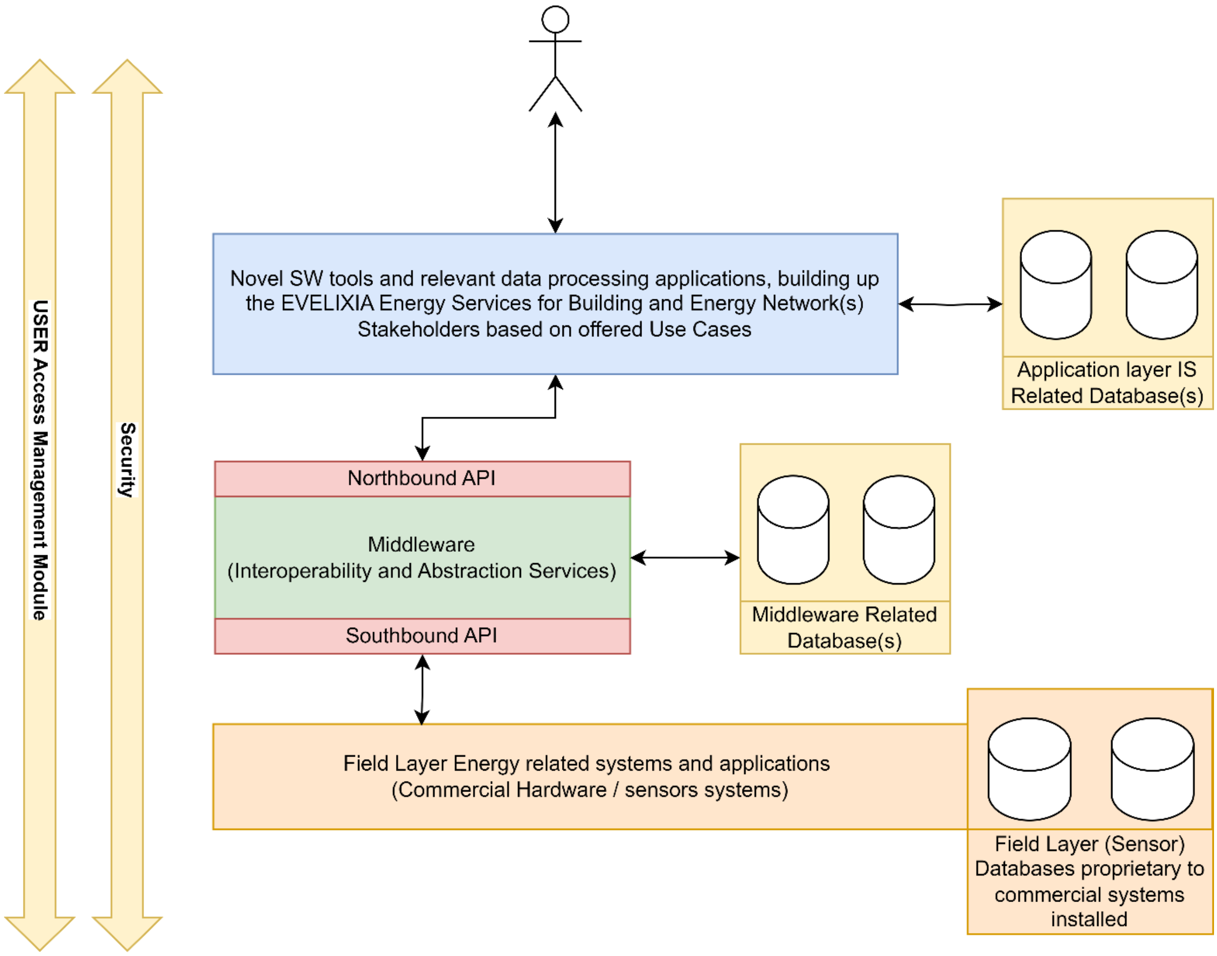


Figure 5 - Conceptual Logical Architecture organized in layers

The logical architecture – apart from the 4 layers - is complemented with the following vertical applied layers:

- **Security:** It is related to security infrastructure which protects the offered platform. This security will be common for the entire architecture and will treat all issues regarding user access, automated provision of user rights, data encryption, data security and system security in a common way.

  A key part of security is Enterprise Management. It is related to functionality that allows administrators to supervise the operation of all architecture tiers under the common environment and proceed to administrative changes and tasks of problems' detections.

- **User Access Management**

  The user management implementation is for streamlining the user login process and to automate administrative tasks such as creating users and assigning them to groups. The proposed solution through Blockchain includes, but is not limited to, the following functionalities:

- Option of storing enhanced user related information
- Assignment of each user-to-user groups according to their accessible services

4.1 Application Services Layer

This layer comprises:

- ❖ The business logic of the system which will be offered through well-known and accredited software frameworks.
- ❖ The central infrastructure of frontend / web pages to handle all user centric applications as well as user admin functionalities. Identities' management, authentication, authorization and security of applications and systems and the related infrastructure that is required.
- ❖ Several logically separate, communicating containers that are implemented in dedicated tasks and subtasks. Each container comprises multiple modules each of which provides a specific functionality. Each module consists of multiple components; each component implements a subset of the module's business logic.

4.2 Data Management Layer

This is the interoperability layer, corresponding to the Interoperability and Abstraction Services developed in EVELIXIA. The common data modelling framework will be delivered along with relevant data ingestion and brokering services for data harmonization, filtering, and event management from various sources.

The Blockchain Infrastructure will be part of this layer, as the main enabler of EVELIXIA Services value extraction and marketability, secure authentication, distributed information logging and verification. The key advantage of this layer is that it allows different applications to communicate with each other by acting as a transit system for carrying data between applications within your enterprise or across the Internet.

4.3 Field Layer

This layer includes all pilot sites' HW equipment (legacy and novel solutions) that interact with EVELIXIA Platform. The various off-the-shelf smart controllers, IoT sensors and devices which will be installed during the demonstration activities will aim to:

i) increase the controllability and observability of (mainly) the thermostatically controllable loads (TCLs) having a significant energy flexibility potential, and

ii) showcase the Platform's interoperability with a variety of solutions from different vendors by adopting different protocols (indicative OSI protocols would include REST, MQTT, HTTP, BACnet, Bluetooth, (w)M-Bus, LoRa Modbus, CAN, RS485, I2C, SMTP, FTP, SFTP, with cycle times between 1ms-20ms).

Appropriately designed APIs will enable the connection with the EVELIXIA platform (Data Management Layer) and the necessary data ingestion from the connected grids.

4.4 Horizontal Modules

4.4.1 EVELIXIA Databases

EVELIXIA will comprise of several databases and varying datasets. As per the Logical Architecture diagram we have 3 set of databases:

i) The databases that accompany systems in the field layer and are proprietary to the companies offering these systems for the pilots,

ii) the database that is for processed data in the middleware layer and

iii) the database that is on the application layer for the energy services and will store specific data.

iv) the database for user data and login mechanisms

For cases ii) and iii) we propose a mixture of SQL and NoSQL databases like Mongo which is characterized by its high efficiency, availability, security and manageability.

For case iv) we propose MariaDB. Also, in case iv), a Database Server is proposed to be deployed for User Accounts, Policies and LogDB.

Database infrastructure relates to storage systems and to information management regardless of whether this relates to transactional data, master data, or data of the offered system databases. The subsystems of the Application Tier will be able to share the common data models and the common data infrastructure to each corresponding database.

4.4.2 Enterprise Security

It is related to security infrastructure which protects the offered system. This security will be common for the entire architecture and will treat all issues regarding user access, automated provision of user rights, data encryption, data security and system security in a common way.

For the support of the user management system and the secure access to the system, Oauth2 with SQL connectivity is recommended. Therefore, the user management implementation will be based on Oauth2 to streamline the user login process and to automate administrative tasks such as creating users and assigning them to groups. The proposed product includes, but is not limited to, the following functionalities:

- Option of storing enhanced user related information
- Assignment of each user-to-user groups according to their accessible services
- Assignment of user groups to proxy access lists

To support wide adoption and usability, offering API tokens (e.g., JWT) and SSO is of high importance for both EVELIXIA partners as well as the Stakeholders communities.

For the support of the user management system and secure access to the system, Oauth2 with connectivity to the chosen database (all well-known databases are supported) is recommended. Therefore, the user management implementation will be based on Oauth2 to streamline the user login process and to automate administrative tasks such as creating users and assigning them to groups. The proposed product includes, but is not limited to, the following functionalities:

- Option of storing enhanced user-related information
- Assignment of each user-to-user group according to their accessible services
- Assignment of user groups to proxy access lists

4.4.3 User Management

After defining the EVELIXA Services and respective ISs they rely upon, together with business flows, the next most important feature of the platform is the functionality of mapping each user / stakeholder to each part of the model needed.

Users are all physical-legal entities (personnel, energy providers, building administrators, etc.) that interact with the platform. For each possible user type there are specific management tasks and access rights so as to be possible to capture all data related to the specific user account and its access rights. Specific capabilities available through this component are:

1. Management of specific personal data (name, surname, phone numbers, email etc.)
2. Management of access data (username, password, connection type etc.)
3. Mapping of the user to specific job descriptions, roles, tasks and actions.

By defining such data in compliance to each EVELIXIA Service then the user acquires access to the workstation by ascending the document access rights defined for each job description, role, task or action that is mapped to.

As far as the legal-technical entities are concerned then their management does not differ at all from the management of any other entity. The only specific part is that their functionality is extended by using platform infrastructure customized for each case and the access rights (platform administrators) are differing in this case.

4.4.4 Enterprise Management:

It is related to functionality that allows administrators to supervise the operation of all architecture tiers under the common environment and proceed to administrative changes and tasks of problems' detections.

The proposed system architecture is based on international standards and is open to many platforms and technologies. Specifically, the standards adhered to are:

- J2EE
- Web Services support

- MariaDB Relational Database system
- Data exchange between different data sources with the use of Web Services (https, REST)
- Support of Java Spring Boot, Node.JS and React
- Improved data access
- Improved security: components that belong to the Application/Business Tier can be secured using a centralised infrastructure. Access to each component can be given or denied separately, facilitating the work of administrators
- Simplified access to external resources

We propose CentOS and Ubuntu as the operating system environment of the system infrastructure but any operating system that can run a Java Virtual Machine (JVM) can be used. The system will be implemented on Java Spring boot framework technology and make use of many modern widely used open-source Java frameworks.

4.5 Mapping of ISs to Architecture Layers

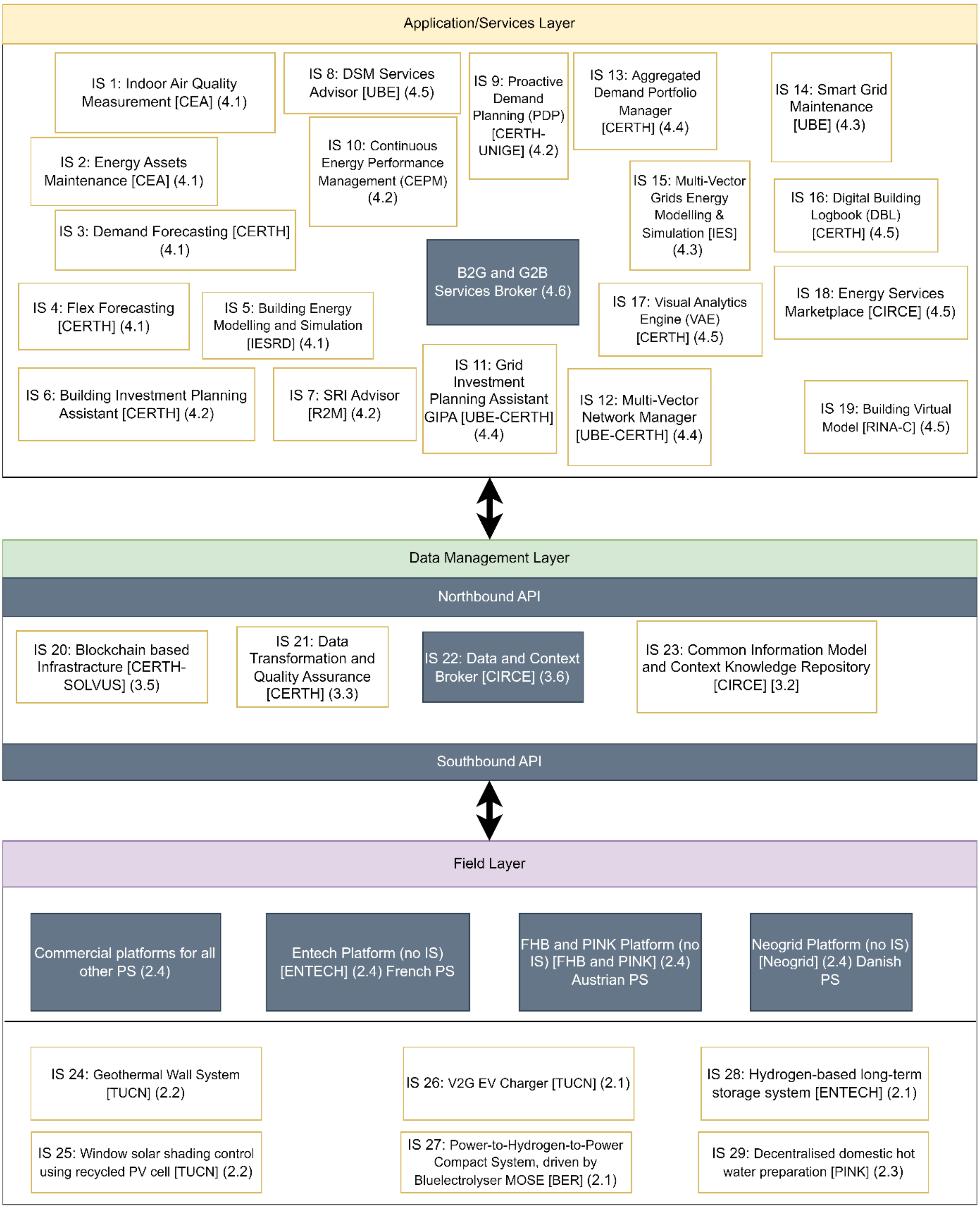


Figure 6 - EVELIXIA Innovative Solutions

4.6 Interconnectivity – Interoperability

Interoperability is widely used in EVELIXIA, allowing:

- Integration of data field sensors, energy equipment and commercial applications with the data management layer
- Integration of ISs between them, thus enabling the usage of the EVELIXIA 6 Innovative Services
- Integration between layers and modules.

Interoperability wide development and operation in EVELIXIA will be done through REST APIs that are mainly enabled through the data / context broker on the data management layer and the services broker on the application services layer.

EVELIXIA also relies on architectural foundations, according to best practices of IDS-RAM (Reference Architecture Model) [3], [4], Data Act, and the European Energy Data Spaces adapting them to the EVELIXIA design and implementation (SAREF) principles.

EVELIXIA platform follows the IDS-RAM principles and guidelines namely:

- It focuses on creating a decentralized, open infrastructure for sovereign data exchange.
- Participants in EVELIXIA are aware of and in control of the data they produce and consume, as well as the associated services (hardware and software).
- The platform emphasizes on data sovereignty, security, and trust.
- It provides detailed specifications for necessary components and their requirements

Furthermore, EVELIXIA architecture complies with the IDS - RAM logical architecture layers, namely: business, functional, information, process and system which are mapped and customized for EVELIXIA and the services to be offered.

Finally, EVELIXIA will capitalize – whenever possible - on data already allowed freely within the EU (e.g., GAIA-X [1], [4]).

Functional / Structural View and Architectural Elements Specification

The current chapter presents the overall functional architecture of the EVELIXIA framework along with a short description of its main components, modules, and sub-modules as well as their functionalities and interactions with the rest of the platform.

4.7 EVELIXIA Functional Architecture and overall flows

The overall functional viewpoint of architecture is depicted in Figure 7.

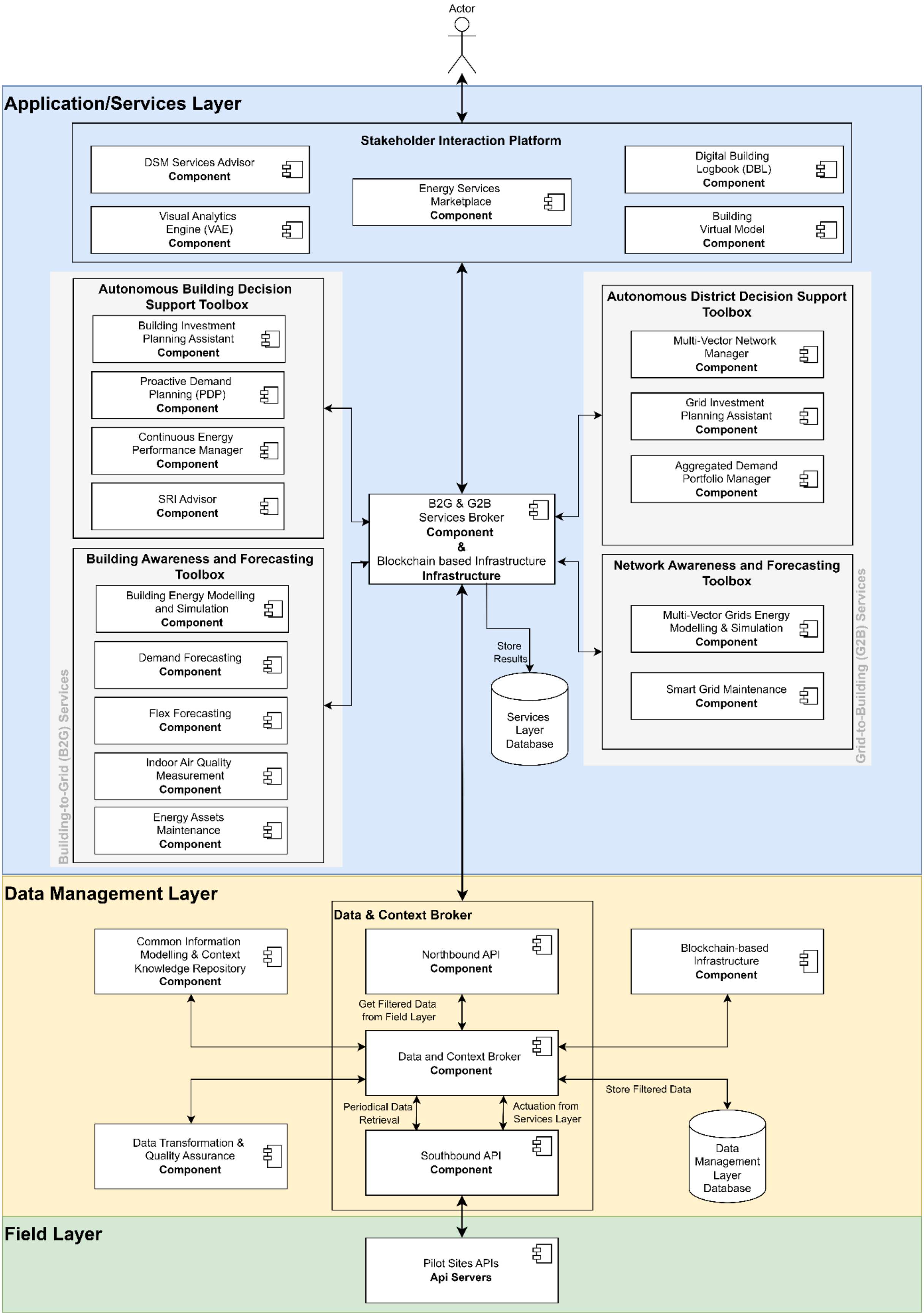


Figure 7 - Overall Functional View

Flows in EVELIXIA platform are bidirectional namely: bottom - up from field layer to application layer and top - down from application layer to field layer as actuation is a key operational function.

Outbound Flow (from Applications/Service Layer to Field Layer):

1. **Applications/Service Layer**: Various applications and microservices in the Application Layer generate data or events. This could include data from smart energy systems, building digital twins, forecasting tools, decision support systems and other services within EVELIXIA. These applications will act as producers, sending data to the Service Broker. Acting as an intermediary, the Service Broker handles real-time ingestion, caching the data appropriately for a predefined period, and thus making it available for consumption for other components. This data is published to specific topics that are carefully configured based on their type. A Kafka Connect source connector (or a custom application?) acts as the endpoint for the Service Layer, consuming the messages that signal the actuation process and then transmitting them to the Northbound API of the Middleware Layer. It is important for the Service Layer Endpoint to format and serialize the actuation messages appropriately.

2. **Data Management Layer (Middleware Layer)**: The request is sent to the Northbound Open API connector of the Data & Context Broker. The Data Broker processes the request and interacts with the Blockchain-based Infrastructure for additional security measures. The Blockchain-enabled Identity, Authorization, and Verification services allow users and assets to prove their identities and acquire access permissions on services/resources based on their roles or attributes. The secured request is then sent to the Southbound Open API connector.

3. **Field Layer**: The secured request is sent to the field layer, where it interacts **(actuates)** with the off the shelf software applications, hardware equipment and IoT devices installed at the pilot sites.

Inbound Flow (from Field Layer to Applications/Service Layer):

1. **Field Layer**: Raw data is collected from various sources such as sensors and Building Energy Management Systems (BEMS). This data is exposed via an API and can be accessed in near real-time or as historical data.

2. **Data Management Layer (Middleware Layer)**: The raw data is received by the Southbound Open API connector of the Data & Context Broker. The Data Transformation and Quality Assurance Module applies machine learning techniques to filter, impute missing values, and generate "healed data". The Data Broker processes the healed data and interacts with the Blockchain-based Infrastructure for additional security measures. The secured, processed data is sent to the Interoperability Service of the Common Information Modelling and Context Knowledge Repository, where it is converted to the SAREF4BLDG ontology in JSON-LD format. The Abstraction Service then generates and maintains an updated knowledge graph that receives queries from a query engine. The query engine uses semantic technologies to interpret and retrieve data and metadata based on its meaning, rather than its syntax or structure.

3. **Applications/Service Layer**: The endpoint of the Service Layer is effectively configured to receive the transformed and formatted/processed data from the Northbound API and publish them to appropriate topics. The Service Broker ensures that the data is durably cached and available for consumption by the different ISs and Components of the Service Layer. These applications and services proceed to consume the data, either in real-time or in a scheduled manner, process it according to business logic, and utilize it for various purposes such as real-time forecasting, recommendations, analytics and notifications.

4.8 Overview of the main components, modules, and sub-modules

This section presents the key components, modules, and sub-modules of the EVELIXIA platform and their main functionalities, and interactions.

