MATRYCS

D4.4 | MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (1st technology release)

WP4 – Data Modelling & Open Modular Data Analytics Toolbox

August 2021

www.matrycs.eu

Modular Big Data Applications for Holistic
Energy Services in Buildings

MATRYCS

Disclaimer

Copyright Message

The MATRYCS project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no.101000158

Grant Agreement Number	101000158	Acronym	MATRYCS
Full Title	Modular Big Data Applications for Holistic Energy Services in Buildings		
Topic	LC-SC3-B4E-6-2020 | Big data for buildings		
Funding scheme	H2020- IA: Innovation Action		
Start Date	October 2020	Duration	36
Project URL	www.MATRYCS.eu		
Project Coordinator	ENG		
Deliverable	MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (1st technology release)		
Work Package	Data Modelling & Open Modular Data Analytics Toolbox		
Delivery Month (DoA)	M11	Version	1.0
Actual Delivery Date	31/08/2021		
Nature	Other	Dissemination Level	Public
Lead Beneficiary	NTUA		
Authors	Haris Doukas [NTUA], Vaggelis Marinakis [NTUA], Panagiotis Kapsalis [NTUA], Giorgos Korbakis [NTUA], Konstantinos Alexakis [NTUA], Zoi Mylona [HOLISTIC], Daniele Antonucci [EURAC], Manuel Mauro [EURAC], Jose Llamas [CARTIF], Ana Lopez Vidal [CARTIF], Gema Hernandez Moral [CARTIF], Leandro Lombardo [ENG], Marija Borisov [ENG], Nikola Dimitrijevic [ENG]		
Quality Reviewer(s):	Jaime Gómez Tribiño [VEOLIA], Marina Bartolomé Lorenzo [VEOLIA], Andrej Janko [BTC], Matej Bregar [BTC], Tomaž Damjan [BTC]		
Keywords	Data Analytics Toolbox, SaaS, IaaS, PaaS, Visualisation Engine, Building Analytics Services, Digital Building Twin, Integration & Testing plan		

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Preface

MATRYCS focuses on addressing emerging challenges in big data management for buildings with an **open holistic solution** for Business-to-Business platforms, able to give a competitive solution to stakeholders operating in building sector and to open new market opportunities. **MATRYCS Modular Toolbox**, will realise a holistic, state-of-the-art AI-empowered framework for decision-support models, data analytics and visualisations for Digital Building Twins and real-life applications aiming to have significant impact on the building sector and its lifecycle, as it will have the ability to be utilised in a wide range of use cases under different perspectives:

- Monitoring and improvement of the energy performance of buildings - **MATRYCS-PERFORMANCE**
- Design facilitation and development of building infrastructure - **MATRYCS-DESIGN**
- Policy making support and policy impact assessment - **MATRYCS-POLICY**
- De-risking of investments in energy efficiency - **MATRYCS-FUND**

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Who We Are

	Participant Name	Short Name	Country Code	Logo
1	ENGINEERING – INGEGNERIA INFORMATICA SPA	**ENG**	IT	
2	NATIONAL TECHNICAL UNIVERSITY OF ATHENS	**NTUA**	GR	
3	FUNDACION CARTIF	**CARTIF**	ES	
4	RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN	**RWTH**	DE	
5	ACCADEMIA EUROPEA DI BOLZANO	**EURAC**	IT	
6	HOLISTIC IKE	**HOLISTIC**	GR	
7	COMSENSUS, KOMUNIKACIJE IN SENZORIKA, DOO	**COMSENSUS**	SL	
8	BLAGOVNO TRGOVINSKI CENTER DD	**BTC**	SL	
9	PRZEDSIEBIORSTWO ROBOT ELEWACYJNYCHFASADA SP ZOO	**FASADA**	PL	
10	MIASTO GDYNIA	**GDYNIA**	PL	
11	COOPERNICO - COOPERATIVA DE DESENVOLVIMENTO SUSTENTAVEL CRL	**COOPERNICO**	PT	
12	ASM TERNI SPA	**ASM**	IT	
13	VEOLIA SERVICIOS LECAM SOCIEDAD ANONIMA UNIPERSONAL	**VEOLIA**	ES	
14	ICLEI EUROPEAN SECRETARIAT GMBH (ICLEI EUROPASEKRETARIAT GMBH)	**ICLEI**	DE	
15	ENTE PUBLICO REGIONAL DE LA ENERGIA DE CASTILLA Y LEON	**EREN**	ES	
16	VIDES INVESTICIJU FONDS SIA	**LEIF**	LV	
17	COMITE EUROPEEN DE COORDINATION DE L'HABITAT SOCIAL AISBL	**HOUSING EUROPE**	BE	
18	SEVEN, THE ENERGY EFFICIENCY CENTER Z.U.	**SEVEN**	CZ	

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Contents

1 Introduction 13

1.1 Purpose of this document 13
1.2 Structure of the document 13

2 General description of MATRYCS Toolbox 14

2.1 MATRYCS Toolbox Definition & Components 15
2.1.1 SaaS definition and application in MATRYCS 16
2.1.2 PaaS definition and application in MATRYCS 17
2.1.3 IaaS and application in MATRYCS 19

3 MATRYCS Toolbox Software as a Service Layer 22

3.1 Visualisation and Reports Engine 22
3.1.1 Components Architecture 23
3.1.2 Visualisation Engine Requests Examples 25
3.2 MATRYCS Analytics Building Services and integration with front-end environments 27
3.3 Digital Building Twins 28
3.3.1 Digital Building Twin – Building level 30
3.3.2 Digital Building Twin – District level 33
3.3.3 Digital Building Twin – Regional level 34

4 MATRYCS Toolbox Platform as a Service Layer 36

4.1 Building Data and Services Innovation Hub 36
4.2 Virtual Workbench - Data Analytics Environment 38

5 MATRYCS Toolbox Infrastructure as a Service Layer 41

5.1 Cloud Infrastructure and Service Orchestrator 41

6 Continuous Integration and Testing 42

6.1 Source Code Management 42
6.1.1 GitHub 43
6.1.2 Git Branching Methodology 43
6.1.3 Features and Repository branches 44
6.1.4 Code commits 45
6.1.5 Merge Requests 45
6.2 Automation Server 45
6.3 Toolbox Testing 49
6.3.1 Unit Testing 49
6.3.2 Integration Testing 49
6.3.3 User Acceptance Testing 52
6.3.4 Requirement Coverage 52
6.3.5 Error Tracking Application 54