4.8.1 Application Services Layer

The Application & Services Layer in EVELIXIA encompasses a variety of novel software tools and data processing capabilities designed to support the agreed upon Energy Services for Building and Energy Network stakeholders [reference]. These toolkits are integrated within the Autonomous Building and District Digital Twins (ABDT, ADTT) framework. An overview of the toolkits is described in more detail in later sections within this chapter of this document.

The key toolkits of this Layer are:

- **EVELIXIA's Building Awareness and Forecasting Toolbox**
- **EVELIXIA's Autonomous Building Decision Support Toolbox**
- **EVELIXIA's Network Awareness and Forecasting Toolbox**
- **EVELIXIA's Autonomous District Decision Support Toolbox**

Complimented by 2 more modules:

- **EVELIXIA's Stakeholders Interaction Platform**
- **Service Broker Integrated B2G and G2B Services Layer**

The tools that comprise each toolkit, referred as Innovative Solutions (IS), are designed to function independently from one another, each aiding in a different way in the successful delivery of the Energy Services.

The Energy Services can be utilized by the end-users and the other stakeholders through EVELIXIA's Stakeholder Interaction Platform, which will act as the frontend of the whole EVELIXIA Platform.

Given that the majority of the ISs utilize big data from diverse sources and operate on different timeframes (e.g., day-ahead forecasting vs. real-time power redistribution) to produce reliable results, there is a significant need for intelligent data coordination and management.

To address the complexity and reduce the computational burden, the B2G & G2B Services Broker (Service Broker) is developed. This broker acts as a mediator and message distribution manager, ensuring seamless communication between different services within the platform.

The Service Layer will leverage Apache Kafka as the core technology for real-time data streaming and processing. The Service Broker will temporarily cache data produced by the various toolkits of the Service Layer. To facilitate efficient and scalable long-term storage, the Service Layer periodically pushes timestamped data to a database system.

A high-level functional architecture diagram for the Applications Service Layer is presented in the Figure 8 and is further analysed in the subsequent chapters.

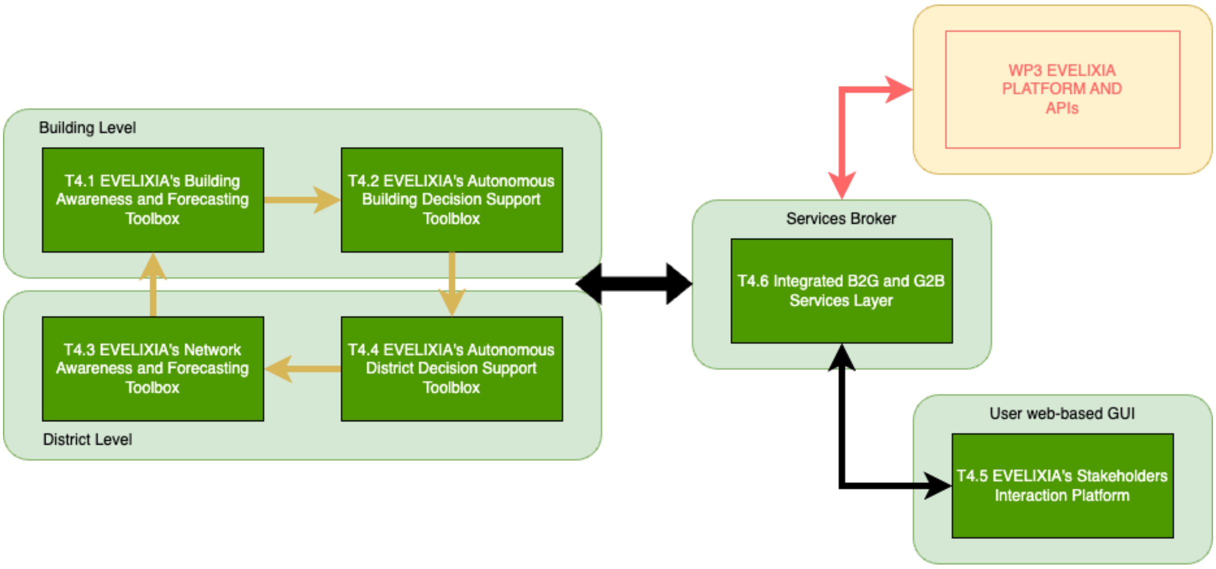


Figure 8 – Application Services High Level Architecture

This is a high-level perception of the architecture from a functional point of view for the Application Services layer. The two pairs of associate Application T4.1 - T4.2, and T4.3 - T4.4 as presented on the diagram include similar conceptual solutions, but from the building and district level, respectively. These pairs of applications are constituted by different sub-services formulating a strong interaction within the Application Services Layer ecosystem. Thus, the output from one Innovation Solution will serve as an input to another Innovation Solution as this is very clearly presented in chapter **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** with the description of all EVELIXIA Services. At the same time, some tasks may need to use solely (or not) raw data. Therefore, a general perception from the Application Layer universe could be the incorporation of “two ways of communication”.

The other key building block / module is the Services Broker which is devoted firstly to the direct communication between tasks / sub-tasks in the application services layer and secondly to the communication with the data and context broker in the Middleware layer.

The Middleware Platform (WP3) serves as the interconnection enabler between the application layer and the field layer and where processed and refined data is stored at different levels (office/room, zone, floor, building).

4.8.1.1 Building Awareness and Forecasting Toolbox

This toolbox presents a suite of tools designed for advanced building situation assessment and forecasting. These tools are developed on a sophisticated building energy Virtual Environment (simulation engine) (IS5). This simulation engine is meticulously calibrated using both real-time (sensory) and static (BIM) building data to create specific instances tailored to pilot projects. The engine's simulation and forecasting capabilities are augmented by both black-box (data-driven) and white-box (analytical) models, which enable the forecasting and evaluation of additional building features through dedicated components, such as:

A. Indoor Air Quality Measurement (IAQ) (IS1): This component assesses indoor air quality parameters, such as CO2 levels, volatile organic compounds (VOCs), humidity, and temperature, based on occupancy, HVAC operations, and ambient conditions.
B. Energy Assets Maintenance (IS2): This component calculates the solicitation, operational performance, and health diagnostics of microgrid-connected energy-intensive systems and appliances, including HVAC systems and energy storage units.
C. Demand Forecasting (IS3): This component provides forecasts for various building system vectors, including electrical, heating, and gas load/generation, at different scales (building and district levels).
D. Flex Forecasting (IS4): This component proactively evaluates demand flexibility levels for thermal and electrical energy.

The resulting product is a multi-dimensional digital Building Awareness toolbox integrated with an advanced web GUI. This toolbox can evaluate multiple building vectors and dimensions, offering a virtual testbed for validating various control scenarios.

4.8.1.2 Autonomous Building Decision Support Toolbox

This toolbox outlines a suite of decision-making and support components designed for building-level stakeholders, including occupants, building managers, and owners. These components aim to create a beneficial synergy between buildings and the grid by leveraging the simulation capabilities of the Building Digital Twin, developed through the "Building Awareness and Forecasting Toolbox." They will employ model-assisted schemes for microgrid planning, management, and control, utilizing advanced techniques such as Monte Carlo exhaustive reinforcement learning (RL), soft actor-critic RL, multi-timescale model predictive control (MPC), and ensemble isolation trees.

Specifically, the application focuses on the development of:

A. Building Investment Planning Assistant (IS6): This component evaluates energy and environmental investment impacts in terms of economic benefits (direct investment profits from energy trading) and economic viability, utilizing the VERIFY platform and the SRI advisor commercial platforms (IS7).

B. SRI Advisor (IS7): This component delivers tailored recommendations to enhance a building's Smart Readiness Indicator (SRI) score. It evaluates both overall performance and individual key functionalities, offering insights into the seven impact categories, such as energy efficiency and comfort. By performing sensitivity analyses on various technology renovation packages and flexibility scenarios, it identifies the most cost-effective upgrades to improve SRI score and performance.

C. DSM Services Advisor (IS8): This component serves as an Advisor for Building Operators, evaluating building capabilities, including consumption and generation patterns, flexibility resources, and storage. It assesses market needs and offers, such as System Operators' requirements for various services, and provides a prioritized strategy for delivering grid services. By leveraging this tool, Building Operators can effectively participate in demand response (DR) and demand-side management (DSM) schemes, maximizing revenue opportunities and reducing energy costs.

D. Proactive Demand Planning (IS9): This component is responsible for reshaping day-ahead demand using novel episodic RL methods and cost-benefit matrices, enabling energy cost savings without compromising energy efficiency at the building level.

E. Continuous Energy Performance Manager (IS10): This component optimizes the control of building systems and operations to enhance user comfort, convenience, and energy efficiency, employing various implementation strategies.

The outcome will be the infusion of autonomous self-decision-making capabilities into the multi-dimensional Building Digital Twin, transforming it into an Autonomous Building Digital Twin.

4.8.1.3 Network Awareness and Forecasting Toolbox

This toolbox details a suite of tools designed for enhanced multi-vector grids situation assessment and forecasting. These tools are based on an advanced network energy simulation engine known as the Intelligent Virtual Network (iVN) (IS15). The iVN engine is a sophisticated district modelling tool capable of performing simulations of city and community-level energy distribution networks using energy conservation and power-flow analysis across multiple energy vectors.

The iVN engine will be precisely calibrated with reduced datasets, real-time and historic sensory data, and static grid data (topology) sourced from the EVELIXIA middleware layer. This calibration will generate instances for the different energy vectors relevant to the various EVELIXIA pilot cases.

Importantly, these instances will incorporate Building Digital Twins as active elements within the modelled multi-vector energy grid. The primary outcome of this task is the creation of a grid-equivalent District Digital Twin, which will serve as a virtual testbed for validating various control scenarios.

As part of this toolbox a Smart Grid Maintenance (IS14) component will be developed. This component will support smart maintenance outage planning for multi-grid related assets (e.g., BESS, MV/LV substations, PVs, HPs, power lines) using aging and health assessment models when connected to electrical networks. It addresses the current time-based outage planning and asset condition monitoring approach, enhancing the detection of extreme deterioration and preventing catastrophic failures.

4.8.1.4 Autonomous District Decision Support Toolbox

This toolbox outlines a set of decision-making and support components focused on the district/grid level. These components target grid stakeholders, including energy aggregators, retailers, agencies, and grid operators, providing mutual benefits from Grid to Building by leveraging the simulation capabilities of the District Digital Twin. The decision mechanisms will employ model-assisted schemes for grid/district planning, management, and control, utilizing techniques such as episodic reinforcement learning with DDPG, optimized fuel assets management with XGBoost, and batched assets management through reward matrices and Q-learning.

Specifically, the development focuses on:

A. Grid Investment Planning Assistant (IS11): This component evaluates current and future network bottlenecks and assesses the necessity of potential infrastructural enhancements. It focuses on the economic benefits (direct investment profits from network enhancements) and the economic viability of such investments.

B. Multi-Vector Energy Network Manager (IS12): This component is responsible for enabling network reserves and managing congestion, addressing various cross-vector scenarios.

C. Aggregated Demand Portfolio Manager (IS13): This component handles real-time daily portfolio replanning and rescheduling. It adjusts the expected demand amounts at aggregated building and district scales as soon as the observed state deviates from the expected one. This flexibility enables proactive resource aggregation and active participation in energy balancing markets, primarily focusing on the electricity vector.

The outcome of this task is the integration of autonomous self-decision-making capabilities into the District Digital Twin, transforming it into an Autonomous District Digital Twin.

4.8.1.5 Stakeholders Interaction Platform

This module focuses on developing components that enhance easiness of use and encourage active participation of building and grid energy stakeholders in the configuration and operation of EVELIXIA, ensuring a comprehensive smartification experience for EVELIXIA's end-users.

Specifically, the development will focus on:

A. Digital Building Logbook (DBL) (IS16): This component is responsible for serving as a common data environment and repository for EVELIXIA use cases, enhancing informed decision-making by facilitating seamless information flow among stakeholders (e.g., building owners/users, operators, utilities, and energy service providers).

B. Visual Analytics Engine (VAE) (IS17): This component is responsible for providing a high-level overview of building operational states, as reflected by the BDT real-time data monitoring flexibility potential. It also offers useful insights and the network location of available flexible assets and their capacities through advanced visual analytics and input from the Building Energy Modelling and Simulation and Multi-Vector Grids Energy Modelling & Simulation components.

C. Energy Services Marketplace (IS18): This component is responsible for a Marketplace that enables data exchange and sharing of BDT and DDT services through semantically enabled subscription schemes. It includes a wiki for service descriptions and endpoints for external agents (users, apps, partners, etc.) to upload data and information. The EVELIXIA marketplace will:

   - Enable Exchange and Sharing: Facilitate the exchange and sharing of Building Digital Twin (BDT) and District Digital Twin (DDT) services.
   - Support Subscription Schemes: Offer semantically enabled subscription and unsubscription schemes for the activation and deactivation of services.
   - Provide Data Upload Endpoints: Supply endpoints for external agents to upload data and information.
   - Explore Metadata: Allow users to explore available metadata in a wiki-style format, including service descriptions, datasets, machine learning pretrained models, and results.

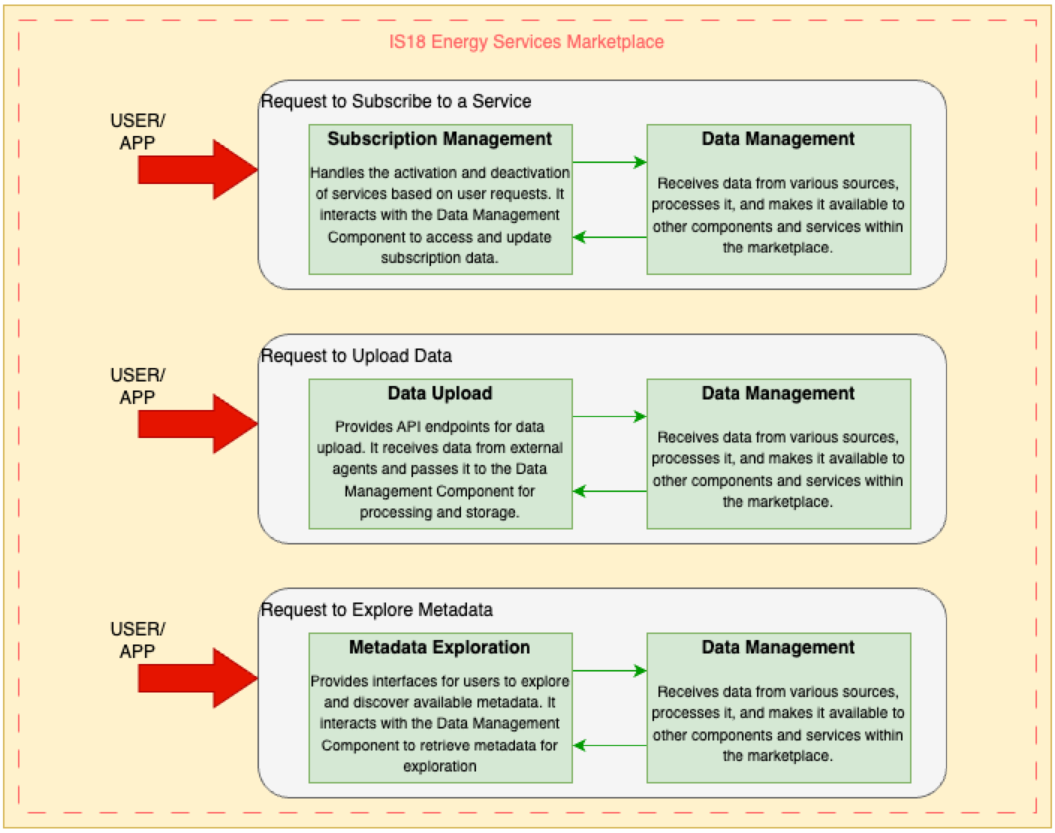


Figure 9 – Energy Services Marketplace Flow

The basic data flow of the EVELIXIA Energy Services Marketplace includes:

- **Data Upload:** External agents upload data and information to the marketplace via the provided endpoints.
- **Service Activation:** Users activate services using the semantically enabled subscription schemes.
- **Data Exchange and Sharing:** BDT and DDT services are exchanged and shared within the marketplace.
- **Service Use:** Users utilize the activated services and explore the available metadata.
- **Service Deactivation:** Users deactivate services through the semantically enabled unsubscription schemes.

D. Building Virtual Model (IS19): This component is responsible for creating and managing the interactive 360° virtual photogrammetric model of a plant, site, or building. It enables virtual navigation through assets, integrates safety procedures and training modules, displays real-time data, maintains an inspection data repository, plans advanced maintenance activities, and supports interactive features for commercial purposes.

E. Blockchain-based Infrastructure (IS20): This component is responsible for securely managing Building-to-Grid and Grid-to-Building services, enabling EVELIXIA stakeholders to engage in the energy services marketplace while ensuring privacy and trust. It supports origin verification of energy sources and available energy flexibility and utilizes an Identity and Authorization system for user and device verification, a Governance Framework for value creation and currency conversion, a Tokenization Service for secure payment transactions, and an Origin Verification Service for energy source validation and capacity assessment.

4.8.1.6 B2G and G2B Services Broker

This module presents a suite of decision-making and support services aimed at the district/grid level. These services are designed for grid stakeholders, such as energy aggregators, retailers, agencies, and grid operators. They aim to provide mutual benefits from Grid to Building by leveraging the simulation capabilities of the District Digital Twin. The decision mechanisms will utilize model-assisted schemes for grid/district planning, management, and control, including episodic reinforcement learning with DDPG, optimized fuel assets management using XGBoost, and batched assets management through reward matrices and Q-learning.

The specific developments include:

A. Grid Investment Planning Assistant (IS11): This component will evaluate current and future network bottlenecks and assess the necessity of potential infrastructural enhancements. It will focus on the economic benefits (direct investment profits from network enhancements) and the economic viability of such investments.

B. Multi-Vector Energy Network Manager (IS12): This component will enable network reserves and manage congestion, addressing various cross-vector scenarios.

C. Aggregated Demand Portfolio Manager (IS13): This component will handle real-time daily portfolio replanning and rescheduling. It will adjust expected demand amounts at aggregated building and district scales as soon as the observed state deviates from the expected one. This flexibility will enable proactive resource aggregation and active participation in energy balancing markets, primarily focusing on the electricity vector.

The result of this task will be the integration of autonomous self-decision-making capabilities into the District Digital Twin, transforming it into an Autonomous District Digital Twin.

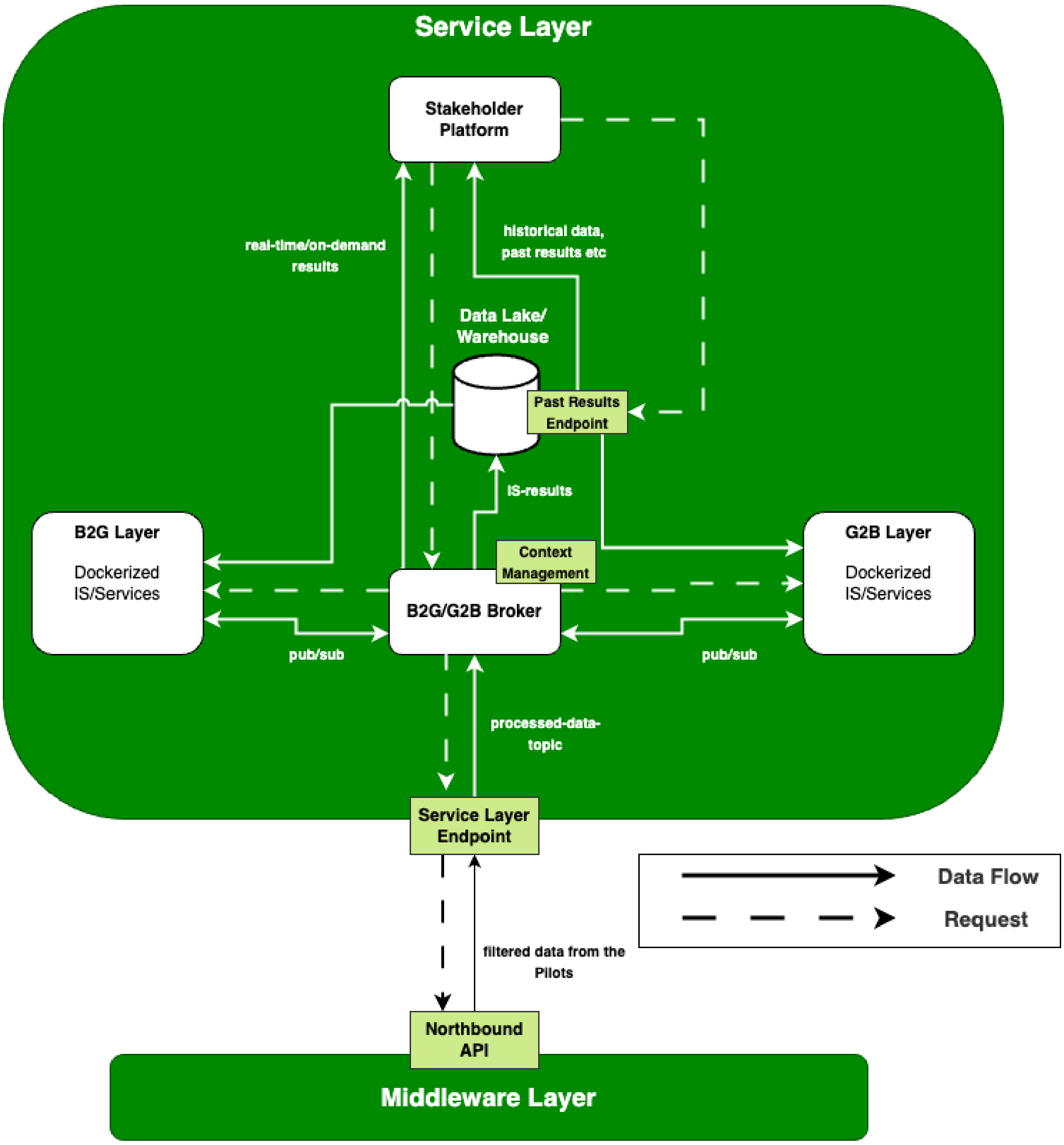


Figure 10 – Services Broker High Level Architecture

4.8.2 Data Management Layer

This is the interoperability layer, corresponding to the Interoperability and Abstraction Services developed in EVELIXIA. The common data modelling framework will be delivered along with relevant data ingestion and brokering services for data harmonization, filtering, and event management from various sources. The Blockchain Infrastructure will be part of this layer, as the main enabler of EVELIXIA Services value extraction and marketability, secure authentication, distributed information logging and verification.

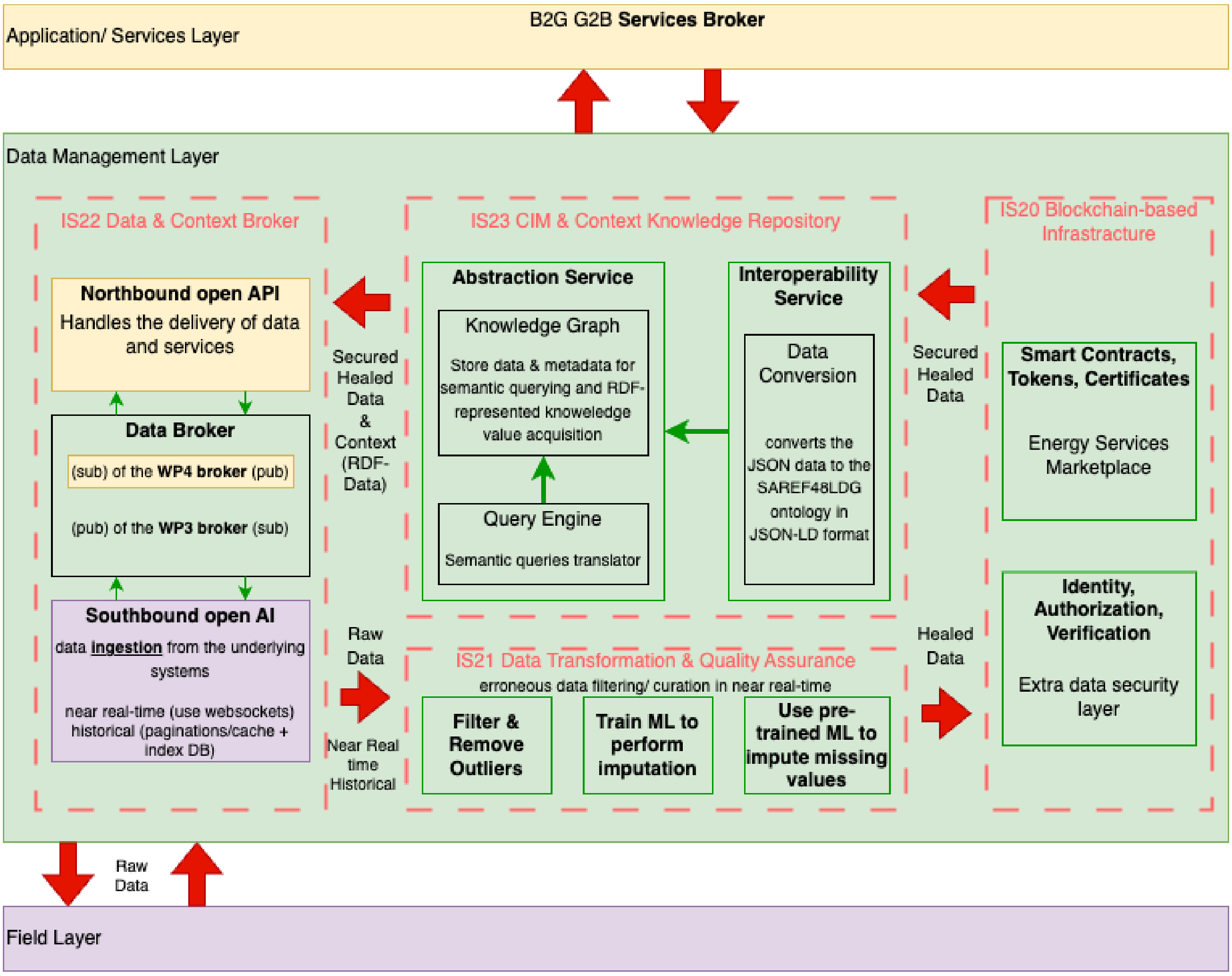


Figure 11 – Data Management High Level Architecture

4.8.2.1 Data and Context Broker

The term "Middleware Broker" is not a component, but rather a term used to describe the overall system or layer that facilitates communication, data management, and service provision between the other two layers in the EVELIXIA architecture.

So, the Middleware Broker does not have a specific location in the sequence of steps, but its functionalities are carried out through the various components of the "IS22 Data and Context Broker" solution. It is the overarching system that enables the flow of data and services from one layer (field layer) to another (application/services layer).

We envision that the middleware broker must focus on its core functionalities, such as managing data and context, and enabling seamless semantical queries to extract RDF-represented knowledge value. The separate component would handle the pre-processing of the data, ensuring that the data is in the right format, is aggregated correctly, and is free of errors before it is sent to the broker.

This division of responsibilities could make the overall system more efficient and reliable. The middleware broker could focus on providing high-level services based on the data, while the separate component ensures the quality and consistency of the data that the broker works with. This could also make the system more modular and easier to maintain, as each component has a clear and distinct role.

The two functionalities, "***managing data and context***" and "***enabling seamless semantical queries to extract RDF-represented knowledge value***", are core to the middleware broker because they directly address the main challenges in handling data in a complex system like a smart grid-efficient interactive building.

1. **Managing Data and Context**: In a system with multiple data sources and types, managing data involves ensuring that data from various sources is properly received, stored, and made available for further processing or querying. Context management involves understanding the relationships between different data entities and providing meaningful insights. This is crucial in a smart grid scenario where data from various sources (like energy consumption, occupancy, weather conditions, etc.) needs to be understood in relation to each other.
2. **Enabling Seamless Semantical Queries to Extract RDF-Represented Knowledge Value**: RDF (Resource Description Framework) is a standard

model for data interchange on the Web. It allows for the representation of information in a way that is both human-readable and machine-readable. Enabling seamless semantical queries means allowing users to easily query this RDF-represented data in a way that is intuitive and meaningful to them. This is important in a smart grid scenario where different stakeholders (like energy providers, building managers, residents, etc.) may need to query the data in different ways to extract the information that is relevant to them.

The overarching use case that these functionalities serve could be "Real-Time Energy Management". In this use case, the middleware broker would receive raw data from various sensors in the field layer (like energy consumption data, occupancy data, weather data, etc.). This data might be in various formats and might come in at different rates.

The middleware broker, located in the data management layer, would manage this data and its context, transforming it into a standard format, aggregating it as necessary, and associating it with the relevant metadata. It would then enable seamless semantical queries on this data, allowing different stakeholders to extract the RDF-represented knowledge value that is relevant to them.

4.8.2.1.1 Northbound Open API

This component handles the delivery of data and services from the middleware broker to the applications or services. It is directly related to the "Service Abstraction" functionality and resolves the "Abstracted Service Provision" use case. Key functions of the module are:

- Handles the delivery of data and services to the applications/service layer.
- It is a connector that provides a communication interface between the data broker and the applications/service layer. It makes calls to the endpoints of an existing API. This connector can authenticate with the API, make requests to the API's endpoints, and handle the responses it receives. The connector acts as a client of the WP4 Applications/Services Layer API.
- Receives processed data from the data broker.
- Sends data to the applications/service layer.
- Input Processed data (JSON-LD, variable, data broker).
- Output Data (JSON-LD, variable, applications/service layer).

- The API will be executed from a Cloud-based server that can optimally be installed in any environment (MS-Azure, AWS, Private)
- For optimal performance, the API requires a server with sufficient storage and processing capabilities that will be defined in a later stage and upon tests and trials.

4.8.2.1.2 Data Broker

Manages the flow of data between the northbound and southbound APIs. It is a component that receives data from the southbound API, processes it, and sends it to the northbound API. Concluding the data broker:

- Manages the flow of data between the northbound and southbound APIs.
- It is a component that receives data from the southbound API, processes it, and sends it to the northbound API.
- Receives data from the southbound API.
- Sends data to the northbound API.
- Input data is from field devices (JSON raw, variable, southbound API).
- Output Parameters comprise processed data (JSON-LD, variable, northbound API).
- The Software execution environment can be any accredited data center with cloud-based server (MS Azure, AWS, private)
- The API will be executed from a Cloud-based server that can optimally be installed in any environment (MS-Azure, AWS, Private)
- For optimal performance, the API requires a server with sufficient storage and processing capabilities that will be defined in a later stage and upon tests and trials.

4.8.2.1.3 Southbound Open API

This component handles the ingestion of data from the underlying systems or databases into the middleware broker. It is crucial for the "Data Management" functionality and resolves the "Real-Time Data Management" use case. The southbound API:

- Handles data ingestion from the underlying systems.
- It is a connector that provides a communication interface between the field layer and the data broker.
- It makes calls to the endpoints of an existing API.
- This connector can authenticate with the API, make requests to the API's endpoints, and handle the responses it receives.

- The connector acts as a client of the WP2 API which exposes the data collected from pilot sites based on building and multi-vector grid monitoring platforms and BEMS.
- Receives raw data from the field layer.
- Sends data to the data broker.
- Input parameters in terms of raw data are from field devices (JSON raw, variable, field layer).
- Output data (JSON raw, variable, data broker).
- The Software execution environment can be any accredited data center with cloud-based server (MS Azure, AWS, private)
- The API will be executed from a Cloud-based server that can optimally be installed in any environment (MS-Azure, AWS, Private)
- For optimal performance, the API requires a server with sufficient storage and processing capabilities that will be defined in a later stage and upon tests and trials.

4.8.2.2 CIM & Context Knowledge Repository

Key function is the conversion of the data to the SAREF ontology-based JSON-LD format and storing it in a knowledge graph. This is an Interoperability Service (Data Conversion from JSON to SAREF Ontology Based JSON-LD) and summing up:

- This component is responsible for converting the data to the SAREF ontology-based JSON-LD format.
- It uses mapping rules to convert JSON data to SAREF ontology-based JSON-LD.
- Receives secured data from the Blockchain-based Infrastructure component.
- Sends ontology-based data to the Abstraction Service component.
- Input parameters are secured data (JSON-LD, variable, Blockchain-based Infrastructure component).
- Output parameters comprise an Ontology-based data (JSON-LD, variable, Abstraction Service component).
- The Software execution environment can be any accredited data centre with cloud-based server (MS Azure, AWS, private)
- For optimal performance, a server with sufficient storage and processing capabilities is proposed which will be defined in a later stage and upon tests and trials.

4.8.2.2.1 Abstraction Service

Brief Conceptual Description:

- This component is responsible for storing the ontology-based data in a knowledge graph and providing a query engine.
- It uses a graph database to store the ontology-based data and provides a SPARQL endpoint for querying the data.
- Receives ontology-based data from the Interoperability Service component.
- Sends data to the Data Broker.
- Input parameters are Ontology-based data (JSON-LD, variable, Interoperability Service component).
- Output parameters data are: (JSON-LD, variable, Data Broker).
- The Software execution environment can be any accredited data canter with cloud-based server (MS Azure, AWS, private)
- For optimal performance, a server with sufficient storage and processing capabilities is proposed which will be defined in a later stage and upon tests and trials.

4.8.2.2.2 Interoperability Services

The Interoperability Service (Data Conversion from JSON to SAREF Ontology Based JSON-LD):

- Is a component that is responsible for converting the data to the SAREF ontology-based JSON-LD format.
- It uses mapping rules to convert JSON data to SAREF ontology-based JSON-LD.
- Receives secured data from the Blockchain-based Infrastructure component.
- Sends ontology-based data to the Abstraction Service component.
- Input parameters consist of secured data (JSON-LD, variable, Blockchain-based Infrastructure component).
- Output parameters are ontology-based data (JSON-LD, variable, Abstraction Service component).
- The Software execution environment can be any accredited data centre with cloud-based server (MS Azure, AWS, private)
- For optimal performance, a server with sufficient storage and processing capabilities is proposed which will be defined in a later stage and upon tests and trials.

4.8.2.3 Data transformation and quality assurance

The primary function of this component is to ensure data quality by filtering and removing outliers, and imputing missing values. It is important to note that data transformation will not include aggregation.

4.8.2.3.1 Filter and Remove Outliers

Brief Component Description:

- Purpose: This component is responsible for filtering and removing outliers from raw data.
- Technical Overview: Utilizes statistical methods to identify and remove outliers.
- Inputs:
  - Raw data from field devices (JSON format) via the Southbound Open API connector.
- Outputs:
  - Filtered data (JSON format) to the Machine Learning component.
- Software Requirements:
  - Development Language: To be determined.
  - Execution Environment: Cloud-based server.
- Hardware Requirements:
  - Server with adequate storage and processing capabilities

4.8.2.3.2 Train Machine Learning to Perform Imputation

Brief Component Description:

- Purpose: This component is responsible for training a machine learning model to impute missing values.
- Technical Overview: Employs machine learning algorithms to learn data patterns and impute missing values.
- Inputs:
  - Filtered data (JSON format) from the Filter and Remove Outliers component.
- Outputs:
  - Trained model (binary format) to the Use Pre-Trained Machine Learning to Impute Missing Values component.
- Software Requirements:
  - Development Language: To be determined.
  - Execution Environment: Cloud-based server.
- Hardware Requirements:
  - Server with adequate storage and processing capabilities.
  - GPU for machine learning tasks.

4.8.2.3.3 Use Pre-Trained Machine Learning to Impute Missing Values

Brief Component Description:

- Purpose: This component utilizes a pre-trained machine learning model to impute missing values in new data.
- Technical Overview: Applies the trained model from the Train Machine Learning to Perform Imputation component to impute missing values.
- Inputs:
  - New data (JSON format) from the Filter and Remove Outliers component.
  - Trained model (binary format) from the Train Machine Learning to Perform Imputation component.
- Outputs:
  - Imputed data (JSON-LD format) to the Data Broker.
- Software Requirements:
  - Development Language: To be determined.
  - Execution Environment: Cloud-based server.
- Hardware Requirements:
  - Server with adequate storage and processing capabilities.
  - GPU for machine learning tasks.

4.8.2.4 Blockchain-based infrastructure

The Blockchain and associate smart contract framework will provide secure utilization of Building-to-Grid and Grid-to-Building Services allowing EVELIXIA stakeholders to participate in the energy services marketplace in a trusted and privacy preserving manner. Moreover, it will act as an origin verification network where users can securely verify the source of energy as well as the energy flexibility/energy excess capacity available per user and asset for potential use in energy services. The infrastructure to be utilised will include:

a) an Identity and Authorization component that allows users and devices to prove their identities and acquire access permissions based on their roles or attributes,

b) a Governance Framework component for handling the creation of value and its potential conversion to real (fiat) currency,

c) a Tokenization Service for secure payment of services utilisation i.e. the transfer of value between bank accounts, and

d) an Origin Verification Service for verifying the source of energy and the energy flexibility/energy excess capacity available per user and asset for potential use in energy services.

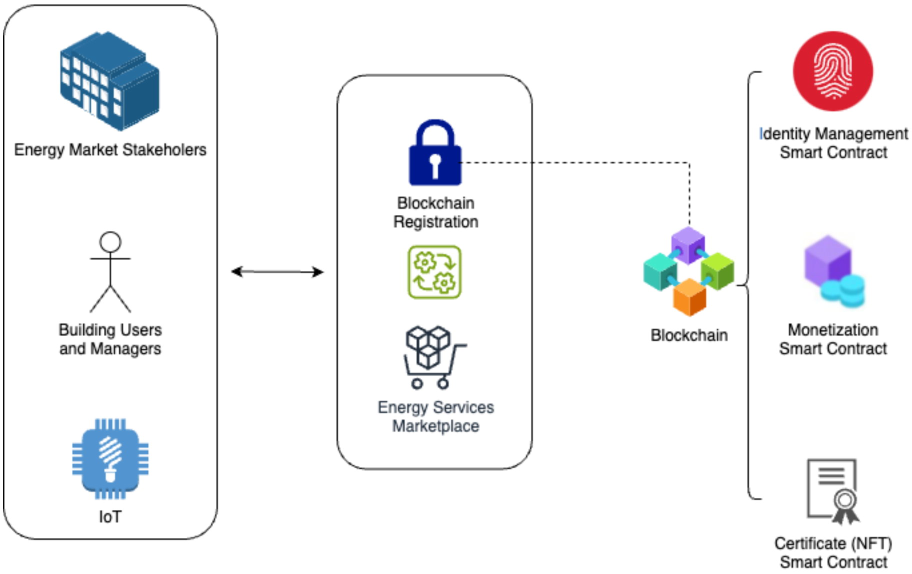


Figure 12 - Blockchain based Infrastructure

Moreover, the blockchain-based infrastructure will ensure the verifiability of the energy origin by leveraging on cutting-edge technologies such as Non-Fungible Tokens (NFTs) as well as the verifiability of energy flexibility / energy excess capacity available per user and asset for potential use in energy services using blockchain deployed smart contracts. A smart contract-based governance and monetization framework that will leverage and extend already existing solutions and knowledge will secure and automate the transactions among stakeholders that use services and exchange resources.

For the Blockchain framework Distributed Ledger technologies will be used that have proven their scientific and operational reliability extensively. The distributed ledger (blockchain-type) technologies translate conventional agreements into smart contracts for automated transactions. Smart contracts will be used to enforce the agreements between buildings and interconnected grids, both at the building and district level. Smart contracts will be based on Ethereum and will be used in Solidity to facilitate transactions among multiple parties. The whole module - app will be developed based on Distributed Ledger Technologies (DLT) and more specifically in an alternative approach called Ricardian Contracts. As a result, Transaction Manager will advance the integration between Smart Contracts and Ricardian contracts. It will also encode, and compile rules written in the Ricardian-type, human readable language into a functional, machine-readable language that will be translatable to smart contracts in the next levels of the pipeline using appropriate compiler techniques and communication APIs.

The dedicated security and transparency BLOCKCHAIN mechanisms will be developed to allow energy smart contracting schemes between buildings and interconnected grids, both at the building and district level.

4.8.3 Field Layer

All delivered novel SW tools (and relevant data processing) are included in this layer, building up the EVELIXIA Energy Services for Building and Energy Network(s) Stakeholders, encompassed in the developed Autonomous Building and District Digital Twins (ABDT, ADTT).

The vast number of services, utilising big data from the same or dissimilar sources, being activated in different timeframes and for different time intervals (e.g. day-ahead forecasting vs real-time power re-dispatch), requires an expansive array of data connectors and sophisticated coordination between them; to address such complexity, while reducing the development and total computational burden, EVELIXIA develops the B2G & G2B Services Broker, which acts as the mediator and manager of messages distribution between different services across the EVELIXIA Platform.

5 DEVELOPMENT VIEW

For the development view, we describe how we will support the development process. To this end, we discuss how we handle the revision control of source code files in a centralised repository and aspects of the development process to enhance the quality of the software.

5.1 Dashboard Development Approaches

This is the main EVELIXIA dashboard that each user will access and experience when logging in to the integrated platform. This dashboard will be the unique interaction point between the end user and the platform and within this dashboard all user related information will be presented.

5.1.1 Server-Centric Web Application

A Server-Centric Web Application (SCWA) uses a Server to collect data, process them and serve the resulting HTML page to the client's browser. The disadvantage of this approach is that the page is posted back to the server; this introduces communication and processing overhead that can decrease the overall performance, as it forces the user to wait for the page to be processed and recreated.

5.1.2 Browser-Centric Web Application

The Browser-Centric Web Application (BCWA) approach embeds all the functional parts (e.g. Scripts) on the client's side and executes them on the client's Internet browser. The advantage of BCWA architecture is the fast response times and minimum overhead (e.g. Data, Processing Power) on the web server. Additionally, BSWA is ideal when the page elements need to be changed without the need to interact with the database.
A very popular technology belonging to the BCWA category, is Single Page Applications (SPA). Essentially, an SPA is a web application that fits in a single page and loads on the initial page request. SPA is suitable for web-centric applications that handle a large amount of data by exploiting asynchronous downloading of features (HTML templates/JSON data), and by re-rendering locally any part of the interface without requiring from the server to re-render the complete HTML page. Finally, SPA allows web developers to give a "native application" experience to the end-user.

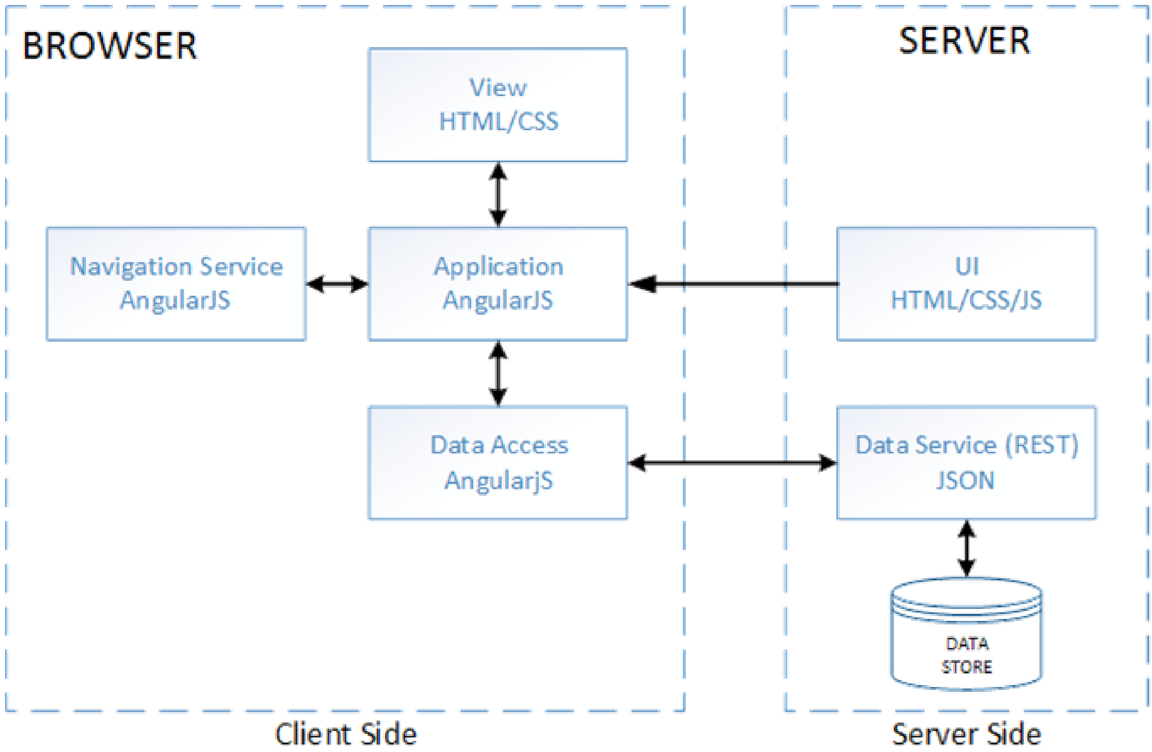


Figure 13 - Single Page Application

5.1.3 Summary

For the Dashboard's implementation, the Browser-Centric Web Application approach was chosen for the following two reasons: scalability and faster response time (excluding the initial load time). For the former, BCWA is more suitable as it diffuses the processing load from the server to the clients; in other words, all the code is downloaded from the server, at the session's beginning and then it runs inside the clients' browsers. For the latter, as all (or almost all) of the data are inside the browser, the time for getting, processing and displaying them, is usually far less than the time needed to request and get them from the server (SCWA paradigm).