7 MATRYCS Toolbox Integration on M11 59

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

8 Future Steps 60

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Figures

Figure 1: The MATRYCS toolbox layers 14
Figure 2: Services of Cloud Computing 15
Figure 3: MATRYCS Analytics Service Development 17
Figure 4: Virtual Workbench as PaaS in MATRYCS 18
Figure 5: Services provided On-premises, IaaS, PaaS and SaaS 20
Figure 6: Visualisation Engine Architecture 24
Figure 7: Pie Chart generated by the visualisation engine for Table 2 request 26
Figure 8: Bar Chart generated by the visualisation engine for Table 4 request 27
Figure 9: General Container infrastructure 28
Figure 10: Digital Building Twin scheme 29
Figure 11: Revit Tool Dynamo data flow 31
Figure 12: Examples of Revit/Dynamo scripts for integrating sensor data into BIM files 31
Figure 13: Xbim explorer sample interface 32
Figure 14: BIM vision sample interface 33
Figure 15: Data Sets list 37
Figure 16: Function catalogue 38
Figure 17: Example of cache data source configuration. 39
Figure 18: Example of HTML widget 39
Figure 19: Example of Table widget 40
Figure 20: GitFlow Workflow schema 44
Figure 21: Travis workflow example 46
Figure 22: Jenkins workflow example 47
Figure 23: Jenkins example with Docker 47
Figure 24: Jenkins's master-slave architecture 48
Figure 25: Top-Down approach for Integration Testing 50
Figure 26: Bottom-Up approach for Integration Testing 51
Figure 27: Hybrid approach for Integration Testing 51
Figure 28: Supported Frameworks and Languages 55

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Figure 29: Example of Error after integration testing with Sentry 56

Figure 30: Installation of Sentry in Python project 57

Figure 31: Redmine's Gantt diagram for issues and bugs 58

Figure 32: MATRYCS Toolbox Components Interconnections 59

Tables

Table 1: Visualisation Engine Docker Compose 24

Table 2: JSON Payload for Requested data 25

Table 3: Backend JSON Response 25

Table 4: JSON Request from Backend 26

Table 5: *JSON* Response 26

Table 6: Jenkins and Travis comparison 48

Table 7: Advantages & Disadvantages of the Top-Down Approach 50

Table 8: Advantages & Disadvantages of Bottom-UP approach 51

Table 9: List of MATRYCS Technical Requirements 52

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Abbreviations and Acronyms

Acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
ANN	Artificial Neural Networks
AWS	Amazon Web Services
BAC	Building Automation Control
CAD	Computer Aided Design
CI/CD	Continuous Integration / Continuous Testing
CPU	Central Processing Unit
CSP	Cloud Service Provider
CSS	Cascading Style Sheets
CSV	Comma-Separated values
DAG	Directed Acyclic Graphs
DB	Database
DBT	Digital Building Twin
DT	Digital Twin
DL	Deep Learning
DLT	Distributed Ledger Technology
ECM	Energy Conservation Measure
EPC	Energy Performance Contracting
EU	European Union
GUI	Graphical User Interface
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
IoT	Internet of Things
IFC	International Finance Coorporation
JS	JavaScript
JSON	JavaScript Object notation
LSP	Large Scale Pilot
ML	Machine Learning
MPA	Multi-Page Application
OSM	Open Street Map
PaaS	Platform as a Service
RAM	Random Access Memory
REST	Representational state transfer
SaaS	Software as a Service
SCM	Source Code Management
SDK	Software Development Kit
SME	Small-Medium size Enterprises
SPA	Single Page Application
SQL	Structured Query Language
STEP	Systemic Test and Evaluation Process
UI	User Interface
VC	Version Control
VM	Virtual Machine

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Executive Summary

The D4.4 - MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (1st technology release) provides a description of the first version of MATRYCS Toolbox according with the MATRYCS high level reference architecture defined in the deliverable D2.3 - *MATRYCS Reference Architecture for Buildings Data v1.0*.

The 1st technology release reported in this deliverable is mainly focused on the preliminary evaluation of the envisaged technologies solutions for the MATRYCS Toolbox and the description of the different layers (SaaS, PaaS, IaaS) as well as their main building blocks. The 1st technology release also focuses on the continuous integration and testing plan (mechanisms, tools, etc.) for the MATRYCS Toolbox. To this end, the tasks that have been performed until M11 in MATRYCS Toolbox are T4.4 - *Advanced Visualisations & Reports Engine*, T4.5 - *MATRYCS Virtual Workbench*, T4.6 – *Continuous Integration and Testing*.

Furthermore, the integration activities until M11 are reported focused on planning the connection of the SaaS components (Advanced Visualisation Engine, Digital Building Twin, Building Analytics Services) along with PaaS layer (MATRYCS Workbench). Specifically, the output of tasks T5.1 - *MATRYCS Digital Building Twin* and T5.7 – *MATRYCS Analytics Services Integration* are reported as part of the Data Analytics Toolbox.

The MATRYCS project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no.101000158

1 Introduction

1.1 Purpose of this document

The purpose of D4.4 - *MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (1st technology release)* is to report the implementation of the first technology release of Big Data Analytics Toolbox for Buildings (MATRYCS-ANALYTICS). In this regard, this deliverable reports the activities and outcome carried out in T4.4 - *Advanced Visualisation & Reports Engine*, T4.5 - *MATRYCS Virtual Workbench*, T4.6 - *Continuous Integration & Testing*, and the integration activities of SaaS layer with T5.1 - *MATRYCS Digital Building Twins and Test-beds* and T5.7 - *Integration of building services analytics towards pilot deployment*.

The work done so far is focused on the preliminary evaluation of the envisaged technologies for the MATRYCS-ANALYTICS layer, highlighting conceptual architectures of the different components, the interaction between them and the implementation aspects.

The results of the MATRYCS-ANALYTICS services, which are the outputs of the T5.2 - *Analytics for Energy Performance – Indoor Condition Evaluation and Intelligent Energy Management*, T5.3 - *Analytics for Buildings and related infrastructure*, T5.4 - *Analytics for Policy Making and Policy Impact Assessment*, T5.5 - *Analytics for De-Risking Investments in Energy Efficiency* and T5.6 - *Geoclustering Services as support to the BD vision* will be reported in D5.1 - *MATRYCS Analytics Building Services v1.0 (October 2021)*.