5.2 Cloud Components Development

To put the Cloud Platform in the context, we need to define the usual DevOps techniques in the development and deployment of cloud applications. This is relevant for EVELIXIA since the whole platform will be basically a standard cloud application, running on a designated cloud provider or cloud-based systems offered from partners and associate resources and being orchestrated by the administrators of the solution (in our context, this is the EVELIXIA administrator(s)). The standard workflow in the development of a cloud application consists of:

- Development process (developing solutions)
- Continuous integration process (making deployment stable and tested)
- Continuous deployment process (where orchestration is the most important)

During the development process we ensure that we develop and provide all the artefacts needed to ensure stable deployment of the application. To ensure that stability and integration between delivered artefacts is complete, we will put in place Continuous Integration and Deployment (CI/CD) processes in the cloud. To do so, we need to make use of several DevOps tools like:

- ✓ Gitlab (or similar): For code versioning/repository management.
- ✓ Jenkins (or similar): For CI/CD and deployment automation.
- ✓ Docker Compose: For orchestrating multi-container Docker applications.

These artefacts are later used for the purposes of Continuous Integration and Continuous Deployment (again, provided by the DevOps tools such as Jenkins and Ansible).
Developing software relies on teamwork and collaboration. It is essential that developers can distribute, merge, integrate and debug their work from early stages of development. In development we can define distinctive steps in the process, starting from coding and ending with deployment of the developed code. The list of these steps encompasses:

1. Code committing / versioning
2. Static code analysis / code inspection
3. Unit testing
4. Artifact generation
5. Deployment to testing environments
6. Integration testing
7. Business logic testing
8. UI testing
9. Application acceptance testing
10. Deployment to production environments
11. Post-deployment testing

Whenever there is an error in any of these steps, the whole process is usually aborted, and reports are sent to the developers who are in charge, until the issues are resolved. Not all steps are always necessary; it depends on the parameters one wants to observe. Most of these steps can be fully automated, but that depends on the individual use-cases and if there are special circumstances for any project or artefact. In the following paragraphs, we provide a short description for each development process step.

Code committing / versioning

The basic principle is that all the resources needed for creating artefacts should be available through one or more code versioning systems. This way one has full access to all the resources, their history and additional information regarding open issues, or code review comments.

Continuous integration

Server can scan the code repository for changes regularly and triggers the cycle.

Static code analysis / code inspection

The code is fully compiled with necessary pre-processing (schema compilation, preparing the external resources, etc.); therefore, every cycle starts with a fresh environment and resources. The code is assessed and searched for potential bottlenecks, locks or unused code.

Unit testing

Unit testing usually covers the basic, smallest business logic blocks. It ensures that any changes introduced in the code base does not change the basic functionality and system logic.

Artefact generation

Artefacts can be the result of code compilation and/or preparing deliverable packages. Usually once an artefact is created, it is stored in a repository, tagged, and a version is assigned to it, to reference and obtain it at any given time. Such practice allows for repeated installation with different artefact versions - basically a full deploy undo system in terms of artefacts.

5.2.1 Continuous Integration process

The process of continuous integration (CI) includes three steps:

- ➢ code committing,
- ➢ code inspection and
- ➢ unit testing.

The code can be merged from different sources, compiled, checked for anti-patterns, perform a style check if coding style guidelines are in practice.

CI always works on top of a code versioning system, providing the integrity of the codebase. **In EVELIXIA it has been decided to leave the source code management to each individual partner, orchestrating ONLY the docker images and having a centralized system only for storing important results/data.** Most important part of the CI process is the unit testing, which should cover all the basic functionality of each component, to ensure that the changes introduced do not compromise the functionality of the whole integrated system. Each component developed in EVELIXIA should have a clear documentation on testing. The CI cycle can be automatically triggered in regular time intervals or on code-base changes. Reports are sent to persons who monitor the project or in case of errors, to the developers that introduced them. Figure 14 provides an overview of the CI process and the environment.

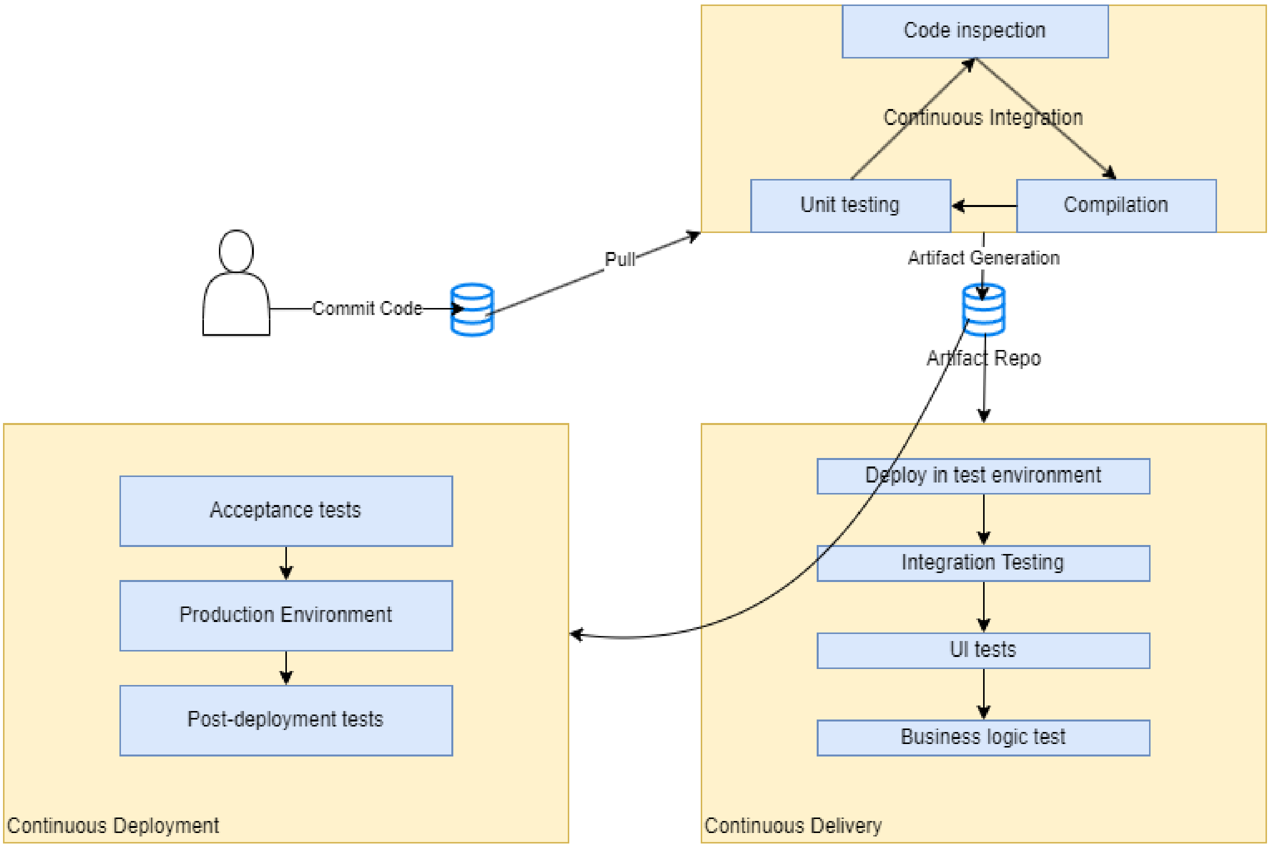


Figure 14 - Standard Continuous Integration environment

5.2.2 Continuous deployment process

Continuous Deployment process is needed to obtain more control over:

- ➢ Quality of committed code
- ➢ Artefact / deliverable generation
- ➢ Deployment process

We can implement different CI processes, depending on the level of automation required. Tools envisioned within EVELIXIA development will be:

Gitlab – Code versioning system, Gitlab is an open-source tool providing basic functionalities for code revision, team collaboration, ticketing, wiki documentation, and with the latest updates it also provides basic continuous integration process.

Jenkins – Continuous integration server, Jenkins is an open-source tool and can be equipped with a set of extensions of 3rd party tools through plugins.

We plan to use the tool to deploy artefacts of components on top of chosen infrastructure for testing and production purposes.

5.3 Coding Conventions

The coding conventions that have been decided so far are the following:

- Information between different modules will be exchanged using JSON objects
- GUID's will be used as object identifiers
- Date/Time fields will be using ISO 8601 format (YYYY-MM-DDYHH-MM-SSTZ) or UNIX timestamps.

More coding conventions will be added as the development continues, and the complete list will be included in the second and final version of the system architecture document (D1.4[b]).

5.4 Use of existing frameworks and tools

This paragraph presents the existing software and hardware development tools that will be used towards implementing the EVELIXIA s solution. This includes the libraries, Application Programming Interfaces (API's) and Software Development Kits (SDK's) that are part of existing frameworks and hence required to develop the project's solution.

- Apache Kafka: A distributed event store and stream-processing platform used to implement the core publish/subscribe mechanism of the Service Broker. It is an open-source system developed by the Apache Software Foundation written in Java and Scala. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. It includes key components such as Zookeeper for managing the Kafka Cluster.
- Kafka Connect: Kafka Connect is an open-source component of Apache Kafka that serves as a centralized data hub for simple data integration between databases, key-value stores, search indexes, and file systems. It will be used to enable seamless data flow between the Service and the Data Broker.
- Docker: A containerization platform used to package EVELIXIA's tools and microservices into portable containers. Docker Compose is utilized to orchestrate the multi-container application, simplifying deployment and management.

6 DEPLOYMENT VIEW

Deployment view describes how and where the system will be deployed, which physical requirements are essential for the system to go live. Another important factor is the dependencies of the various components, modules, and sub-modules, their hardware requirements, and their physical constraints. In this view the physical environment where the system will run is defined, including:

- Hardware and Technical requirements.
- Mapping of software elements to the runtime environment.
- Third-party software requirements.
- Network requirements.

It is noted here that all the microservices that compose the EVELIXIA holistic system will run in a cloud environment over isolated containers managed by Docker. A docker compose file will be defined with the purpose of providing an easy way to deploy all to the ecosystem. On the other hand, at the cloud side, several virtual operating systems will be deployed, to accommodate the various needs of the architecture. The virtual OS's will be Linux based.

6.1 Deployment Architectures

This section presents the selected and alternative deployment approaches for the EVELIXIA Platform. It concludes with a brief justification for the selection.

6.1.1 Monolithic

A monolithic architecture is defined in the traditional client-server model. In this architecture, all the functionalities are contained within a single service, which is the most common way to develop and package web applications. A monolithic web application is typically composed of the client, the server and the database. In this context, the Users' Dashboard along with the other modules are developed as a single application, exploiting one common web development framework, and are deployed in one virtual machine.

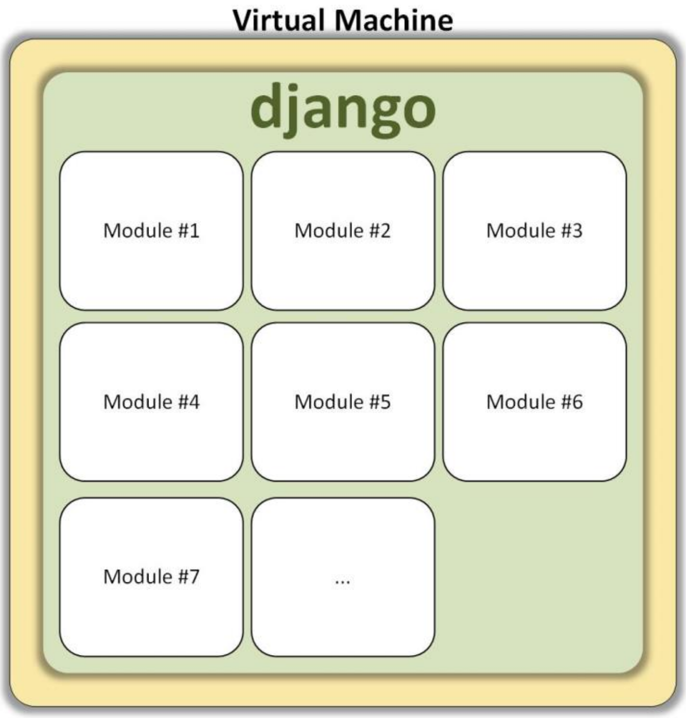


Figure 15 - Monolithic Application Example

6.1.2 Micro-Services

Micro-services architectural paradigm allows for the decomposition of heavy applications, dissected in several smaller components, named micro-services, which can be distributed across multiple locations. Each micro-service focuses on the execution of one or a small set of tasks, is independently deployable and usually utilises an HTTP API for communicating with the rest of the micro-services. In this context, each application module may be deployed as a distinct micro-service, exploiting a different development framework. Moreover, and for avoiding library versioning incompatibilities and conflicts, each micro-service can be deployed in a distinct virtual machine.

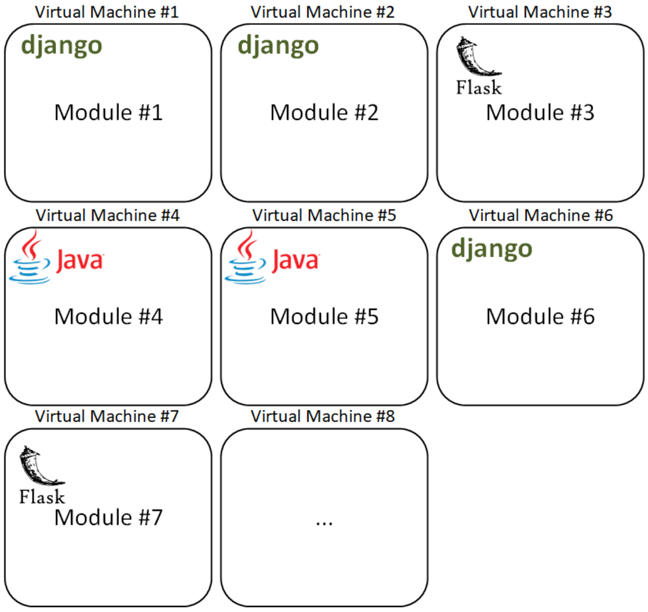


Figure 16 - Micro-Service Application Example

6.1.3 Containerized Micro-Services

Containerized micro-services refer to the deployment of each micro-service, inside a docker container. Docker is a container that packages all the software (e.g. code, libraries, third-party tools, etc.) needed to implement and run the micro-service. This allows the deployment of the micro-service, on any computer, on any infrastructure (e.g. cloud, fog, etc.). In this context, each EVELIXIA Marketplace Service (6 in total) can be deployed as a distinct micro-service, exploiting a different development framework. However, the use of dockers isolates modules from each other, allowing developers to use different libraries, in different versions, avoiding incompatibilities between modules (library versioning). Therefore, all modules can be deployed as dockers, in a single virtual machine (Figure 21).

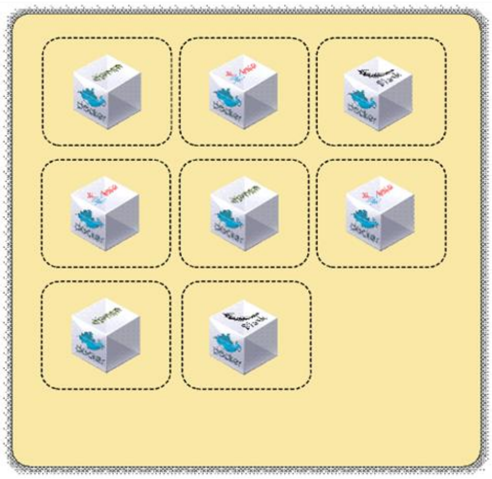


Figure 17 - Containerised Micro-Service Application Example

6.1.4 Conclusion

For the design and deployment of the EVELIXIA Platform, the containerized micro-service approach is selected. This approach will allow the agile development of the platform, by enabling the consortium developers to individually develop each module, using their tools/libraries /frameworks of preference, without having to abide by a single architecture/development directive. Furthermore, the utilisation of the Docker technology, enables the deployment of all modules, in a single virtual machine.

6.2 Cloud platform components deployment requirements

6.2.1 Third-party Software Requirements

The EVELIXIA system is designed as a stand-alone system, so no major third-party software requirements exist. To this end, the most common web browsers (i.e. Firefox and Chrome) can be used for accessing the offered Dashboards.

6.2.2 Network Requirements

A network connection is required between the EVELIXIA Platform and the cloud for interacting with the EVELIXIA Application / web Marketplace and the cloud system in general. The network capacity should be enough for supporting the fast deployment of all modules and for transferring, monitoring and controlling data back and forth between the EVELIXIA data management layer and the Application Layer / Marketplace. In the same context, the cloud should provide adequate network resources for allowing the developers to upload fast their solutions / applications.

7 SERVICES INFORMATION VIEW

Based on the foreseen technologically innovative solutions (IS) landscape, the EVELIXIA project aims to develop and provide six (6) highly elaborate innovative services designed to address energy efficiency challenges both at building and district level. To achieve this, the technical management of the project, together with other key technical partners, designed six (6) different collections/ecosystems of IS and workflows, one for each different service respectively.

One initial level of the six (6) services' categorizations considered the origin of the available observations and decisions/controls for providing the respective service. As a result, the proposed services can be divided in two (2) categories according to the level of equipment the type of request. The level of equipment can be determined by its location, either within the premises of a building infrastructure or at the level of the energy (distribution) grid.

Therefore, each category can be defined as follows:

- Behind (below) the Meter: Services that operate within the customer's premises, managing energy consumption, generation, and storage to optimize energy use, reduce costs, and enhance energy efficiency.
- Front-of (above) the Meter: Services that operate at the grid level, managing energy distribution, generation, and storage to maintain grid stability, ensure reliable power supply, and optimize overall grid performance.

The request for each service can be characterized by the frequency of the service's execution. The following two main categories are identified in this context:

- On (User's) Demand: They are activated in response to a request send by the end user through the Energy Services Marketplace in the Stakeholders Interaction Platform (see Task 4.5, IS18). Two (2), out of the six (6) specified services, related to planning of investments are enclosed in this category with the aim to provide long-term (e.g., 20-year) future projection results with monthly or annual granularity at building and grid level, respectively.
- Schedule Based: These services operate according to a pre-defined timetable, ensuring their continuous availability to operate at a regular basis, providing different types of management and decision-making services of energy assets/resources both at building and district level.

- Hourly (10-Minute Granularity): Manages energy assets/resources accordingly, to meet the potential goals for participating in the balancing market, applicable across dynamic (intra-day) balancing market pricing schemes.
- Daily (60-Minute Granularity): Manages energy assets/resources accordingly, to meet the potential goals for participating in the spot market, applicable in a day-ahead manner.

Below, the six (6) EVELIXIA services are presented, each illustrated by a corresponding UML diagram. These diagrams were developed as a result of a comprehensive analysis of the 23 different EVELIXIA Innovative Solutions (ISs), which are detailed in the Annex of this deliverable. The analysis aimed to capture the intricacies and interactions of these solutions to provide a clear and precise representation of how each service operates.

The sequence diagrams are the result of the analysis of all the EVELIXIA ISs' inputs and outputs. An IS running in a later stage than another IS does not mean that it requires its input by default. It may as well require an input from a solution that relies on other downstream ISs. The list of all ISs inputs and outputs as well as a more detailed description of their functionality can be found in the first Annex of this deliverable.

The following sections offer an in-depth analysis of the dynamic aspects of the EVELIXIA services. The descriptions and sequence diagrams illustrate the interdependencies between the Information Systems (ISs). The precise execution flow of each service will be finalized in the next version of this deliverable.

7.1 SERVICE 1: DSM IMPLICIT (price-based shifting)

EVELIXIA's "DSM IMPLICIT (price-based shifting)" service enables building managers and owners to adjust demand fractions by reducing or shifting energy-intensive loads from expensive periods to cheaper ones, optimizing energy use and costs within the same energy vector or across different sectors (sector coupling). This approach contributes directly to reducing energy costs, while implicitly enhancing grid stability, and promoting sustainability by efficiently managing energy consumption and reducing peak demand. The service is characterized as behind the meter, schedule-based Service.

7.1.1 Service 1 Execution Analysis

The closed loop sequence of the specified service begins after the end-user has subscribed (has selected to activate) to this specific service in the "Services Marketplace" (IS18). The ISs "Building Energy Modelling and Simulation" (IS5), "DSM Services Advisor" (IS8), and "Multi-Vector Grids Energy Modelling & Simulation" (IS15) are the ones to run first to retrieve data from the field layer through the services broker and publish their results back to the services broker for any ISs to be able to access. Then, "Indoor Air Quality measurement" (IS3) and "Demand Forecasting" (IS1) are the ones to follow on the schedule as they depend on results provided by the previous solutions to forecast and estimate the values of energy and indoor air quality conditions. Afterwards, IS "Flex Forecasting" (IS4) may start to estimate the available flexibility in a day-ahead manner. Then IS "Continuous Energy Performance Management" (IS10) and "Proactive Demand Planning" (IS9) are activated to optimize the usage patterns (demand) both of energy systems (HVACs, RES, storages, EVs, etc.) and domestic shiftable loads. At the end "Visual Analytics Engine" (IS17) and "Building Virtual Model" (IS19) are running in parallel, to allow end-users to visualize results and interact with EVELIXIA in a seamless manner. These two components can provide the latest possible results to the Platform for the users to see and interact with.

The schedule to be created should attempt to facilitate the sequence provided here both for the initial run and the operational service loop repetition frequencies.