1.2 Structure of the document

The D4.4 - *MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (1st technology release)* is organized as follows:

- In section 1, the purpose of the document and related structure is presented.
- In section 2, an overview of the MATRYCS Toolbox is presented.
- In section 3, MATRYCS SaaS is presented with a general description of building blocks (Advanced Visualisations & Reports Engine, Digital Building Twin, MATRYCS Analytics and integration with Frontend environments).
- In section 4, MATRYCS PaaS is presented focusing on the implementation of MATRYCS Virtual Workbench which is the main component of PaaS layer.
- In section 5, MATRYCS IaaS and its main components are presented.
- In section 6, continuous Integration & Testing plans are demonstrated.
- In section 7, MATRYCS Toolbox Integration until M11 is demonstrated.
- In section 8, conclusions and next steps are outlined.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

2 General description of MATRYCS Toolbox

MATRYCS aims to create a unified framework in the field of Big Data management for buildings, aspiring to be an open-source solution for different actors working in the building domain. To reach this aim, MATRYCS will deploy a Reference Architecture for Building Data exchange, management and real-time processing in the form of an Open, Cloud-based Data Analytics Toolbox, the MATRYCS Modular Toolbox (Figure 1).

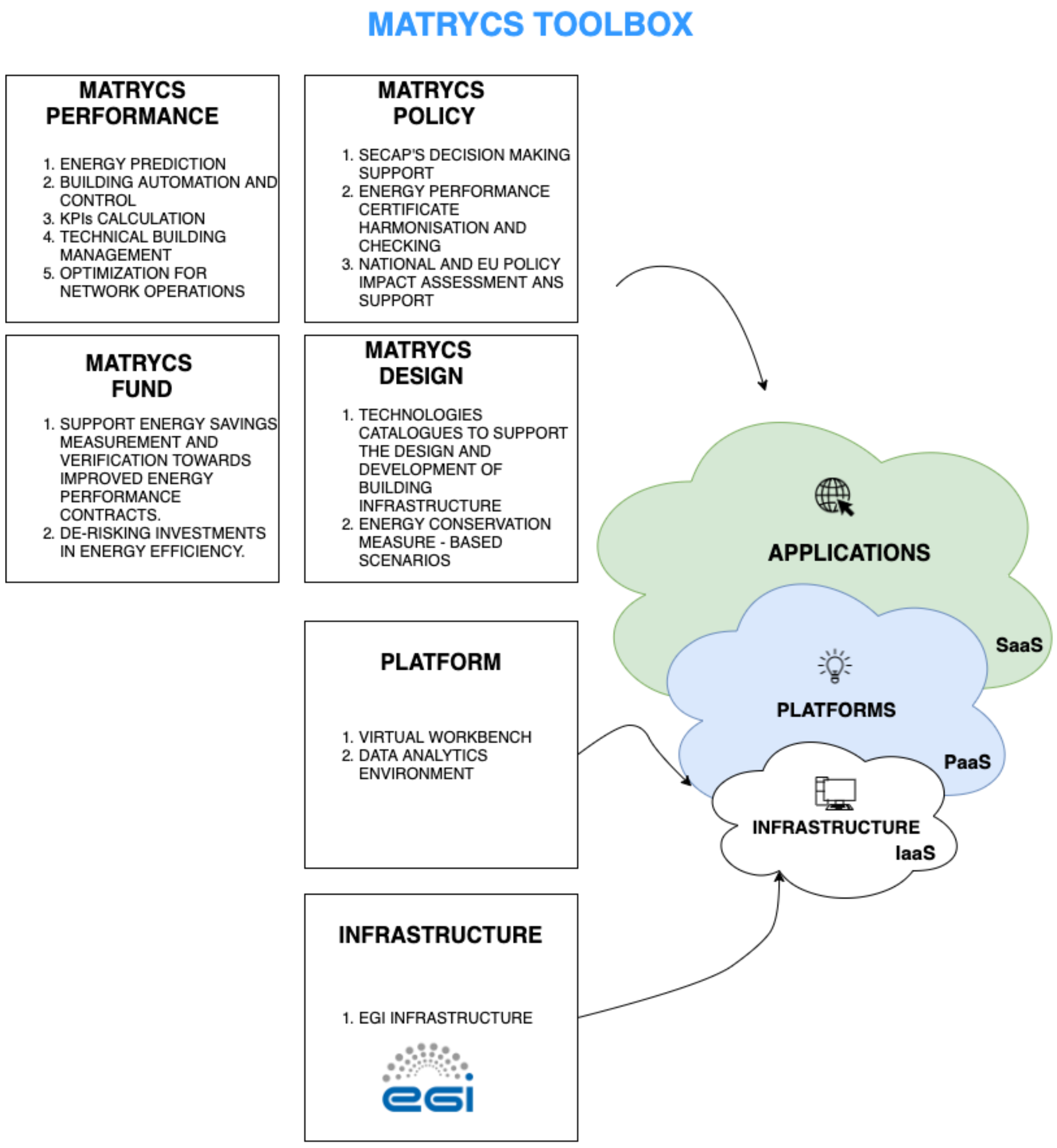

MATRYCS TOOLBOX
MATRYCS PERFORMANCE
1. ENERGY PREDICTION
2. BUILDING AUTOMATION AND CONTROL
3. KPIs CALCULATION
4. TECHNICAL BUILDING MANAGEMENT
5. OPTIMIZATION FOR NETWORK OPERATIONS
MATRYCS POLICY
1. SECAP'S DECISION MAKING SUPPORT
2. ENERGY PERFORMANCE CERTIFICATE HARMONISATION AND CHECKING
3. NATIONAL AND EU POLICY IMPACT ASSESSMENT ANS SUPPORT
MATRYCS FUND
1. SUPPORT ENERGY SAVINGS MEASUREMENT AND VERIFICATION TOWARDS IMPROVED ENERGY PERFORMANCE CONTRACTS.
2. DE-RISKING INVESTMENTS IN ENERGY EFFICIENCY.
MATRYCS DESIGN
1. TECHNOLOGIES CATALOGUES TO SUPPORT THE DESIGN AND DEVELOPMENT OF BUILDING INFRASTRUCTURE
2. ENERGY CONSERVATION MEASURE - BASED SCENARIOS
PLATFORM
1. VIRTUAL WORKBENCH
2. DATA ANALYTICS ENVIRONMENT
INFRASTRUCTURE
1. EGI INFRASTRUCTURE
egi
APPLICATIONS
SaaS
PLATFORMS
PaaS
INFRASTRUCTURE
IaaS

Figure 1: The MATRYCS toolbox layers

The **Software as a Service (SaaS)** will develop services based on the inputs and requests from the LSPs. The services will cover aspects related to building indoor condition and energy production and consumption, building infrastructure, policy assessment and building efficiency investments.

The **Platform as a Service (PaaS)** will offer a "virtual workbench" to incorporate a variety of assets, including data, third party services, ML models, computing resources, storage resources as tradable

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

assets. It will provide a set of tools targeting SMEs, developers and potential innovators, who design and develop new services for the buildings sector.

At **Infrastructure as a Service (IaaS)** level, the platform will package its services, as an adapted open - source sandbox, in container-based workflows that could be flexibly deployed over cloud infrastructures through a service orchestrator. The pre-existing available local-level capabilities for sensing, gathering, integrating and processing data (e.g. IoT devices) through available in-house analytics platforms can be augmented with the capabilities of the proposed analytics reference toolbox.

In the following sections the three services (SaaS, PaaS, IaaS) will be discussed in more detail.

2.1 MATRYCS Toolbox Definition & Components

The realization that cloud services are capable of providing a major boost to business has made Cloud Computing one of the pillars of technological innovation in recent years. Cloud computing is the technology that allows taking advantage, via remote server, of software and hardware resources and the use of which is offered as a service by a provider.

With the ever-increasing availability of data from buildings, the use of Cloud Computing with the sure resources and proper organization of the data and associated services is crucial.

In this regard, the MATRYCS Toolbox could be seen as a "container" that aims to provide three main services of Cloud Computing on the building domain.

These services are:

1. Software as a Service (SaaS)
2. Platform as a Service (PaaS)
3. Infrastructure as a Service (IaaS)

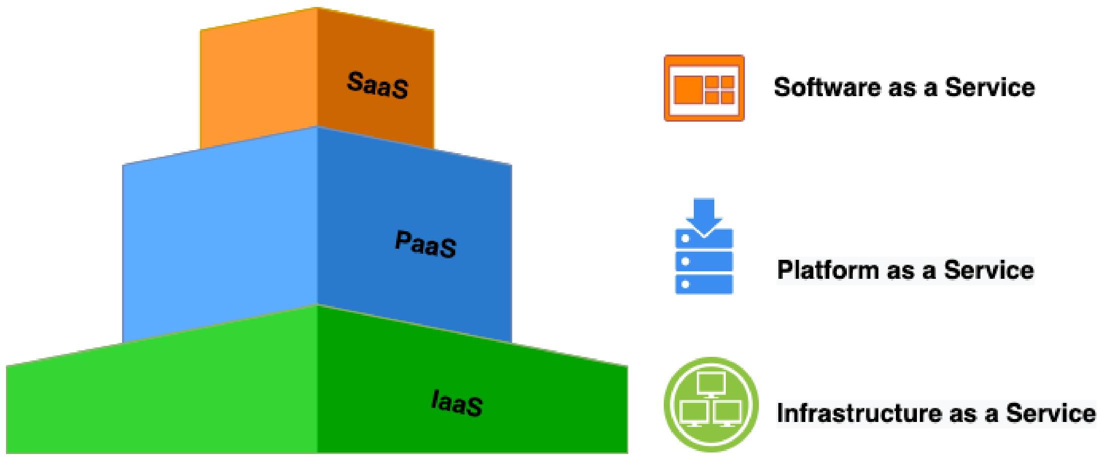


Figure 2: Services of Cloud Computing

From a conceptual point of view, IaaS, PaaS, and SaaS can be considered as three levels of abstraction (Figure 2), diversified according to the type of service offered and the type of user. Thus, it is possible to relate these models. *A service offered by an IaaS supports a PaaS service, which in turn supports a SaaS service*.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

2.1.1 SaaS definition and application in MATRYCS

SaaS is a model that encompasses applications and software systems, accessible from any type of device (computer, smartphone, tablet, etc.), through the simple use of a client interface. In this way, the user does not have to worry about managing the resources and infrastructure, as they are controlled by the provider.

According to G. Kulkarni [1], SaaS is one of the methodologies of Cloud Computing, which is based on a "one-to-many" model whereby an application is shared across multiple clients.

This means that a single instance of the software is available to serve multiple customers. Therefore, all users and applications share a common infrastructure and code base that is maintained centrally [2]. In this case instead of updating every single instance, the code base is updated only one time and instantly accessible to all users.

Much like any other software, SaaS can also take advantage of the Service Oriented Architecture to allow software applications to communicate with each other. SaaS can work both as service provider and service request, collecting data and using functionalities from other services.

SaaS is a very advantageous cloud solution, it substantially lowers costs in several aspects, such as the purchase of software licensees and hardware investment. Thanks to Cloud Computing models, hosting by providers of this service, enables low-cost integration. Furthermore, multi-tenancy also reduces upfront cost because of the shared infrastructure as well as low maintenance costs.

SaaS responds well to the demand for scalability of resources as services/users grow. Indeed, it is easily upscale or downscale the resources when a business's requirements change. This means that there is no need to purchase more hardware if required, but expand only computing capacity such as data storage, bandwidth, RAM, CPU, etc.

Using SaaS also has disadvantages compared to an "on-premises" application, such as:

- There is a strong dependency with the service provider. If there is a drop in connection with the cloud provider or if it decides to discontinue the services, the application does not work anymore.
- Sometimes the performance of a on-premises application could be better than on cloud. This may be due to the speed and reliability of internet connection.
- Using on-premises applications means that everything is managed and controlled by the user, whereas in the SaaS, the third party has the control of the overall process.

  In MATRYCS different analytics services for buildings will be developed using both SaaS and on-premises technologies.

The MATRYCS analytics will expose services to multiple stakeholders.

At the SaaS level the MATRYCS toolbox will incorporate:

- A **Visualisations and Reports Engine**, responsible for the visual representation of the stored data and the results produced from the analytical components. It will offer a variety of visual representations including charts and map visualisations, based on specific KPIs developed in the project.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

- A range of innovative **Analytics Building Services** will be provided, such as:
  - Analytics for energy performance - indoor condition evaluation and intelligent energy management.
  - Analytics for building systems and infrastructure.
  - Analytics for policy making and policy impact assessment on building level.
  - Analytics for building efficiency investments.

These Analytics Building Services, combined with the Digital Building Twins, will be exploited in order to apply holistic energy services. These are categorised in four main groups, covering energy performance and indoor condition evaluation of buildings (MATRYCS-PERFORMANCE), building infrastructure (MATRYCS-DESIGN), policy making and policy impact assessment (MATRYCS-POLICY) and building efficiency investments (MATRYCS-FUND).