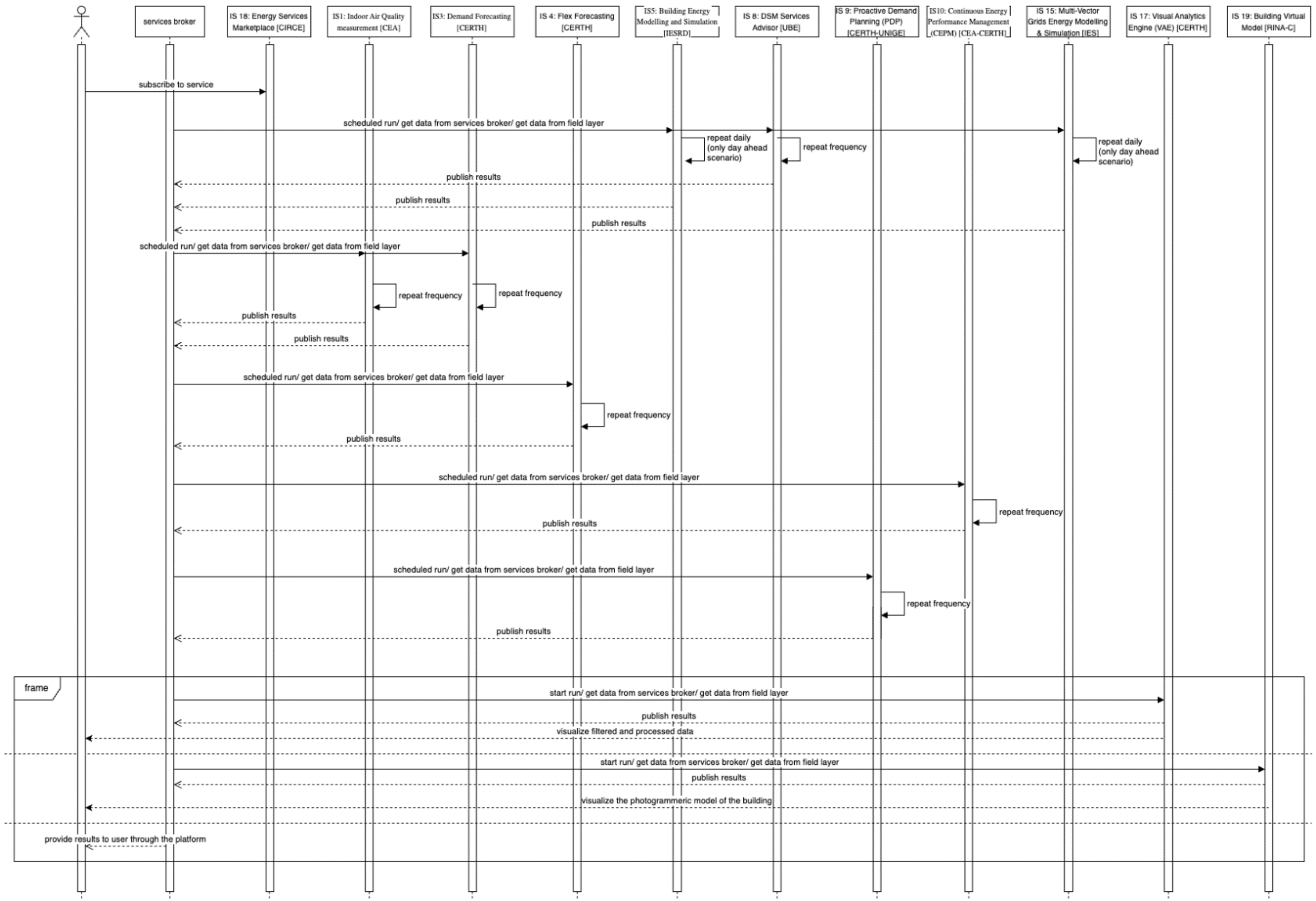


Figure 18 – UML Sequence diagram of Service 1

7.2 SERVICE 2: Building Investment Planning

EVELIXIA's "Building Investment Planning" service focuses on the optimal sizing, parameterization, dimensioning of energy infrastructure to ensure the sustainability, dependability, and efficiency of buildings. This service seeks to contribute strategic insights to building owners and managers that are required to improve the operational efficiency and the energy efficiency achieve long-term sustainability goals This Service is also a behind the meter, on demand Service.

7.2.1 Service 2 Execution Analysis

The end-user registers or selects to activate this service in the "Service Marketplace" (IS18), initiating the closed loop procedure on this service. The first ISs to retrieve data from the services broker are the "DSM Services Advisor" (IS8) and "Digital Building Logbook" (IS16). Afterwards, the former ISs post their results back to the services broker. In addition, the ISs "Energy Assets Maintenance" (IS2) and "Building Investment Planning Assistant" (IS6) and 'SRI Advisor' (IS7) are following, as they depend on the results of the previous ISs. At the end, IS17 'Visual Analytics Engine' and 'Building Virtual Model' (IS19) operate concurrently providing end users with the generated results.

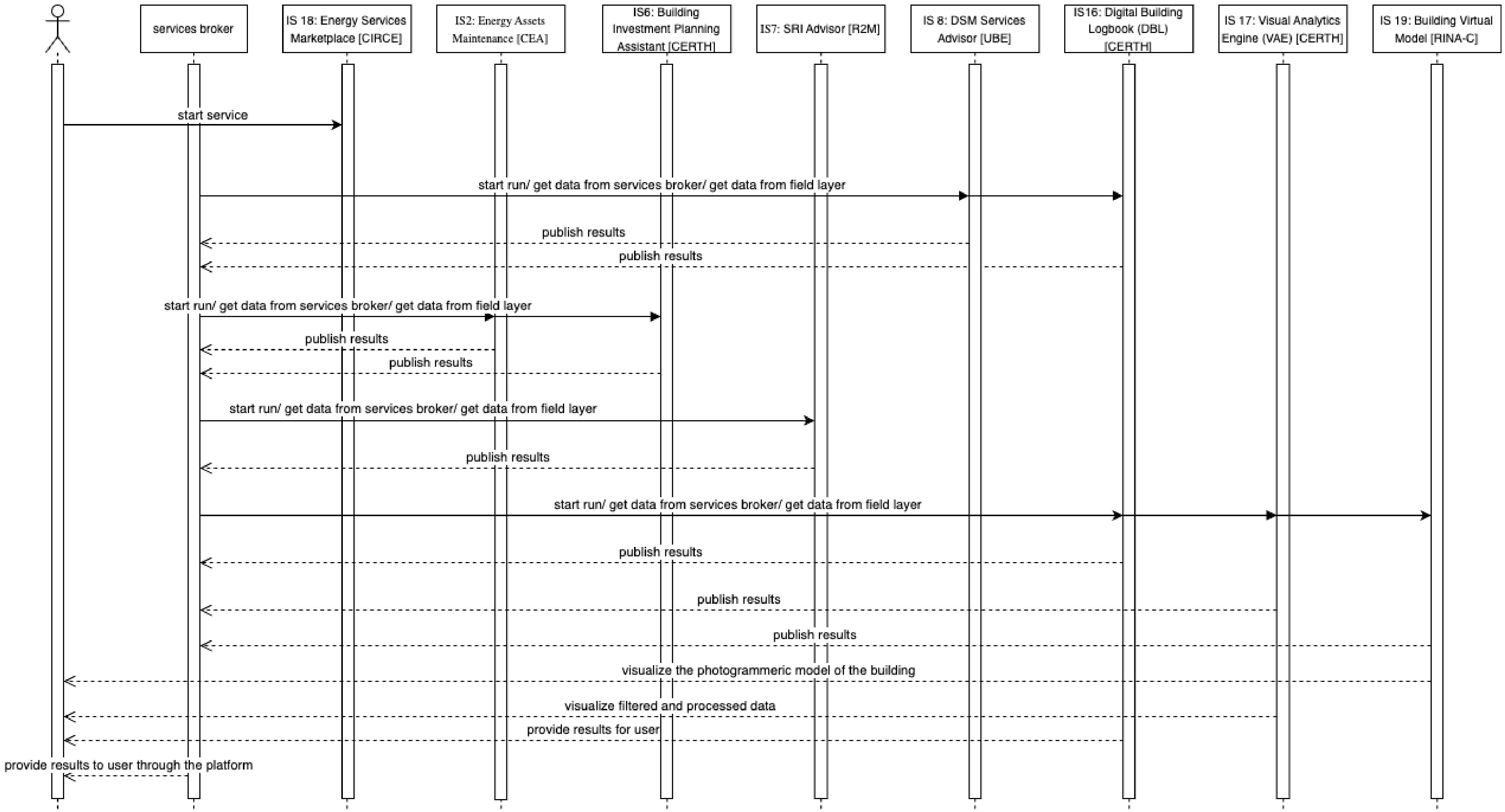


Figure 19 - UML Sequence diagram of Service 2

7.3 SERVICE 3: DER DISPATCH (congestion management)

EVELIXIA's "DER Dispatch (Congestion Management)" service focuses on optimal real-time dispatch based on grid conditions and financial (market) incentives for the available cross-sector distributed generation and storage assets. In addition, this service aims to use available storage on the local network during the expected peak hours to meet local demand, thus reducing congestion levels on the main electricity distribution lines. This service aims to allow DSOs and aggregators to manage and balance energy supply and demand dynamics. The service aims to improve grid resilience, boost operational efficiency, and promote sustainable energy management practices. This service is a front of the meter, schedule-based Service.

7.3.1 Service 3 Execution Analysis

The closed loop sequence of the specified service begins after the end-user has subscribed and activate this specific service in the 'Services Marketplace' (IS18). The innovative solutions "Demand Forecasting" (IS3) and "Multi-Vector Grids Energy Modelling & Simulation" (IS15) should be run first to retrieve data from the field layer via the services broker and post their results back to the services broker for all ISs to access. Next in line is the "Aggregated Demand Portfolio Manager" (IS13) which will require access to outputs from preceding solutions, followed by the "Multi-vector Network Manager" (IS12). Finally, the "Visual Analytics Engine" (IS17) operates to enable users to visualize results and interact seamlessly with EVELIXIA.

This sequence is followed by the operational schedule, which guarantees alignment for both the first execution and future iterations of the operational loop.

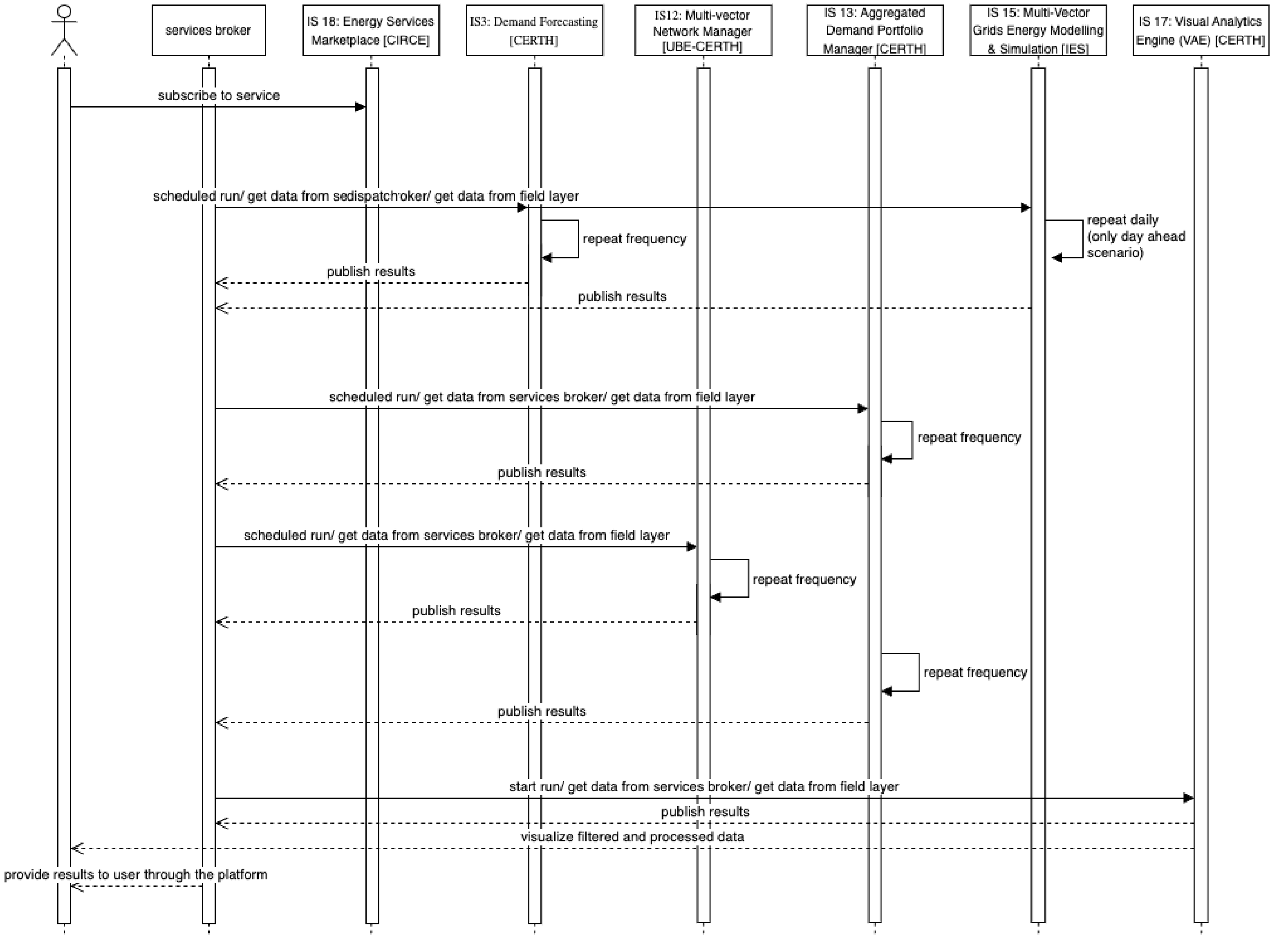


Figure 20 - UML Sequence diagram of Service 3

7.4 SERVICE 4: P2P energy (flexibility) trading

The EVELIXIA "P2P Energy (Flexibility) Trading" service is designed to synchronize energy demand among mutualized, aggregated, or intra-community customers in a decentralized manner, providing grid stability and efficiency. By implementing increase-to-decrease demand schemes, it balances energy loads within communities using tools like grid signal emulation and DR contracts. The primary users of this service are aggregators, DR service providers, large industrial customers, and pooled non-industrial customers, all aiming to optimize energy usage and dynamically respond to grid demands. This service is a front of the meter, schedule-based Service.

7.4.1 Service 4 Execution Analysis

The specified service's closed loop sequence begins after the end user subscribes to this service in the 'Services Marketplace' (IS18). The process starts with the innovative solutions "Demand Forecasting"(IS3) and "Energy Modeling and Multi-Vector Network Simulation" (IS15), which retrieve data from the field level through the service broker and then publish their findings back to the broker so that other ISs can access them. Then "Flex Forecasting" (IS4) receives data from the services broker and starts to estimate the amount of available flexibility The 'Multi-vector network manager' (IS12) performs its functions once the 'Aggregated Demand Portfolio Manager' (IS13) has processed the results of the previous solutions. The 'Visual Analytics Engine' (IS17) is executed at the end of the sequence to allow end users to seamlessly view and interact with the results.

This sequence is followed by the operational schedule, which guarantees alignment for both the first execution and future iterations of the operational loop.

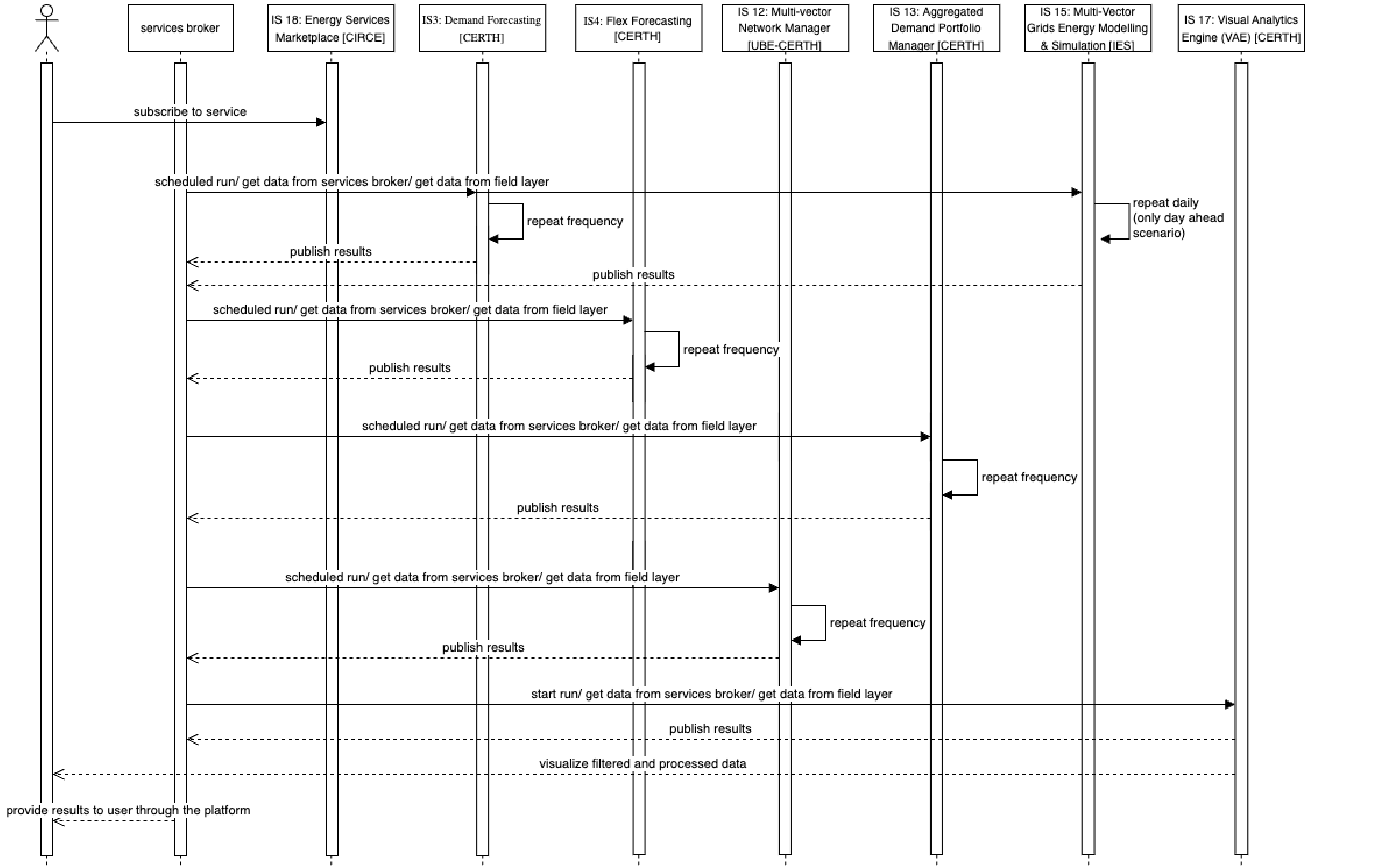


Figure 21 - UML Sequence diagram of Service 4

7.5 SERVICE 5: Portfolio management services (day-ahead/ intra-day)

EVELIXIA provides the “Portfolio management services (day-ahead/intra-day)” service designed for aggregators and DR service providers. This service includes the strategic management of assets to optimise day ahead (spot market) and real-time trading (balancing market) of available flexibility in the energy market and to maximise bidding returns by applying explicit DR schemes in line with market offers/packages/bids. This service also aims at supporting informed selling/bidding of the potential/estimated outcome of demand response actions in electricity markets. This service is a front of the meter, schedule-based Service.

7.5.1 Service 4 Execution Analysis

The specified service's closed loop sequence begins after the end user subscribes to this service in the 'Energy Services Marketplace' (IS18). The process initiates with "Demand Forecasting" (IS3) and "Energy Modeling and Multi-Vector Network Simulation" (IS15), which retrieve field-level data through the service broker and publish their findings back to facilitate access by other components. "Flex Forecasting" (IS4) then estimates available flexibility based on data from the service broker. The "Multi-vector Network Manager" (IS12) operates after the "Aggregated Demand Portfolio Manager" (IS13) processes the previous results. Finally, the "Visual Analytics Engine" (IS17) executes last, enabling end users to interact with and visualize the outcomes seamlessly. This operational sequence adheres to scheduled iterations ensuring consistent alignment and execution.

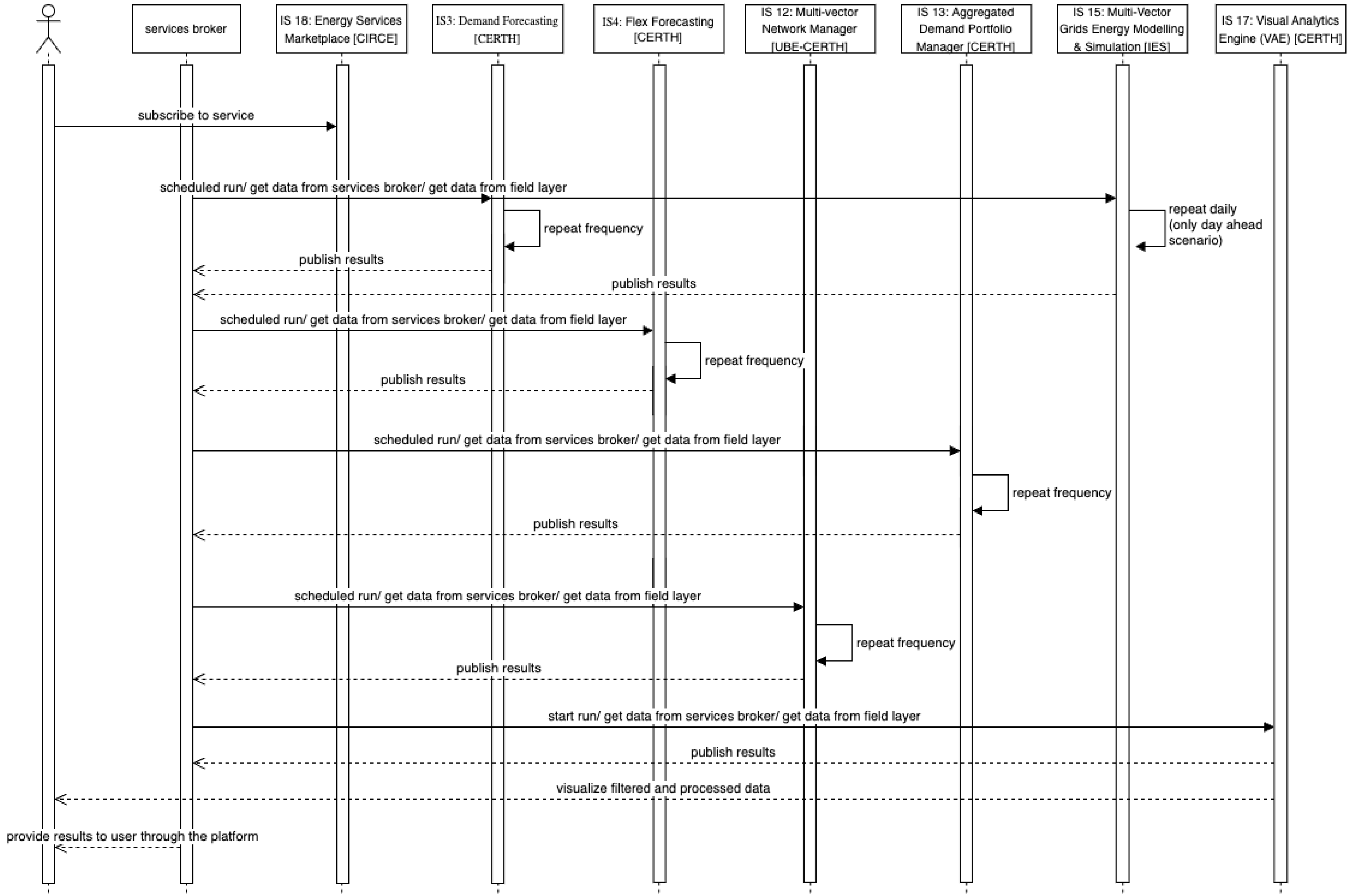


Figure 22 - UML Sequence diagram of Service 5

7.6 SERVICE 6: Network Investment Planning

EVELIXIA's "Network Investment Planning" service is the counterpart of Service 2: "Building Investment Planning", focuses on the optimal sizing, dimensioning and investment strategies design of grid infrastructure to effectively address its current and future requirements in sectors such as electricity, heating, steam and gas. This implies ensuring reliability, efficiency and sustainability in energy/power distribution. This service is important for DSOs providing them with key insights for strategic planning and investment in network infrastructure to effectively support the evolving (expected/projected) energy needs, in an "on-demand" fashion. This service is a front of the meter, schedule-based Service.