The MATRYCS Digital Building Twins include all the new end-user applications that will be developed or existing that will be extended by exploiting the developed and trained models served by the Artificial Intelligence (AI) Services Layer. Based on the concept of Digital Building Twins, the developed applications will create digital representations of physical systems like buildings, energy systems and more, in order to perform analytics, simulations, examine multiple scenarios, run tests, make predictions and estimations to the digital space and generate actionable insights and optimization recommendations for a number of use cases at the physical world based on custom criteria.

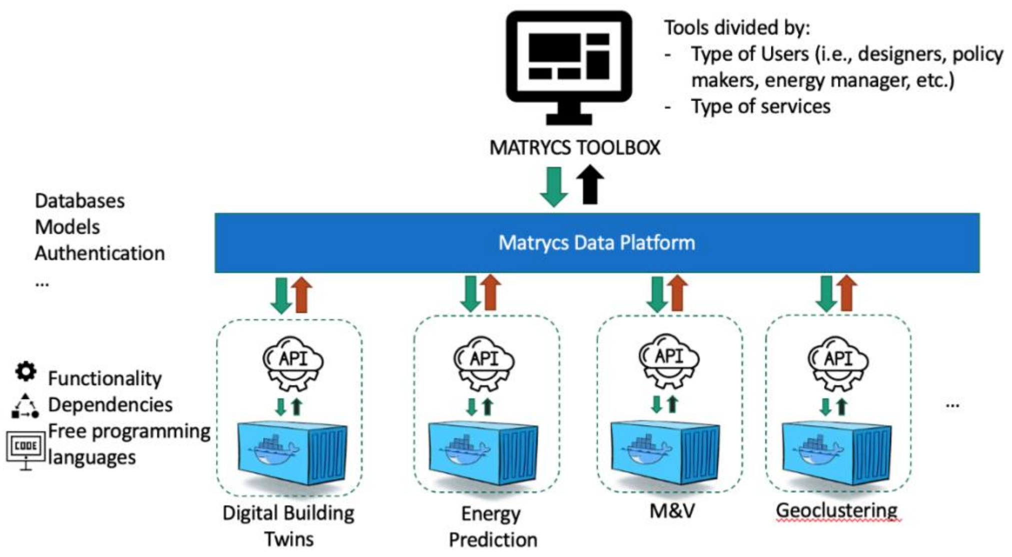


Figure 3: MATRYCS Analytics Service Development

The figure above (Figure 3) demonstrates how each single application will be developed and integrated in the platform where the user can access and select the most useful service for its application. The latter will have a dedicated graphical interface (GUI).

2.1.2 PaaS definition and application in MATRYCS

PaaS is a model in which online platform services are located. Thanks to it a user, usually a developer, can develop and run his own applications through the tools provided by the provider, who guarantees

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

the proper functioning of the underlying infrastructure.

Unlike a SaaS platform, the PaaS platform is accessible to developers to create software. PaaS environments offer developers the opportunity to focus on software and application development without worrying about operating systems, software updates, storage or infrastructure.

Developers can customize apps without the fear of maintaining the software, while a reduced amount of coding is required. In addition, the PaaS model enables automation of business policies and can simplify workflows when multiple developers are working on the same project.

In the MATRYCS project the PaaS will be the Virtual Workbench (4). The latter provides to the user an intuitive and easy GUI to create new services in the public sector.

It will integrate two modules:

- **Building Data & Services Innovation Hub**: it will provide SMEs, developers and potential innovators a GUI that will facilitate the development of new applications on the base of the existing data sets and the available set of the developed and trained models being produced by MATRYCS.
- **Data Analytics Environment**: it will allow users (e.g. analysts) to perform freeform queries and data analytics on the assets which are accessible to the platform (models and data).

Moreover, it will incorporate a variety of MATRYCS assets and tools including:

- **Data** (MATRYCS LSPs datasets)
- **ML/DL models** (Trained Models Library)
- **Third party services**

The Figure below (Figure 4) demonstrates the interaction of the virtual workbench with the different components of the SaaS and databases.

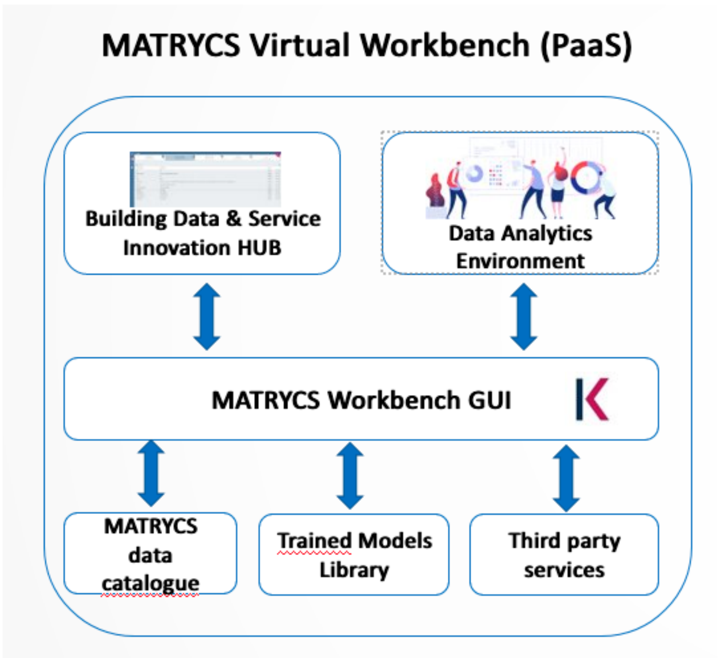

MATRYCS Virtual Workbench (PaaS)
Building Data & Service Innovation HUB
Data Analytics Environment
MATRYCS Workbench GUI
MATRYCS data catalogue
Trained Models Library
Third party services

Figure 4: Virtual Workbench as PaaS in MATRYCS

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

2.1.3 IaaS and application in MATRYCS

IaaS is a model in which virtualized hardware resources are made available so that users can create and manage their own infrastructure on the cloud according to their needs, without worrying about where the resources are allocated.

These cloud services are accessed via dashboards or APIs, giving IaaS customers complete control over the entire infrastructure. In effect, an IaaS platform provides technology and functionality identical to a conventional on-premises data center without the need to physically manage it. Some of the most important providers of IaaS in the market are: Amazon Elastic Cloud Compute (EC2), Amazon Simple Storage Service (S3), Amazon Virtual Private Cloud (VPC), Google Cloud Engine, Google Cloud Storage, etc.

An IaaS platform has benefits in many ways. It helps companies scale their infrastructure up or down, depending on the needs of change.

Organizations purchase IaaS resources on an as-needed basis, for as long as they need them, without long procurement and deployment cycles. In addition, hosting the required hardware and services in the cloud reduces the work required to host and maintain these services on-premises.