7.6.1 Service 6 Execution Analysis

The end-user registers or elects to activate this service in the "Service Marketplace" (IS18), initiating the closed loop procedure on this service. The first ISs retrieve data from the field layer through the services broker are the "Smart Grid Maintenance" (IS14) and "Multi-Vector Grid Energy Modelling & Simulation" (IS15). Afterwards, they post the results back to the services broker for any ISs to be able to access. In the following, the "Grid Investment Planning Assistant" (IS11) gets access to the results provided by the previous solutions to provide recommendations of specific grid investments. Finally, the ISs "Visual Analytics Engine" (IS17), are running in parallel, to allow end-users to visualize the latest possible results and interact with EVELIXIA in a seamless manner.

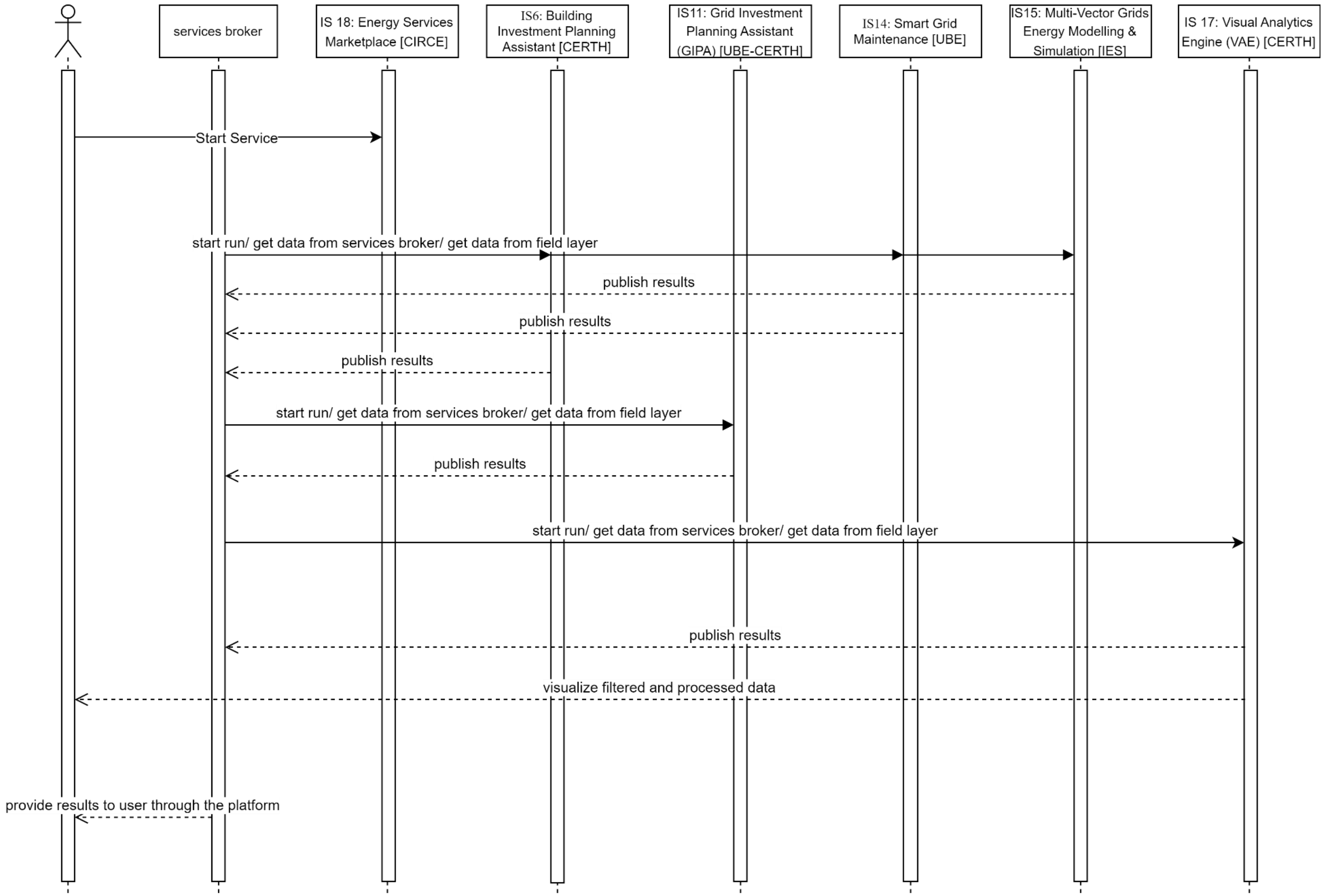


Figure 23 - UML Sequence diagram of Service 6

8 CONCLUSIONS

This document introduced the initial version of EVELIXIA architecture framework. In total, 6 main user services and 23 Innovative solutions along with various sub-modules comprise the core EVELIXIA system. Every responsible partner delivered input about internal architecture, the features, as well as the inputs and outputs along with a brief description of the main components, modules, and sub-modules.

In Chapter 2 there is a detailed identification of the stakeholders along with a map of what services will be offered per stakeholder category.

In Chapter 3 the methodology that has been followed for the design of the framework architecture and the architectural design process are described. Accordingly, in Chapter 4 the conceptual architecture of EVELIXIA is presented along with a description of the logical architecture, the respective horizontal modules with respect to security and user access and interconnectivity principles.

Following the standard IEEE 42010 "Systems and software engineering - Architecture description" three different viewpoints of the system architecture have been presented, namely:

- The Functional View, described in Chapter 5, presents all the system's functional elements, their responsibilities, their functionalities and primary interactions with other elements.
- The Development View, described in Chapter 6, presents the way the front-end dashboard of EVELIXIA and cloud development process are depicted and chosen.
- The Deployment View, described in Chapter 7, describes the deployment of the system, the dependencies of the components, modules and sub-modules along with their physical, software and hardware requirements.
- The Services Information view, described in Chapter 8, outlines the EVELIXIA services and the informational flow as well as their distribution by means of UML sequence diagrams.

Within the next and final version of the EVELIXIA architectural design deliverable (M30), the components will be revised according to the evolution and implementation of the components. It will be a living document, which will be revised until the finalization and the integration of the EVELIXIA framework. To this end, the interconnection of EVELIXIA components with details, especially at the communication part and the deployment part on the cloud, will be presented, improving the descriptions of all four views, functional, development, deployment and information.

9 REFERENCES

[1] "Gaia-X," [Online]. Available: https://gaia-x.eu/.

[2] B. Frost, " "Atomic Design,," 20 October 2021. [Online]. Available: https://bradfrost.com/blog/post/atomic-web-design/ . [Accessed 24 May 2024].

[3] "IDS RAM 4.0," International Data Spaces Association, [Online]. Available: https://docs.internationaldataspaces.org/knowledge-base/ids-ram-4.0.

[4] A. R. A. E. A. E. Boris Otto, "GAIA-X and IDS," Berlin, Germany, January 2021.

10 ANNEXES

10.1 Annex I Overview of EVELIXIA Services

10.1.1 IS01 Indoor Air Quality measurement [CEA]

Component/Service Name:	Indoor Air Quality management
Relevant Project Task:	T4.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS1
Owner / Leading partner:	CEA
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application / Services Layer – [T4.1] Building Awareness & Forecasting Framework
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	CEA intends to develop an Indoor Air Quality forecasting model, taking into account CO2 exposure and potentially the humidity levels inside buildings. This model is to be implemented on the various pilot sites that are interested and candidates. CEA envisions taking into account ventilation rate, occupancy, potentially the states of openings, the pollution sources from outside and inside, if any, to estimate in a short time horizon (e.g. 2 days)

	indoor air quality as equivalent CO_2 equivalent. This forecast of indoor air quality can feed into the energy optimization tools developed within the IS10, to offer sort of flexibilities in terms of ventilation operation and rate while fully meeting the ambient quality level required. Therefore, the ambient air volume will be used as a storage or reserve of qualitative air.
Brief technical description (100-200 words)	CEA has yet to specify the coding environment. It may be appropriate to select an open-source language. But it is definitely a MILP approach that will certify that the trajectory is optimal. Input data such as occupancy, CO2 equivalent pollution sources will be provided by the EVELIXIA information platform, whereas ventilation rate or states of openings can be declared as variables. CEA will tackle the difficulty of linearizing the problem, using limited developments or multivariate adaptative regression splines. This kind of approach would be innovative in this domain. However, it still needs to be confirmed.
Inputs/ IS	
Numerical raw data collected from the available sensory/actuation platforms: Building zone floor area Building zone ceiling height Building zone opening size & orientation Building tightness (total leakage area) Openings state Availability of AHUs Occupancy Categories of occupants Main activities / Indoor humidity sources other than people Indoor CO2 sources other than people Current AHU fan motor operating frequency (from speed drive)	

Outdoor air temperature / Outdoor air humidity Upper and lower bounds for fan motor frequency (AHUs) Outdoor air pressure / Wind speed and direction Outdoor air CO2 concentration/ Indoor air temperature Indoor air humidity / Indoor air pressure Indoor air CO2 concentration – initial value / Upper and lower bounds for indoor air temperature / Upper and lower bounds for indoor air humidity / Upper and lower bounds for indoor CO2 concentration / Upper and lower bounds for indoor air pressure / Upper and lower bounds for air change rate / Upper and lower bounds for fan motor frequency (AHUs) / (IS20) IS5, IS22
Outputs/ IS
Proactive ventilation (rate, duration) strategies, respecting IAQ bounds from international standards Air change rate AHU fan motor frequency / IS10 (decision), IS22 (command), IS17 (recommendations)

10.1.2 IS02 Energy Assets Maintenance [CEA]

Component/Service Name:	Energy Assets Maintenance
Relevant Project Task:	T4.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS2
Owner / Leading partner:	CEA

Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application / Services Layer – [T4.1] Building Awareness & Forecasting Framework, but hosted in the field layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	CEA envisions developing two separated tools dedicated to the Battery Energy Storage Systems (BESS) ageing forecasting and to the Heating & Ventilation (HVAC) systems fault detection, respectively. The ageing forecasting tool will allow to estimate the BESS State of Health (SoH) decrease over the defined time horizon. This variable can be useful to an Energy Management System (EMS) drawing the optimized trajectory of the flexibilities over the time horizon, under the constraint of battery ageing control. CEA plans to develop a methodology using the ageing forecasting model to calculate the parameters of a new optimization model. The tool for HVAC systems will be devoted to fault detection. The nature of the faults to be detected are still to be discussed but it seems appropriate to address topics as follows: anti-freeze heat resistor before the first filtering barrier, faults and operating deviations on pumps or fans. This second tool shall operate with the EMS under constraint of the actual availability of ventilation and heating systems

	and the maintenance agenda in the process of drawing the optimized trajectory. Both tools will rely on simplified models enable to predict the SoH or detect faults by comparison with the measurements.
Brief technical description (100-200 words)	The CEA's BESS ageing forecasting tool is based on Matlab code. The code environment for developing the optimization model is still open. This latter will be easy to linearize, if not linear. Optimization will be performed using a MILP approach. CEA will host both models and their interface on its own IT network, connected to the EVELIXIA platform. The envisioned methodology will also specify the frequency for executing the ageing forecasting model and its time horizon, in order to adjust the parameters of the optimization model at the appropriate timesteps. The ageing forecasting tool will be fed with real data from the field. The HVAC fault detection tool is still to be created. It will rely on one or two models emulating the systems to be scrutinized and by comparing their outputs with the measurements in the field. Beyond detecting potential operating degradation, it aims to set the most suitable date for corrective maintenance. Systems downtimes will then be fed back to the EMS optimization tool, so that a new simulation can be run to adapt the trajectory in consequence. This could be achieved using an open source encoding environment like Python. However, it still needs to be confirmed.
Inputs/ IS	

Numerical raw data collected from the available sensory/actuation platforms: Building zone floor area Building zone ceiling height Building zone opening size & orientation Building tightness (total leakage area) Openings state Availability of AHUs Occupancy Categories of occupants Main activities / Indoor humidity sources other than people Indoor CO2 sources other than people Current AHU fan motor operating frequency (from speed drive) Outdoor air temperature / Outdoor air humidity Upper and lower bounds for fan motor frequency (AHUs) Outdoor air pressure / Wind speed and direction Outdoor air CO2 concentration/ Indoor air temperature Indoor air humidity / Indoor air pressure Indoor air CO2 concentration – initial value / Upper and lower bounds for indoor air temperature / Upper and lower bounds for indoor air humidity / Upper and lower bounds for indoor CO2 concentration / Upper and lower bounds for indoor air pressure / Upper and lower bounds for air change rate / Upper and lower bounds for fan motor frequency (AHUs) / (IS20) IS5, IS22
Outputs/ IS
Proactive ventilation (rate, duration) strategies, respecting IAQ bounds from international standards

Air change rate AHU fan motor frequency / IS10 (decision), IS22 (command), IS17 (recommendations)

10.1.3 IS03 Demand Forecasting [CERTH]

Component/Service Name:	Energy Forecasting Services
Relevant Project Task:	T4.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS3 – Demand Forecasting
Owner / Leading partner:	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer/ Building Awareness & Forecasting Framework
Type:	Software as a service (SaaS)
Brief conceptual description (100-200 words)	In EVELIXIA, CERTH's forecasting services will predict the consumption and production levels of electricity, heating, and gas networks at both the building and district levels, focusing particularly on non-dispatchable plants. The

	module will autonomously determine the most suitable algorithm for each selected case by evaluating a specific set of attributes and metrics. Following its training phase, it will simplify the forecasting process, making it accessible for individuals with less expertise to engage in the practice. An additional functionality regards the clustering of similar load profiles to support Aggregators of large building portfolios in energy management.
Brief technical description (100-200 words)	Based on the popular open-source Python language, the module automatically selects the most promising algorithm capitalizing on either statistical or Machine Learning (ML) approaches based on a set of attributes and metrics, including several network types (ANN, SVR, LSTM, GBT, ARIMA) and topologies (two-stage, ensembled, hybrid). The forecasting will be available as a service through a RESTfull API. The clustering functionality of similar load profiles will be based on generic-purpose clustering algorithms, such as K-means and the Density-based clustering of applications with noise (DBSCAN), which are scalable and perform well for a medium number of clusters. This feature will offer enhanced support to Aggregators of large building portfolios. The clustering service will also be available through the API.
Inputs/ IS	
The incoming data will optimally include 24 values for the system operation during the day. These will used for logging and retraining (if required) the forecasting models. However, the service can also be provided with no input implying that the user requests the forecast for the next day.	

/ (IS20) IS5, IS22
Outputs/ IS
The output of the service will include the hourly (24) values for the forecasted quantity, e.g. building load, photovoltaic generation etc. / IS4,IS9,IS10,IS17

10.1.4 IS04 Flex Forecasting [CERTH]

Component/Service Name:	EVELIXIA's Building Awareness and Forecasting Toolbox
Relevant Project Task:	T4.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 142 of GA)	IS4 - Flex Forecasting
Owner / Leading partner:	CERTH – ITI/C
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/ Services Layer – EVELIXIA Building-to-Grid Services/ Building Awareness & Forecasting Framework
Type:	Software Module /Implemented Functionality

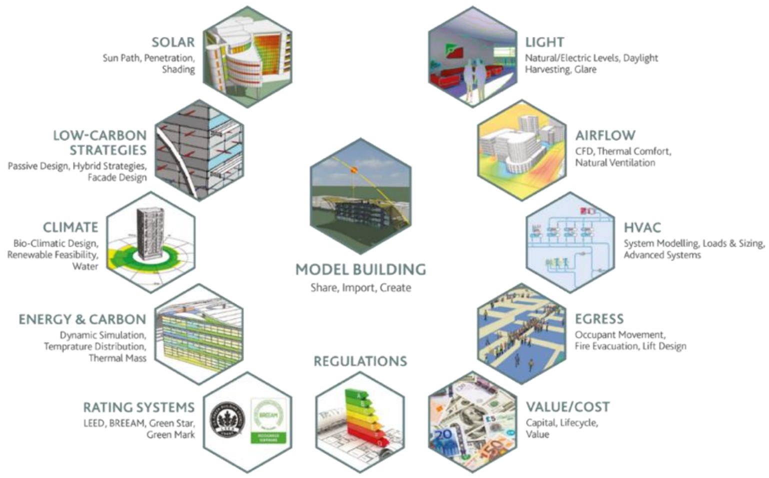
Brief conceptual description (100-200 words)	This tool developed by CERTH focuses on proactively assessing and forecasting demand flexibility at the building level, particularly targeting thermal and electricity demand. The tool initiates its analysis by examining datasets to identify demand response events, categorizing them based on the activation or deactivation of devices. These events play a pivotal role in comprehending energy consumption patterns and optimizing energy systems efficiently. It utilizes data-driven machine learning approaches informed by simulation results to optimize energy usage and resource allocation within buildings or districts. Following event identification, the tool employs a Random Forest Regressor model. Additionally, the tool consolidates room-level data into building-level metrics for energy consumption forecasting. Hourly averages are computed, and Random Forest models are utilized to predict energy consumption based on aggregated features. This enables better understanding and optimization of energy usage within buildings. Overall, the tool offers a comprehensive approach to assess demand flexibility, predict energy consumption, and optimize energy usage within buildings or districts, ultimately contributing to more sustainable and efficient energy management.
Brief technical description (100-200 words)	The tool is implemented as a containerized solution in Python 3.8+, leveraging the robust capabilities of the PANDAS library for detecting and mitigating anomalies in the measured data. Both unsupervised and supervised machine learning mechanisms are employed for outlier elimination, with the selection of appropriate methods influenced by the nature of anomalies observed in the dataset. To assess the

	performance of the tool, a thorough evaluation is conducted by comparing predicted values generated by each forecasting method. Additionally, artificial outliers are intentionally introduced into the dataset to simulate real-world scenarios. These outliers are then filtered and mitigated using the implemented methods, and their performance is evaluated using statistical techniques. The original values from the dataset serve as ground truth for this evaluation process.
Inputs/ IS	
Numerical raw data collected from the available sensory/actuation platforms. This data should encompass information such as indoor and outdoor temperatures, energy consumption and generation metrics, device statuses (on/off), whether the device is heating, cooling, or inactive, the type of device (e.g., wall or ceiling-mounted), and other relevant parameters. This data serves as the foundation for analysing demand response events, forecasting energy consumption, and optimizing energy systems within buildings or districts. I will need data indicative with the following information: 1) General Building Consumption (building level) 2) Global HVAC Consumption (building level) 3) Outdoor temperature 4) Indoor temperature, Humidity, CO2 level (room level) 5) People counting (room level) 6) HVACs monitoring (device state, heating or cooling, setpoints ON/OFF, mode) (room level) 7) HVAC consumption per room 8) Ventilation System and DHW consumption / (IS20) IS3, IS22	
Outputs/ IS	

The service proactively assesses and forecasts demand flexibility levels, targeting both thermal and electricity demand at the building level. / IS9, IS10,IS13,IS17

10.1.5 IS05 Building Energy Modelling and Simulation [IESRD]

Component/Service Name:	Building Energy Modelling and Simulation
Relevant Project Task:	T4.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS5
Owner / Leading partner:	IES R&D
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Applications/Services Layer
Type:	Software Module / Implemented Functionality EVELIXIA Building-to-Grid Services/ Building Awareness & Forecasting Framework
Brief conceptual	The Virtual Environment is IES's historical software and is built around a thermal building dynamic

<table>
<tr>
<td>description (100-200 words)</td>
<td>simulation engine called Apache. VE helps the energy modeller in each phase of model creation providing an easy-to-use UI and navigator tabs that suggest procedures for model creation and analysis. Simulation modules such as solar analysis, lighting assessments, airflow, CFD, life cycle, energy and carbon and building compliance are integrated within the same user interface.

</td>
</tr>
<tr>
<td>Brief technical description (100-200 words)</td>
<td>Central simulation processor which enables you to assess every aspect of thermal performance as well as share results and input across a wide variety of other VE applications.
Features include:
Advanced dynamic thermal simulation at sub-hourly timesteps for better computation of building components
Assess solar gain on surfaces, surface temperatures and radiant exchanges.
Extensive range of results variables for buildings and systems.
Building & room-level annual, monthly, hourly, and sub-hourly analysis.
Assess passive performance, thermal mass, and temperature distribution.</td>
</tr>
</table>

	• Link results from ApacheHVAC, MacroFlo, Suncast & RadianceIES and use as integral thermal simulation inputs. • Ability to run multiple simulations in parallel using the Parallel Simulation Manager • Ability to run recurring simulations on the cloud, using Apache on the cloud module and APIs
Inputs/ IS	
Static and metered data collected from site visits or sensors/meters/bills / (IS20) IS22	
Outputs/ IS	
Energy simulation results / IS1,IS2,IS3,IS4,IS6,IS8,IS10,IS16,IS17	

10.1.6 IS06 Building Investment Planning Assistant [CERTH]

Component/Service Name:	Virtual integrated platform on life cycle analysis (VERIFY)
Relevant Project Task:	T4.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS6 - Building Investment Planning Assistant

Owner / Leading partner:	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer / Autonomous Building Decision Support
Type:	Software as a service (SaaS)
Brief conceptual description (100-200 words)	CERTH's VERIFY will constitute the basis of the EVELIXIA Investment Planning Assistant. It is an online web-based platform performing environmental and cost-oriented analysis and computations. The platform can evaluate the environmental gains and associated cost savings, i.e., life cycle assessment (LCA) and life cycle cost (LCC), for the implementation of RES-based strategies at component and system (e.g. district, grid, etc.) level, taking into consideration the production, exchange and the disposal of all type of energy flow streams, through an automated process. The platform uses programming languages such as Ruby, JavaScript, and Python. In the back end of the VERIFY platform, a PostgreSQL database is responsible for keeping all the important information secure and updated for efficient systems functionality. In the middleware layer, big data analysis is performed through innovative smart grid algorithms. Finally, in the front-end layer, the user interface (UI) provides the ability of easy navigation through multiple web pages and reactive forms. External systems are connected to the platform to

	receive data automatically in real-time through communication channels. The outcome of the platform assists its users to increase economic profits, make environmental-friendly investments, and promote the related social benefits.
Brief technical description (100-200 words)	VERIFY's main advantages compared to commercial tools: 1) the web server approach (no local installation needed), 2) personalized project setup through a friendly user interface, 3) the variety of computations it can perform, both in real-time (use of smart sensors' data) and long term (no capturing data needed - use of VERIFY synthetic data), 4) ability for users to reuse own measurements derived from previous projects, 5) real-time LCA, LCC monitoring and smart sensor control, 6) wide project parametrization through smart grid elements, geolocation, and building construction parameters, 7) LCA and LCC utilization using multiple external data sources, starting from relational databases to large data repositories (Data Lakes). Furthermore, regarding the sorts of applications, VERIFY specializes in sectors including 1) private and public buildings (which may contain production sources), 2) transportation elements (e.g. electric vehicles, charging stations), 3) energy produced by RES and non-RES technologies (e.g. large energy production parks), 4) energy storage systems (e.g. centralized battery energy storage systems) and 5) public infrastructure energy-related elements (e.g. public lighting, transformers). The LCA methodological approach in VERIFY uses specific ISO standards 14040 and ISO 14044. On the other hand, the LCC methodology approach follows ISO 15686-5. In addition to above mention ISO, the methodology followed by ISO

	16745-1, provides requirements for determining and reporting carbon metrics of an existing building, associated with the operation of the building, also addressed through the VERIFY platform.
Inputs/ IS	
The incoming data will include all relevant operational measurement values from the demo sites. Historical values will be used to benchmark the EVELIXIA advances in terms of LCA / LCC outcomes. A list of data sources and available data formats will need to be compiled and partners will need to indicate how they will source the data. / (IS20) IS5, IS22	
Outputs/ IS	
Query posted on Restful API. Parameters will need to be determined and agreed with data consumers. / IS7, IS16,IS17	

10.1.7 IS07 SRI Advisor [R2M]

Component/Service Name:	SRI ADVISOR TOOL
Relevant Project Task:	T4.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS07

Owner / Leading partner:	R2M
Supporting partners / Co-developers:	CERTH
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/ Service Layer. A part of EVELIXIA Building-to-grid services => Autonomous Building DT => Autonomous Building Decision Support Framework
Type:	Tool/service
Brief conceptual description (100-200 words)	The Smart Readiness Indicator Advisor tool will provide Building Owners and Managers with tailored recommendations on how to improve their SRI score (i.e., both the overall SRI score, score of key functionalities, and the score of the 7 SRI impact categories (e.g. energy efficiency, comfort, energy flexibility etc.). The tool will perform different sensitivity analyses based on possible technology renovation packages and/or different flexibility scenarios, with a view to determining the most cost-effective building upgrades to achieve a higher SRI score-performance.
Brief technical description (100-200 words)	In order to tailor the recommendations of the SRI advisor tool to meet the prioritized needs of the building occupants, questionnaires will be designed to adapt the benefits corresponding to each of seven SRI's impact criteria to the user's preference. The outcomes of the user preference questionnaires will be fed into the tool along with, SRI assessment data (collected for the SRI assessment), characterisation of the building (collected from the autonomous digital twin if available - possibly supplemented by direct user input) and building upgrade costs (collected from Building Investment Planning Assistant

	(VERIFY tool) - possibly supplemented by market research and expert knowledge). The objective of the SRI advisor tool is to optimize the most cost-effective upgrade package solution to level-up the SRI score by 1 class.
Inputs/ IS	
(IS20) IS6	
Outputs/ IS	
IS16, IS17	

10.1.8 IS08 DSM Services Advisor [UBE]

Component/Service Name:	DSM Services Advisor
Relevant Project Task:	T4.5: EVELIXIA's Stakeholders Interaction Platform (SIP)
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS8
Owner / Leading partner:	UBE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer/ IP1: Building-to-Grid Services/ Autonomous Building Decision Support Framework (ABDSF) (Figure 2, pg. 141, Autonomous Building Digital Twin)

Type:	Service
Brief conceptual description (100-200 words)	A decision-making/support service at the building level. The tool constitutes a recommendation engine tailored for building operators, which evaluates the capabilities of the building (i.e., consumption and generation patterns, flexibility resources, storage etc.), the needs and offers of the market (i.e., System Operators requirements for different services) and provide a prioritization of grid services provision to the market. By holistically equipping building operators with insights, actionable guidance, and performance evaluation metrics, this integrated engine serves as a cornerstone for optimizing flexibility within distribution networks and paves the way for a resilient, sustainable, and efficient energy future.
Brief technical description (100-200 words)	The tool will be provided as a dockerized microservice image created by MATLAB® Compiler SDK™.
Inputs/ IS	
The DSM Services Advisor will evaluate the capabilities of the building (i.e., consumption and generation patterns, flexibility resources, storage etc.), the needs and offers of the market (i.e., System Operators requirements for different services) and provide a prioritization of grid services provision to the market. / (IS20) IS5	
Outputs/ IS	
IS17	

Funded by the European Union

10.1.9 IS09 Proactive Demand Planning (PDP) [CERTH-UNIGE]

Component/Service Name:	Proactive demand planning service
Relevant Project Task:	T4.2.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS9
Owner / Leading partner:	CERTH
Supporting partners / Co-developers:	UNIGE
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	[T4.2] Autonomous Building Decision Support Framework
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	This framework describes a multi-criteria and multi-level asset for energy management services. A proactive demand planning service/module/component operates, at a high level, in order to proactively provide with day-ahead demand reshaping at building level. The forecasted electricity price information is also considered. Thus, an energy management aggregator has the global view of the system and therefore can generate proper actions fulfilling the energy needs. The produced cumulative energy decision (building level) utilizes cost-

	benefit matrices to enable energy cost-savings without jeopardizing energy efficiency at building level, consumption baseline profiles from IS10, thermal and electricity flexibility graphs from IS4 and local energy consumption and generation forecasting (IS3). This service is only for shiftable components. On top of the HVACs already defined consumption (IS10), schedule other multi-carrier loads (e.g., batteries charging/discharging, electric appliances, EVs charging/discharging) to optimize users time convenience (preference) + total energy costs (considering tariffs as well as local RES expected generation)
Brief technical description (100-200 words)	Model Based and Model Free Reinforcement Learning techniques, alongside with approximate dynamic programming approaches for online non episodic optimization will be investigated, leveraging both literature as well as CERTH's existing inventory. Moreover, transfer learning schemes, where simulation resulted policies could be migrated in different microgrid setups will be also investigated. The current tool does not relate to automatic synchronous control and as a result its core functionality relates mostly to resource management, suggesting for high-level energy usage and policy tuning instead. The implementation of the tool will leverage containerized wrapping developed in Python 3.8.10+ framework to ensure maximized future compatibility.
Inputs/ IS	
Data collected from IS3, IS4 while also price tariffs. / (IS20) IS10	

Outputs/ IS
A time series signal that is the day ahead demand reshaping profile/decision (only for shiftable/flexible loads) to avoid demand peaks, to improve energy costs. / IS13, IS17 (recommendations), IS22 (control)

10.1.10 IS10 Continuous Energy Performance Management (CEPM) [CEA-CERTH]

Component/Service Name:	Continuous energy performance manager service
Relevant Project Task:	T4.2.2.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS10.a
Owner / Leading partner:	CERTH
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	[T4.2] Autonomous Building Decision Support Framework
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	Continuous energy performance manager service (IS10), employing building systems and operations' optimized control in terms of users' comfort, convenience and energy efficiency

	based on black-box policy optimization by CERTH. This is only for HVACs.
Brief technical description (100-200 words)	Model Based and Model Free Reinforcement Learning techniques, alongside with approximate dynamic programming approaches for online non episodic optimization will be investigated, leveraging both literature as well as CERTH's existing inventory. Moreover, transfer learning schemes, where simulation resulted policies could be migrated in different microgrid setups will be also investigated. The current tool does not relate to automatic synchronous control and as a result its core functionality relates mostly to resource management, suggesting for high-level energy usage and policy tuning instead. The implementation of the tool will leverage containerized wrapping developed in Python 3.8.10+ framework to ensure maximized future compatibility.
Inputs/ IS	
Local RES Battery SOC Appliances (shiftable) energy consumption Solar energy generation (current & forecast) Controllable light consumption Ventilation consumption EV charging rate (SoC and capacity of EV) Electricity prices (hourly) District heating buffer tank SoC and energy consumption for electric heater / (IS20) IS3, IS4, IS5	
Outputs/ IS	

A time series signal that is the control decisions implementing the day ahead demand reshaping profile from IS09 for HVACs and domestic appliances. Filtered data streams, where the detected anomalies have been mitigated and can be used for advanced data analysis applications (i.e., WP4). / IS22 (control), IS17 (recommendations)

Component/Service Name:	Continuous Energy Performance Management
Relevant Project Task:	T4.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS10.b
Owner / Leading partner:	CEA / CERTH
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application / Services Layer – [T4.2] Autonomous Building Decision Support Framework
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	CEA is very keen to develop an Energy Management Service (EMS), as a Model Predictive Control (PMC), first to ensure matching energy supply and demand while optimizing the costs or the use of the

	Renewable Energy Sources (RES) at the expense of the carbonized ones, depending on the way to set the models. This PMC is intended to be able to calculate a new optimal trajectory while the previous one is no longer feasible due to the occurrence of hazards (weather or loads uncertainties…). CEA plans to rely on its homemade so-called mOre Gams Library. This software pool is well suited to manage use cases embedding all components: Battery Energy Storage System (BESS), Charging Station, Grids, Loads, RES, Building…. This solution is mainly dedicated to the Greek and French pilot sites. The scope of the use-cases is still to define.
Brief technical description (100-200 words)	CEA intends to develop an EMS setting its existing component models and splitting the optimization management into different sub controllers, Energy Management Module (EMM) and Power Management Modules (PMM). CEA will then, use a multi-level approach (from EMM down to RTC), based on a multi-time scale optimization scheme. CEA will leverage the field data collected via the EVELIXIA platform and consider the previous trajectory to correct potential deviations and to refine the optimal trajectory. The outputs of the IS1 (IAQ management) and IS2 (extension of BESS life range) might be considered for a more holistic optimization approach. At least, the PMC could be informed of the components availability, those involved into the system considered, retrieving the outcomes from the IS1, IS2 (ventilation breakdown, HVAC system fault detection, Battery SOH).
Inputs/ IS	

Needs are the following: - IS1 (IAQM) and IS2 (EAM) inputs and outputs - Information on building's envelop and facilities, size, structure, lighting, heating and cooling systems (with power boundaries) - Information on occupancy and setpoint temperatures with boundaries - eV charging station information with charging/discharging current boundaries - Information on energy storage systems (battery, tank...) in terms of capacity, availability with the boundaries (charge/discharge current, efficiency, power...) - Information on Power-H2-Power and H2 storage facilities, capacity, power, flowrate with upper and lower limits (IS28) - Information from the Flexibility Demand Forecast (IS4) - Information on grid capacity and status (upper and lower bounds of power supply, mains voltage, mains frequency, availability...) - Current or periodic status of consumers, generators in terms of power consumption, capacity, voltage, current, temperature, level, SoC, SoH, availability - Current and forecast weather conditions (solar radiation, wind, temperature) - Current and forecast energy generation profiles (RES) --> which IS ? - Current and forecast energy consumption profiles from IS3 (eVs charging/discharging, Building's appliances, HVAC systems including HWTank, lighting, H2 facility, RES and storage systems including auxiliaries...) - Energy market Prices - Grid services commitments (FCR, aFRR...) - Flexibility trajectory from the IS9 (PDP) / IS1, IS2, IS3, IS4, IS5, IS9, IS12, IS13, IS14, IS18, IS22, IS28
Outputs/ IS
A time series signal that is the day ahead demand reshaping profile. Filtered data streams, where the detected anomalies have been mitigated and can be used for advanced data analysis applications (i.e., WP4). The output variables are the following: + Energy flow from/to the grid

+ Flexibility trajectories as power flows (battery and eV charging / discharging, Buildings loads shifting / shedding) + Outputs from the IS1 (IAQM) and IS2 (EAM) + Flexibility trajectories as power flows (battery and eV charging / discharging, Buildings loads shifting / shedding) / IS22 (retrieved by system operators), IS17 (recommendations)

Component/Service Name:	Continuous Energy Performance Management with District Heating Production
Relevant Project Task:	T4.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS10.c
Owner / Leading partner:	NEOGRID
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer / Autonomous Building Decision Support
Type:	Software Module / Implemented Functionality
Brief conceptual	The service displays and operates the pool of assets connected to the Danish Pilot Site in

description (100-200 words)	Aabenraa to cost-optimize energy operation and decentralized production of district heating. It is an addon to Neogrid's existing public API. With Building Aggregated Service (BAS), it is possible to get details about the connected assets and see a list of the supported use cases. Based on extracted measurement data an optimized operation schedule can now be sent to the BAS

Brief technical description (100-200 words)	Service details
	• Display Danish Pilot Site characteristics
	    o Asset characteristics (like type, flexibility, constraints)
	    o Asset types defined so far:
	        ▪ EV charger, Danfoss ECL, DHW tank, Accumulation tank, Smart plugs, ventilation, lights, PV and battery
	• Optimization and operation strategies according to a list of use-cases
	• Data collections for external visualization and analysis
	
	Control Input from external optimized operation schedules

10.1.11 IS11 Grid Investment Planning Assistant (GIPA) [UBE-CERTH]

Component/Service Name:	Grid Investment Planning Assistant (GIPA)
Relevant Project Task:	T4.4
Relevant Project Innovative Solution (IS):	IS11

(section 1.1.2, pg. 140 of GA)	
Owner / Leading partner:	UBE
Supporting partners / Co-developers:	CERTH/CPERI
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer/ IP2: EVELIXIA Grid-to-Building Services/ Autonomous Network Decision Support Framework (ANDSF) (Figure 3, pg. 144, Autonomous District Digital Twin)
Type:	Service
Brief conceptual description (100-200 words)	A service evaluating current and future network bottlenecks and assessing the necessity of potential infrastructural future enhancements at the grid level in terms of economic benefit (direct investment profits from network enhancements) and economic viability. More specifically, the GIPA purpose is the identification of the energy and capacity needs for the local energy network based on multi-vector congestion forecasting, based on long-term projected district scenarios and capacity building analysis from RES and storage integration, energy demand evolution (all these data will be fed by the iVN simulations at the grid level) and evaluate how specific grid investments, either on the generation side or the energy distribution and substation upgrades can impact the overall social welfare. The tool will valorise the cross-vector energy system modelling tools of EVELIXIA (i.e., the iVN – IS15) to study the impact to the social welfare of various grid upgrade scenarios i.e., re-conductoring, new transformers and

	substations, in comparison with efficient smart grid scheduling optimization.
Brief technical description (100-200 words)	The tool will be provided as a dockerized microservice image created by MATLAB® Compiler SDK™.

Inputs/ IS
Simulation models' data at the grid level (e.g., grid assets' dimensioning from power flow analysis, grid level congestion forecasting) from iVN engine (IS15) Key financial variables at the building level from VERIFY LCC module (IS6) / (IS20) IS6, IS15

Outputs/ IS
Recommendations of specific grid investments, either on the generation side or the energy distribution and substation upgrades, based on CBA methodologies (e.g., ENTSOe guidelines). / IS17

10.1.12 IS12 Multi-vector Network Manager [UBE-CERTH]

Component/Service Name:	Multi-vector Network Manager
Relevant Project Task:	T4.4

Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS12
Owner / Leading partner:	UBE
Supporting partners / Co-developers:	CERTH/ConvCAO
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer/ IP2:EVELIXIA Grid-to-Building Services/ Autonomous Network Decision Support Framework (ANDSF) (Figure 3, pg. 144, Autonomous District Digital Twin)
Type:	Service
Brief conceptual description (100-200 words)	A service responsible for enabling grid services at the distribution level, covering the different cross-vector cases, exploiting the vector-specific network management capabilities without violating operational bounds of the different energy networks (leveraging the grid level simulation capabilities regarding power-flow analysis across multiple energy vectors of the District Digital Twin – iVN, IS15). More specifically, the Multi-vector Network Manager will enable grid operators negotiating and balancing energy demand with supply through differently placed bids in different market designs and schemes and respond to specific mid (daily) and long-term (monthly) grid balancing needs in a profitable (cost-efficient) manner for them, being adapted to fit specificities of microgrids and active building nodes while de-risking grid operations (reserves and congestion management).

	This tool will leverage the capabilities of the buildings' cross-vector resources for fast response to electricity exchanges (IS13) to create appropriate flexibility bids.
Brief technical description (100-200 words)	The tool will be provided as a dockerized microservice image created by MATLAB® Compiler SDK™.
Inputs/ IS	
Data regarding flexibility pools from the different buildings of the grid from the Aggregated Demand Portfolio Manager (IS13) / (IS20) IS13, IS15	
Outputs/ IS	
Predictive congestion management and non-spinning reserves planning for DSOs. / IS17	

10.1.13 IS13 Aggregated Demand Portfolio Manager [CERTH]

Component/Service Name:	An aggregated demand portfolio manager service
Relevant Project Task:	T4.4.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS13
Owner / Leading partner:	CERTH

Supporting partners / Co-developers:	
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	[T4.4] Autonomous District Decision Support Toolbox
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	An aggregated demand portfolio manager service (IS13), responsible for real-time daily portfolio replanning/rescheduling – as soon as the observed state differs from the expected one (assuming a deterministic operational energy environment) the rapidly flexible/shiftable expected demand amounts at an aggregated building and district scale, in order to enable proactive resources (demand flexibility) aggregation and allow active participation in energy balancing markets (mostly focusing on the electricity vector). Based on the expected optimized consumption as well as the remainder flexibility amounts by the different flexibility assets per individual building (type of flexibility - activation or curtailment response times - that was left unused after the demand of the different buildings has been optimized by IS9 and IS10), optimize the mixture of the available (unused) flexibility amounts that could be selected from each building which - when accumulated - could meet specific: a) short-term (10-15 min decision frequency for the next 1-2 hours control horizon) and b) mid-term (1 hour decision frequency for the next 12 or 24 hours control horizon) DR balancing requests by the DSO, in the most cost efficient manner for the energy retailer (aggregator).

Brief technical description (100-200 words)	Model Based and Model Free Reinforcement Learning techniques, alongside with approximate dynamic programming approaches for online non episodic optimization will be investigated, leveraging both literature as well as CERTH's existing inventory. Moreover, transfer learning schemes, where simulation resulted policies could be migrated in different microgrid setups will be also investigated. The current tool does not relate to automatic synchronous control and as a result its core functionality relates mostly to resource management, suggesting for high-level energy usage and policy tuning instead. The implementation of the tool will leverage containerized wrapping developed in Python 3.8.10+ framework to ensure maximized future compatibility.
Inputs/ IS	
(IS20) IS3, IS4, IS9, IS10, IS15	
Outputs/ IS	
A time series signal that is the day ahead demand reshaping profile/decision. / IS12, IS17	

10.1.14 IS14 Smart Grid Maintenance [UBE]

Component/Service Name:	Smart Grid Maintenance
Relevant Project Task:	T4.3

Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS14
Owner / Leading partner:	UBE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Application/Services Layer/ IP2: EVELIXIA Grid-to-Building Services/ Network Awareness & Forecasting Framework (NAFF) (Figure 3, pg. 144, Autonomous District Digital Twin)
Type:	Service
Brief conceptual description (100-200 words)	A service that supports smart maintenance outage planning for the grid assets that are interconnected with buildings, to extend the capabilities of iVN engine (IS15). The tool performs assessment of the health/degradation levels of multi-grid related assets (e.g., BESS, MV/LV substations, PVs, HPs, power lines) when connected to electrical networks (main energy carrier), based on black-box data driven models (predictive maintenance and infrastructure ageing models) and co-optimization of the outage scheduling of grid assets with buildings' DERs' generation. The service blends optimal scheduling algorithms of DER assets with health assessment models of grid assets, to achieve the optimization of the maintenance planning of the grid assets, taking into consideration the buildings and their DERs capabilities for energy

	services, thus avoiding power cut-offs, during which buildings can provide services.
Brief technical description (100-200 words)	The tool will be provided as a dockerized microservice image created by MATLAB® Compiler SDK™.
Inputs/ IS	
(IS20) IS15, IS22	
Outputs/ IS	
IS16, IS17	

10.1.15 IS15 Multi-Vector Grids Energy Modelling & Simulation [IES]

Component/Service Name:	Multi-Vector Grids Energy Modelling & Simulation (iVN)
Relevant Project Task:	T4.3
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS15
Owner / Leading partner:	IES
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal	

(Figure 6, pg. 151 of GA)	
Type:	Software Module
Brief conceptual description (100-200 words)	IES iVN (intelligent Virtual Network) is a district modelling tool for performing simulations of city/community-level energy distribution networks using a combination of accounting, physics, and engineering concepts, such as energy conservation and power-flow analysis, across multiple energy vectors. The iVN uses a combination of 3D building geometry models and 2D schematic tools to model city infrastructure. It can then perform simulations, which aggregate the demand for commodities, such as electricity and heat, and allocate loads to providers to meet the demand.
Brief technical description (100-200 words)	The purpose of the iVN is to calculate high-level estimations of the current performance of various utilities and quantitatively predict the impact of changes to city-infrastructure. The following are a small sample of the questions that the IVN can address: What is the impact of replacing a coal-fired power station with a Combined-Cycle Gas Turbine (CCGT) power station? What is the reduction in CO_2 emissions, and will alternative energy supplies be required to meet demand? What is the total reduction in carbon emissions associated with installing solar panels on every home in a particular suburb? What is the performance of the solar panels based on the weather found at the location of the city/district? How capable is an electrical storage device in meeting peak demand when energy provision is lower than the demand?