Compared to other delivery models, the IaaS platform offers the highest level of control over software and hardware.

One of the main concerns of this technology is the IT security. Developers strive to innovate faster and faster, deploying more VMs, storage, etc.

At this stage of the project, the IaaS in MATRYCS has not yet been defined. Different options are under evaluation, but the use of EGI[1] infrastructure is the first choice. EGI is a federation of hundreds of data and compute centers worldwide and tens of cloud providers united by a mission to support research activities.

EGI provides both technical and human services, from integrated and secure distributed high-throughput computing and Cloud Computing; storage and data resources to consultancy; support and co-development.

The benefits to use the EGI are:

- Easy discoverability and access to the services.
- Access to the services provided through various service specific workflows.
- Efficient sharing of resources and provision or e-infrastructure services to researchers.
- Facilitates inter-disciplinary research by providing access technologies typically considered outside of a particular field.
- Collaborative improvement of services and resources.
- Allows researchers and institutions to focus on value creation as opposed to maintaining redundant resources.

The following graph (Figure 5) summarizes the services that each component (IaaS, PaaS, and SaaS) can provide together with the "on-premises" application:

[1] https://www.egi.eu/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

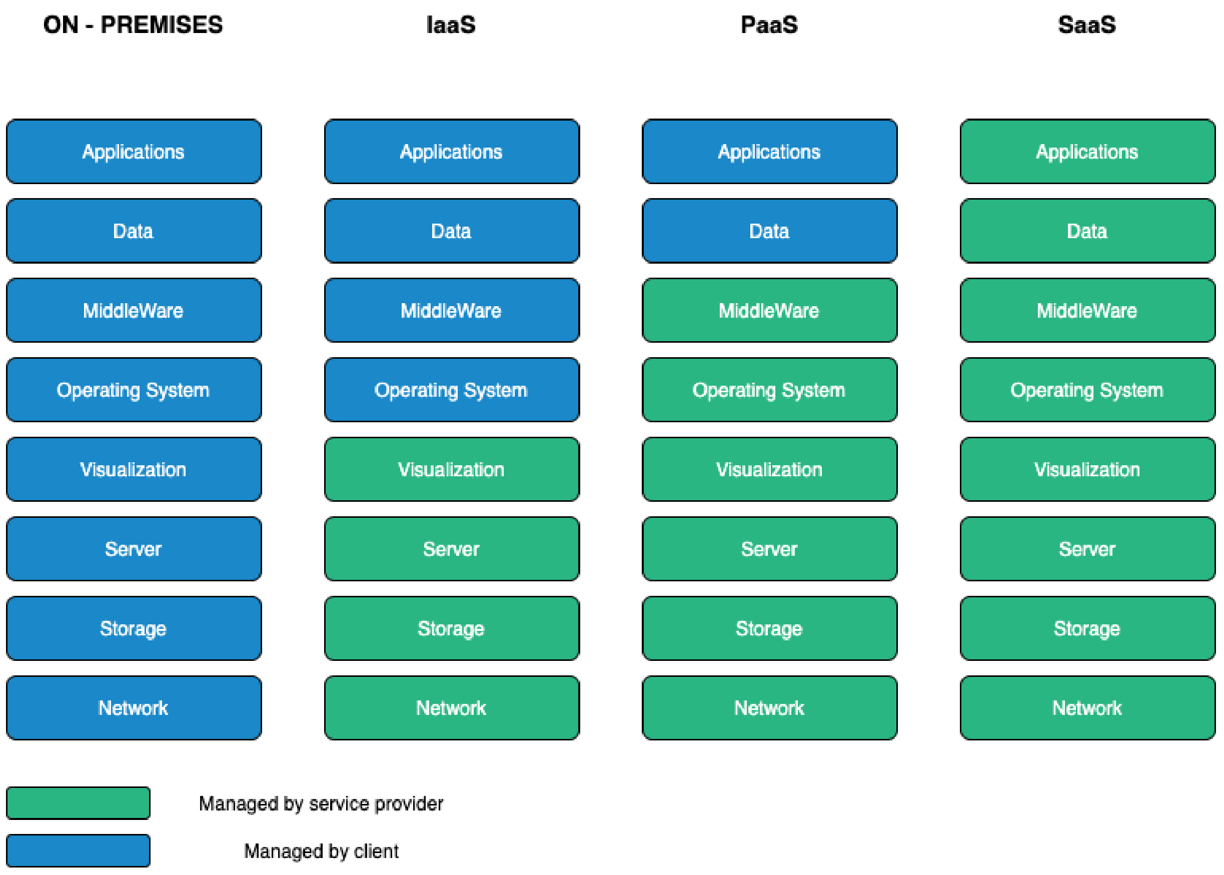


Figure 5: Services provided On-premises, IaaS, PaaS and SaaS

Therefore, on the on-premises application, all services are directly managed by the user, whereas with the SaaS everything is managed by the provider of the software solution.

One of the points to consider using the IaaS, PaaS and SaaS in a cloud is the security. Attacks are highly likely if everything is launched online. This is probably the most challenging part in Cloud Computing. IaaS, PaaS and SaaS have increase security concerns compared to other platforms because of its consistent interaction with different users.

Indeed, with traditional on-premises data centers, data and infrastructure security were the unique responsibility of the internal security team. Clearly defined perimeters make the application of security controls a relatively simple process.

By migrating to the cloud, these boundaries are blurred and security teams have to deal with infrastructure and services beyond traditional perimeter security, increasing the difficulty in defining a robust security strategy.

In short, the Cloud Service Provider (CSP) is responsible for the security of the cloud, but the customer is responsible for protecting his or her own data within cloud environments.

In an IaaS platform, the CSP is responsible for protecting data centers and other hardware components that support the infrastructure, including networks, storage disks and VMs.