	What is the impact of an additional suburb on the existing power grid or district heating network? Can existing utilities manage an increase in demand within existing constraints? If not, what changes to energy storage or local generation are required to achieve this? What is the impact of building energy efficiency improvements (such as insulation, double-glazing or efficient central heating) on the electricity grid or district heating demand?
Inputs/ IS	
Static grid information and real-time sensory measurements {Added a comment in this cell for more detail} / (IS20) IS22	
Outputs/ IS	
Total demand (or consumption) of electricity at this node due to attached prosumers, attached installations and child node residual demands. Total generation (or provision) of electricity at this node due to attached prosumers and installations. Total generation (or provision) of electricity at this node due to renewable installations. Total generation (or provision) of at this node due to non-renewable installations. Total electricity charged to storage. Total electricity discharged from storage. Current amount of electrical energy sitting in storage. Total losses whilst charging electricity storage installations. Total losses whilst discharging from electricity storage installations. Total losses due to leakage from electricity storage installations.	

Total losses associated with electricity storage installations attached to this node.
Losses associated with electricity conversion at the node for energy feeding back to the network.
Losses associated with electricity conversion at the node for energy meeting demand in the network.
Losses associated with transmittance of electricity along the link between this node and the parent node.
Total electricity losses from all sources
The unmet electricity demand passed up to the parent node.
The CO_2 factor associated with direct and indirect carbon emissions at this node.

Total demand (or consumption) of heat at this node due to attached prosumers, attached installations and child node residual demands.
Total generation (or provision) of heat at this node due to attached prosumers and installations.
Total generation (or provision) of heat at this node due to renewable installations.
Total generation (or provision) of heat at this node due to non-renewable installations.
Total heat charged to storage.
Total heat discharged from storage.
Current amount of heat sitting in storage.
Total losses whilst charging thermal storage installations.
Total losses whilst discharging from thermal storage installations.
Total losses due to leakage from thermal storage installations.
Total losses associated with thermal storage installations attached to this node.
Losses associated with heat conversion at the node for energy feeding back to the network.
Losses associated with heat conversion at the node for energy meeting demand in the network.
Losses associated with transmittance of heat along the link between this node and the parent node.
Total heat losses from all sources

The unmet heat demand passed up to the parent node. The CO2 factor associated with direct and indirect carbon emissions at this node. Total demand (or consumption) for cooling at this node due to attached prosumers, attached installations and child node residual demands. Total generation (or provision) for cooling at this node due to attached prosumers and installations. Total cooling generation (or provision) at this node due to renewable installations. Total cooling generation (or provision) at this node due to non-renewable installations. Total cooling charged to storage. Total cooling discharged from storage. Current amount of cooling sitting in storage. Total losses whilst charging cooling storage installations. Total losses whilst discharging from cooling storage installations. Total losses due to leakage from cooling storage installations. Total losses associated with cooling storage installations attached to this node. Losses associated with cooling conversion at the node for energy feeding back to the network. Losses associated with cooling conversion at the node for energy meeting demand in the network. Losses associated with transmittance of cooling along the link between this node and the parent node. Total cooling losses from all sources The unmet cooling demand passed up to the parent node. The CO2 factor associated with direct and indirect carbon emissions at this node. Total waste heat generated by prosumer units or cooling generators which requires processing Total waste heat processed by cooling towers Losses associated with heat conversion at the node

Losses associated with transmittance of waste heat along the link between this node and the parent node. Total waste heat lost to the environment from all sources The unprocessed waste heat passed up to the parent node. Total demand (or consumption) of potable water at this node due to attached buildings, attached installations and child node residual demands. Total generation (or provision) of potable water at this node due to attached buildings and installations. Total generation (or provision) of potable water at this node due to renewable installations. Total generation (or provision) of potable water at this node due to non-renewable installations. Total potable water charged to storage. Total potable water discharged from storage. Current amount of potable water sitting in storage. Total losses whilst charging potable water storage installations. Total losses whilst discharging from potable water storage installations. Total losses due to leakage from potable water storage installations. Total losses associated with potable water storage installations attached to this node. Losses associated with potable water fed back to the network. Losses associated with potable water manipulations at pumping stations or other intermediate infrastructure at the node. Losses associated with transmittance of potable water along the link between this node and the parent node. Total potable water losses from all sources The unmet potable water demand passed up to the parent node. The CO2 factor associated with direct and indirect carbon emissions at this node. Total amount of wastewater at this node due to attached buildings, attached installations and child node residual waste. Total wastewater processed at this node due to attached buildings and installations. Losses associated with wastewater fed down the network.

Losses associated with wastewater manipulations at pumping stations or other intermediate infrastructure at the node.
Losses associated with transmittance of wastewater along the link between this node and the parent node.
Total wastewater losses from all sources
The residual wastewater passed up to the parent node.

The total electricity demand
The total electricity generation
The total district heating demand
The total district heating generation
The total district cooling demand
The total district cooling generation
The total potable water demand
The total potable water generation
The total waste heat processed by this building
The total waste heat produced
The total wastewater processed
the total wastewater produced
The total demand for commodity "x". Replace "x" with a commodity of your choosing, e.g. "Gas demand" or "Cheese demand"
The total generation for commodity "x". Replace "x" with a commodity of your choosing, e.g. "Gas generation" or "Cheese generation"
The total direct CO_2 emissions associated with fuel consumption.
The calculated indirect CO_2 emission due to the sourcing of electricity from the network
The calculated indirect CO_2 emission due to the sourcing of heat from the network
The calculated indirect CO_2 emission due to the sourcing of cooling from the network
The calculated indirect CO_2 emission due to the sourcing of potable water from the network
The total calculated indirect CO_2 emission
The calculated CO_2 factor considering both direct and indirect CO_2 emissions from the consumption of power.
The calculated CO_2 factor considering all CO_2 emissions from the consumption of fluids (water).

Electricity taken from the electricity network to charge something else

Electricity generated by the CHP installation
Heat generated by the CHP installation
Fuel consumed by the CHP installation (given in terms of energy released on combustion)
Rate of CO_2 emitted by this installation due to combustion

Cooling generated by the cooling generator
Waste heat generated by the cooling generator
Electricity consumed to operate the cooling generator
Efficiency of the cooling generator

Cooling placed into storage
Cooling released from storage
Amount of cooling currently sitting in storage
Cooling losses due to charging (due to, for example, intermediate heat exchange)
Cooling losses due to discharging (due to, for example, intermediate heat exchange)
Cooling losses due to leakage from the store (due to, for example, coolant leakage or heat exchange with surroundings)
Total cooling losses

Waste heat processed by the cooling tower
Electricity consumed to operate the cooling tower
Potable water consumed by the cooling tower
Total water withdrawn by the cooling tower (same as consumption).
The coefficient of performance for the entire cooling tower installation

Amount of water provided per second to the node
Heat consumed to operate the desalination plant
Electricity consumed to operate the desalination plant

Heat transferred by the heat pump (includes waste heat)
Electricity consumed to operate the heat pump (i.e. work done)

The coefficient of performance of the electric heat pump

Electricity placed into storage
Electricity released from storage
Amount of electrical energy currently sitting in storage
Electricity losses due to charging (due to, for example, resistance losses)
Electricity losses due to discharging (due to, for example, resistance losses)
Electricity losses due to leakage from the store (due to, for example, current leakage and induction)
Total electricity losses

The CO2 emissions associated with metered CO2 (not fuel combustion)
The CO2 captured by on-site carbon capture technology (offsets emissions)
The amount of cooling required by this installation
The amount of cooling provided by this installation
The amount of electricity required by this installation
The amount of electricity generation by this installation
The yield obtained from the consumption of gas
The amount of gas supplied (in terms of yield)
The amount of heat required by this installation
The amount of heat generated by this installation
The amount of potable water required by this installation
The amount of potable water provided by this installation
The amount of waste heat processed (i.e. rejected) by this installation
The amount of waste heat produced by this installation
The amount of wastewater processed (i.e. treated and disposed) by this installation
The amount of wastewater produced by this installation
The total yield obtained from the consumption of all fuels
Total CO2 emitted by this installation (from combustion of fuels and from metered consumption)

Heat generated by the heat generator
Efficiency of the heat generator (can exceed 1... this variable is more like a coefficient of performance)

Fuel consumed by the heat generator (given in terms of energy released on combustion) CO_2 emitted by this installation from combustion Electricity consumed in the ballast or LED driver and lost as heat energy Electricity consumed by the lamp Total electricity consumed by the lighting installation (sum of both ballast and lamp electricity consumptions) Water placed into storage Water released from storage Amount of water currently sitting in storage Water losses due to charging (due to, for example, leaks in the filling pipe) Water losses due to discharging (due to, for example, leaks in the drainpipe) Water losses due to leakage from the store (due to, for example, leaks in the water tank) Total water losses Fraction of total electricity generated by the steam turbine that is used to power auxiliary equipment at the power plant (i.e. parasitic power). Electricity generated by the power station which heads over to the grid (before line losses). Electricity generated by the steam turbine. This is the sum of the grid electricity generation and the auxiliary power consumption. Whole-plant efficiency. It is defined as the ratio between the grid electricity generation and the fuel consumption Heat yielded from fuel that requires cooling by the cooling system. Heat lost from the power plant via the flue gas exhaust pipe or other direct heat loss mechanisms. This loss is proportional to the fuel consumption. Fuel consumed by the power station (given in terms of energy released on combustion; a.k.a. heat rate) Waste heat processed on-site by the power plant's cooling system. Water consumed by the power station (typically by the cooling system).

Water withdrawn from the potable water supply or from local freshwater resources in order to operate the power plant (typically required by the cooling system)

Direct CO2 emitted by this installation from combustion

Electricity generated by the CHP installation

Incident irradiance on the panel's surface per unit area.

Heat generated by the Solar water heater

Electricity consumed to operate the solar water heating equipment

Potable water consumed during operation

Heat placed into storage

Heat released from storage

Amount of heat energy currently sitting in storage

Heat losses due to charging (due to, for example, intermediate heat exchange)

Heat losses due to discharging (due to, for example, intermediate heat exchange)

Heat losses due to leakage from the store (due to, for example, water leakage or heat exchange with surroundings)

Total heat losses

Electricity generated by the installation

Amount of influent wastewater processed

Electricity consumed to operate the desalination plant

The amount of electricity consumed per unit volume of influent wastewater treated

Amount of water provided per second to the node

Electricity generated by the CHP installation

The wind speed corrected for the height of the turbine blades

/

IS11, IS12, IS13, IS14, IS17

10.1.16 IS16 Digital Building Logbook (DBL) [CERTH]

Component/Service Name:	Digital Building Logbook (BDL)
Relevant Project Task:	T4.5: EVELIXIA's Stakeholders Interaction Platform (SIP)
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS 16
Owner / Leading partner:	CERTH/ ITI-V
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	T4.5: EVELIXIA's Stakeholders Interaction Platform (SIP) At the top of Application/ Services Layer
Type:	Service
Brief conceptual description (100-200 words)	The dynamic digital building logbook will be able to digitally capture and store the relevant operational data and technical documentation. The users of the DBL will be able to extract tailor-made reports regarding building's technical systems.
Brief technical description (100-200 words)	The module will be developed using Python programming language, to be installed in the dockized back-end services environment. It will

	be able to interact with the buildings DT database in order to track any building modifications and log them. The DBL will serve as a common building data environment and repository. The relevant APIs and connectors will be developed to infuse/transmit BIM/raw data will be developed to support the extraction of data from EVELIXIA BDT and DDT applications.
Inputs/ IS	
Static building data archive / (IS20) IS5 (shared static building archive for developing the simulation models / DTs)	
Outputs/ IS	
The VAE will provide the high-level overview of building operational states, reflected by the BDT real-time data monitoring flexibility potential different building and grid actors' insight needs	

10.1.17 IS17 Visual Analytics Engine (VAE) [CERTH]

Component/Service Name:	Visual Analytics Engine (VAE)
Relevant Project Task:	T4.5: EVELIXIA's Stakeholders Interaction Platform (SIP)
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS 17

Owner / Leading partner:	CERTH/ ITI-V
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	T4.5: EVELIXIA's Stakeholders Interaction Platform (SIP) At the top of Application/ Services Layer
Type:	Service
Brief conceptual description (100-200 words)	The VAE will provide the high-level overview of building operational states, reflected by the BDT real-time data monitoring flexibility potential different building and grid actors' insight needs. It will also provide useful insights as well as the network location of available flexible assets and their capacities via advanced visual analytics. and input available from the Building Energy Modelling and Simulation and Multi-Vector Grids Energy Modelling & Simulation components.
Brief technical description (100-200 words)	Angular programming language will be used alongside with proper visualization libraries built in a docker container which will be deployed on a server.
Inputs/ IS	
Visualize filtered data (IS21) and post processed data (IS results) / (IS20) IS1-IS15, IS18, IS21->IS22	
Outputs/ IS	

The VAE will provide the high-level overview of building operational states, reflected by the BDT real-time data monitoring flexibility potential different building and grid actors' insight needs.

10.1.18 IS18 Energy Services Marketplace [CIRCE]

Component/Service Name:	Data Management Component Subscription Management Component Data Upload Component Metadata Exploration Component"
Relevant Project Task:	T4.5
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS18 Energy Services Marketplace
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Applications/Services Layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	The Marketplace consists of four main components: "Data Management",

	“Subscription Management”, “Data Upload”, and “Metadata Exploration”. These components work together to manage data storage, retrieval, sharing, user subscriptions, data upload, and metadata exploration.
Brief technical description (100-200 words)	The Marketplace provides a platform that facilitates interaction between different stakeholders and users. It manages data, handles subscriptions, allows data upload, and enables metadata exploration. The components interact with each other as per the execution flow to fulfil these functionalities.
Inputs/ IS	
The system accepts data subscriptions from stakeholders and services subscription requests from users. The tool is responsible to deploy appropriate containerised IS tools to provide the selected EVELIXIA services. / (IS20) IS22	
Outputs/ IS	
The system provides data, subscription status, and metadata (JSON, variable, stakeholders/users). The “Data Management” component generates these outputs based on the processed inputs and passes them to the respective components for delivery to the users. / IS10	

10.1.19 IS19 Building Virtual Model [RINA-C]

Component/Service Name:	Building Virtual Model

Relevant Project Task:	T4.5
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS19
Owner / Leading partner:	RINA-C
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Stakeholder Interaction Platform
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	Rina Virtual Model tool is an interactive web based 360° Virtual photogrammetric 2D model of a Plant, Site or Building, allowing its user to virtually navigate through specified assets. The virtual model will be used to build virtual navigation models of the Pilot Sites. A new picture shooting methodology will be used, allowing to dynamically scale the details of the reproduced asset, providing automatically the optimized 360° picture location on the building plot plan, to ensure the right trade-off between the physical shot and level of virtual model details. The pictures will be taken with a high-resolution (360°) camera. The model will also be able to show inside the model data generated from the sensors.

Brief technical description (100-200 words)	The virtual model will be created using a specific commercially available software and the data generated by the sensors will be integrated developing a custom application using Angular and JavaScript. The model will be accessible through a standard link and the measurements of from the sensors will be embedded in the model. The data will be displayed using graphs and tables depending on the type of data that needs to be shown.
Inputs/ IS	
Numerical data collected from the available sensors The photogrammetric model of the building. / (IS20) IS16, IS22	
Outputs/ IS	
There are no outputs that can be used from other tools. The tool only visualizes the photogrammetric model of the building.	

10.1.20 IS20 Blockchain-based Infrastructure [CERTH-SOLVUS]

Component/Service Name:	Blockchain infrastructure
Relevant Project Task:	Task 3.4, Task 3.5
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS20 Blockchain-based Infrastructure
Owner / Leading partner:	CERTH

Supporting partners / Co-developers:	CERTH
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Middleware Layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	This system designs and develops monetization mechanisms alongside ID, authentication, and authorization systems for users and assets in the energy ecosystem. Solutions like PKIs, SSI, DIDs, VCs will be explored for this purpose. Additionally, they will create a service for green energy verification using Renewable Energy Certificates as NFTs, facilitating monetization opportunities within the ecosystem.
Brief technical description (100-200 words)	The Blockchain infrastructure focuses on implementing advanced identification, authentication, and authorization mechanisms within the blockchain infrastructure. It delves into both centralized (e.g., PKIs) and decentralized (e.g., SSI, DIDs, VCs) technical solutions. Moreover, it includes the development of a specialized service for green energy verification, utilizing Renewable Energy Certificates as minted NFTs. To ensure seamless operation, secure oracles will be integrated, facilitating the linkage between NFTs, Verifiable Credentials, and Decentralized Identifiers. These technical elements form the backbone for secure access control and green energy verification, ultimately enabling robust monetization opportunities within the ecosystem.

Inputs/ IS
Data regarding IPR information and NFTs metadata / IS22 (+ services layer broker)
Outputs/ IS
Data regarding IPR information and NFTs metadata / IS18, IS22,IS23

10.1.21 IS21 Data Transformation and Quality Assurance [CERTH]

Component/Service Name:	Data Quality Assurance and Filtering Mechanisms
Relevant Project Task:	T3.3
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS21
Owner / Leading partner:	CERTH
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Data Management Layer / Interoperability & Abstraction Services
Type:	Software Module / Implemented Functionality

Brief conceptual description (100-200 words)	This tool comprises a set of data anomalies detection models and mitigation rules, to ensure an adequate pre-processing of raw measured data, establishing a solid basis for more advanced (e.g., WP4) data-driven services. The first implementation step of this tool is to detect the outliers, using statistical methods. After an outlier is detected, it is removed from the original dataset and is mitigated. To mitigate missing values, both statistical and machine learning imputation mechanisms can be used. Statistical methods may include ARMA and ARIMA, linear and polynomial interpolation, as well as general regression, to predict the missing values. Machine learning mechanisms may include a wide range of algorithms used to predict missing values in a timeseries, such as Facebook Prophet.
Brief technical description (100-200 words)	Containerized solution in Python 3.8+ exploiting detecting and mitigating capacity of PANDAS library as well as unsupervised and supervised machine learning mechanisms for outlier elimination. The selection of simple or more advanced solutions will be affected by the different kinds of anomalies detected / observed in the measured data. A performance study will be conducted, to evaluate the predicted values by each forecasting method. For the evaluation of the specific tool, there will be created artificial outliers in the dataset. These outliers will be filtered and mitigated using the aforementioned methods and their performance will be evaluated by statistical methods, using as ground truth the original values from the dataset.
Inputs/ IS	

Structured and transformed numerical raw data / (IS20) IS23
Outputs/ IS
Filtered data streams, where the detected anomalies have been mitigated and can be used for advanced data analysis applications (i.e., WP4) / IS22

10.1.22 IS22 Data and Context Broker [CIRCE]

Component/Service Name:	Data Broker
Relevant Project Task:	T3.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS22 Data and Context Broker
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Data Management Layer / Middleware layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	Manages the flow of data between the northbound and southbound APIs.

Brief technical description (100-200 words)	It is a component that receives data from the southbound API, processes it, and sends it to the northbound API.
Inputs/ IS	
Secure (IS 20) data retrieval from field devices (JSON raw, variable, southbound API). / IS20	
Outputs/ IS	
Routed query data (JSON-LD, variable, northbound API). / IS23	

Component/Service Name:	Northbound Open API
Relevant Project Task:	T3.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS22 Data and Context Broker
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Data Management Layer / Middleware layer
Type:	Software Module / Implemented Functionality

Brief conceptual description (100-200 words)	Handles the delivery of data and services to the applications/service layer.
Brief technical description (100-200 words)	It is an API that provides a communication interface between the data broker and the applications/service layer.
Input Connections & Interfaces:	Receives processed data from the data broker.
Output Connections & Interfaces:	Sends data to the applications/service layer.

Component/Service Name:	Southbound Open API
Relevant Project Task:	T3.1
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS22 Data and Context Broker
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Data Management Layer / Middleware layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	Handles data ingestion from the underlying systems.

Brief technical description (100-200 words)	It is an API that provides a communication interface between the field layer and the data broker.
Input Connections & Interfaces:	Receives raw data from the field layer.
Output Connections & Interfaces:	Sends data to the data broker.

10.1.23 IS23 Common Information Modelling and Context Knowledge Repository [CIRCE]

Component/Service Name:	Interoperability Service (Data Conversion from JSON to SAREF Ontology Based JSON-LD)
Relevant Project Task:	T3.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS23 Common Information Modelling and Context Knowledge Repository
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module / Layer from the proposal (Figure 6, pg. 151 of GA)	Data Management Layer / Middleware layer
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	This component is responsible for converting the data to the SAREF ontology-based JSON-LD format.

Brief technical description (100-200 words)	It uses mapping rules to convert JSON data to SAREF ontology-based JSON-LD.
Indicative web sources / publications (if any)	N/A
Input Connections & Interfaces:	Receives secured data from the Blockchain-based Infrastructure component.
Output Connections & Interfaces:	Sends ontology-based data to the Abstraction Service component.
Inputs/ IS	
Secure raw unstructured data / (IS20) IS22	
Outputs/ IS	
Ontology-based data (JSON-LD, variable, Interoperability Service component)	

Component/Service Name:	Abstraction Service (Knowledge Graph, Query Engine)
Relevant Project Task:	T3.2
Relevant Project Innovative Solution (IS): (section 1.1.2, pg. 140 of GA)	IS23 Common Information Modelling and Context Knowledge Repository
Owner / Leading partner:	CIRCE
Supporting partners / Co-developers:	-
Architectural Module /	Data Management Layer / Middleware layer

Layer from the proposal (Figure 6, pg. 151 of GA)	
Type:	Software Module / Implemented Functionality
Brief conceptual description (100-200 words)	This component is responsible for storing the ontology-based data in a knowledge graph and providing a query engine.
Brief technical description (100-200 words)	It uses a graph database to store the ontology-based data and provides a SPARQL endpoint for querying the data.
Inputs/ IS	
Ontology-based data (JSON-LD, variable, Interoperability Service component).	
Outputs/ IS	
IS21	