IaaS customers need to protect their data, operating systems and software stacks running their applications. The PaaS model places more responsibility in the hands of the platform providers, but it is the customer's responsibility to protect their applications and associated data.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

In the MATRYCS project, in order to establish user authentication and authorization, to secure the non-open data of the transactions as well as to comply with the EC regulations on Data Protection (GDPR), and finally for logging user actions and system events, a security layer will be developed covering all components of the MATRYCS architecture, in the sense that it spans and interacts with several building blocks of the latter. The security framework used to protect data of the pilots is described in the D3.2 - *End-to-End Security Framework*.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

3 MATRYCS Toolbox Software as a Service Layer

MATRYCS' aim is to capitalise and harmonically combine existing modern technological breakthroughs in the areas of the DLTs, blockchain, ML, DL and Big Data, in order to develop a new, innovative decision-making and data analytics solution which will lead to energy-efficient buildings. To this end, MATRYCS will create a holistic and state-of-the-art AI-empowered framework for decision-support models, data analytics and advanced visualisations for Digital Building Twins and real-life applications. Thus, a significant impact is expected on the building sector and its lifecycle, due to the resulting MATRYCS Modular Toolbox, which will have the ability to be utilized in a wide range of use cases, under various perspectives. The MATRYCS Toolbox's capabilities will include monitoring and improving buildings' energy performance, facilitating the design and development of buildings' infrastructure, supporting policy making and policy impact assessment and de-risking investments in energy efficiency.

3.1 Visualisation and Reports Engine

As Task 4.4 - *Advanced & Visualisation Engine* includes all the activities performed to create, the advanced visualisations and reports engine created for the MATRYCS project offers a variety of capabilities, including creating advanced visualisations and reports, as well as generating maps in order to depict the fetched data.

The Visualisation & Reports Engine's role is to create advanced visualisations and reports on flexible dashboards over stored data that will be offered to the end users. It will offer a variety of visual representations including charts and map visualisations able to be customized upon the different needs. Moreover, direct, intuitive and informative feedback for the analysed power consumption data in a building can be provided using a visualisation engine [3].

Following an architecture based on microservices and by leveraging the functionalities of frontend frameworks and libraries, such as React.js[2], Material-UI[3], amCharts.js[4] and Leaflet.js[5], the Visualisation Engine is a user-friendly environment that offers a flexible, fully extendable, reusable and highly customizable dashboard to the end users. They will be able to choose the desired data between the available datasets, as well as how they want to visualise them, choosing between a variety of available charts/diagrams and maps.

All the functionalities have been encapsulated into a dashboard through which the user is able to query the available data using properly designed forms or directly execute SQL queries to the database. A variety of charts is offered (Pie Charts, Gauge Charts, Column Charts, Line Charts, etc), as well as maps and adaptable tables, with the vision to serve the needs of each case and each user. Additionally, the visualisations are also served as in an embeddable format, extending, in this way, the usability of the visualisation engine.

The under-development dashboard's functionalities are expected to be extended, as the project progresses. These functionalities include -but are not limited to- the implementation of further and more

[2] https://reactjs.org/

[3] https://material-ui.com/

[4] https://www.amcharts.com/

[5] https://leafletjs.com/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

complex visualisations, as well as a user registration functionality, which will offer the capability of storing data and queries into user's personal profile.

3.1.1 Components Architecture

The dashboard created for Task 4.4 - *Advanced & Visualisation Engine*, is Single Page Application (SPA) which is developed using the React JS framework. SPAs are a state-of-the-art approach to web development and offer several advantages compared to Multi-Page Applications. Specifically, SPAs are faster than Multi-Page Applications since most resources (HTML/CSS/Scripts) are only loaded once throughout the lifespan of an application and from that point on they do not update the entire page but only the required content. SPAs also offer better caching, easier debugging and in general a better user experience.

The dashboard is designed and implemented with user-friendliness in mind, meaning that it should be simple, legible and fully responsive, with an interface that adjusts perfectly in both desktop and mobile devices.

The CSS framework used for the creation of the dashboard is the Material-UI library, one of most popular libraries used with React. It has been created by Google and can be used as a single resource for all the styling needs, since it provides components, styles, themes and icons.

The visualisations are developed using the amCharts library, which is a library for data visualisation that includes basic and advanced charts, as well as various extendable plugins and other functionalities. Finally, the map visualisations will be developed using the leaflet.js library, which is an open-source JavaScript library for creating interactive maps. Leaflet.js library t has a variety of mapping features, and it is designed with simplicity, performance and usability in mind.

When using the dashboard, the user will be able to choose between a variety of visualisations, which are displayed in a sidebar in the left of the screen. By filling a properly designed HTML form, the user will be capable of choosing between the available datasets and the specific columns that need to be visualised. The form will be able to support common SQL commands (ORDER BY, LIMIT, DESC/ASC), which will then be depicted in the corresponding chart/diagram. After submitting a valid form, the frontend, that is a ReactJS based SPA, connects to the backend and sends a REST API call using the Axios[6] library, with the form's data as payload, in JSON format. The backend retrieves the requested data, proceed with properly formatting, and feeds them back to the frontend, again in JSON format, where they are processed and injected into the requested by the user visualisation chart.

Other than forms, the user will also be able to directly execute SQL queries to the database by writing plain SQL queries in a textbox the platform offers. After submitting a valid query, the requested data are fetched following the process described above and are displayed in fully responsive tables, offering a clean and legible view of the data.

The architecture described above, is depicted in Figure 6:

[6] https://www.npmjs.com/package/axios

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

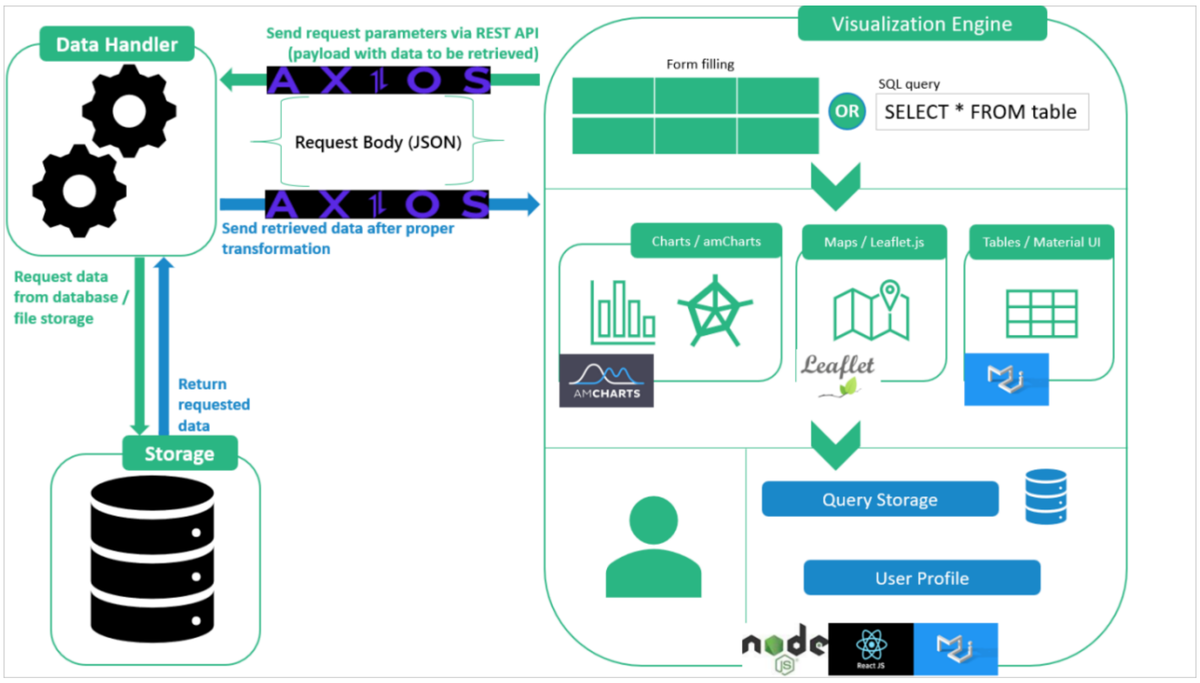


Figure 6: Visualisation Engine Architecture

The ReactJS application is installed using Docker[7] and docker-compose scripts. The docker-compose.yaml file (Table 1) that is used for the module's installation is attached below:

Table 1: Visualisation Engine Docker Compose

```
version: '3'
services:
  nodejs-server:
    build:
      context: ./api
    ports:
      - "3080:3080"
    container_name: node-api
    volumes:
       - ./api:/usr/src/app/api
       - /usr/src/app/api/node_modules
  react-ui:
    build:
      context: ./my-app
    ports:
      - "3000:3000"
    container_name: react-ui
```

[7] https://www.docker.com/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

```
stdin_open: true
volumes:
  - ./my-app:/usr/src/app/my-app
  - /usr/src/app/my-app/node_modules
```

3.1.2 Visualisation Engine Requests Examples

In this section, some indicative requests, responses and visualisation engine outputs will be displayed.

As mentioned in the previous sub-sections, the visualisation engine has been created following a microservices approach. Thus, the backend and the frontend are two separate applications, properly connected in order to harmonically collaborate and provide the desired output. The communication between the backend and the frontend is achieved through JSON requests/responses which are sent/received by using the Axios library in the frontend application.

Below (Table 2) an example of the payload sent to the backend in a JSON form when the user requests data for a chart is presented:

Table 2: JSON Payload for Requested data

```
{
  "aggregation_column": "co2_emission_ratio"
  "aggregation_metric": "SUM"
  "aggregation_metric_alias": "co2_emission_ratio"
  "grouping_columns": ["municipality"]
  "limit": "10"
  "table": "eren_building"
}
```

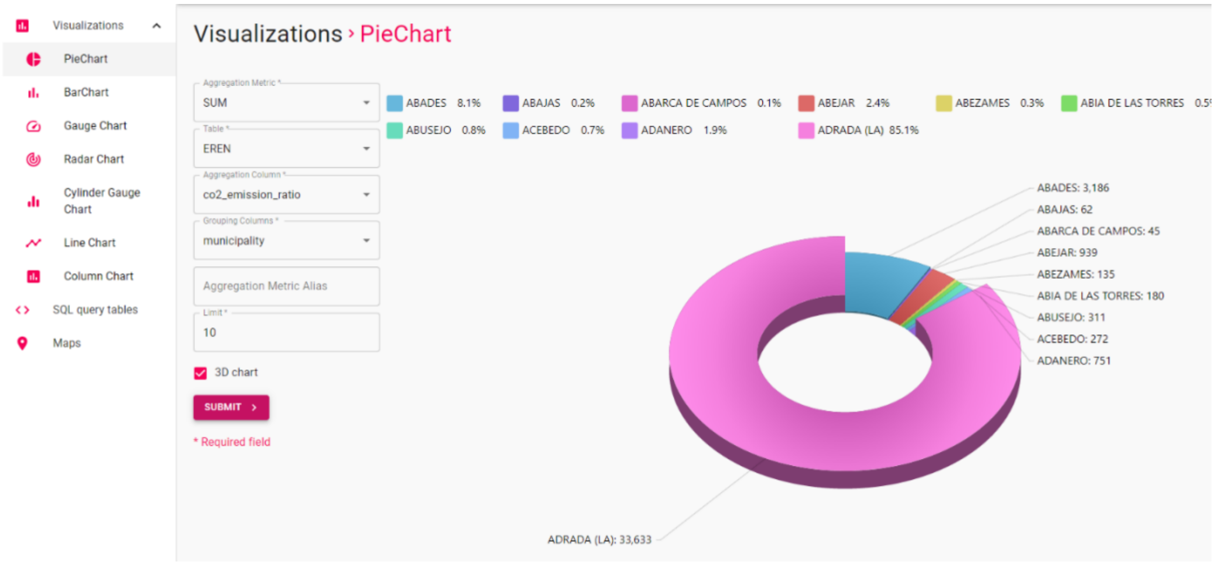
Table 3 presents the returned data from the query above Table 2:

Table 3: Backend JSON Response

```
[
  {co2_emission_ratio: 3186, municipality: "ABADES"}
  {co2_emission_ratio: 62, municipality: "ABAJAS"}
  {co2_emission_ratio: 45, municipality: "ABARCA DE CAMPOS"}
  {co2_emission_ratio: 939, municipality: "ABEJAR"}
  {co2_emission_ratio: 135, municipality: "ABEZAMES"}
  {co2_emission_ratio: 180, municipality: "ABIA DE LAS TORRES"}
  {co2_emission_ratio: 311, municipality: "ABUSEJO"}
  {co2_emission_ratio: 272, municipality: "ACEBEDO"}
  {co2_emission_ratio: 751, municipality: "ADANERO"}
  {co2_emission_ratio: 33633, municipality: "ADRADA (LA)"}
]
```

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Figure 7 presents the Pie Chart that is formed with the returned data.



Visualizations › PieChart

Aggregation Metric *: SUM
Table *: EREN
Aggregation Column *: co2_emission_ratio
Grouping Columns *: municipality
Aggregation Metric Alias
Limit *: 10
☑ 3D chart

* Required field

Figure 7: Pie Chart generated by the visualisation engine for Table 2 request

Another example is presented in Table 4. The procedure is initiated with the payload sent to the backend in a JSON form when the user requests data for a chart:

Table 4: JSON Request from Backend

```
{
    aggregation_column: "*"
    aggregation_metric: "COUNT"
    aggregation_metric_alias: "total"
    grouping_columns: ["co2_emissions_rating"]
    limit: "0"
   table: "eren_building"
}
```

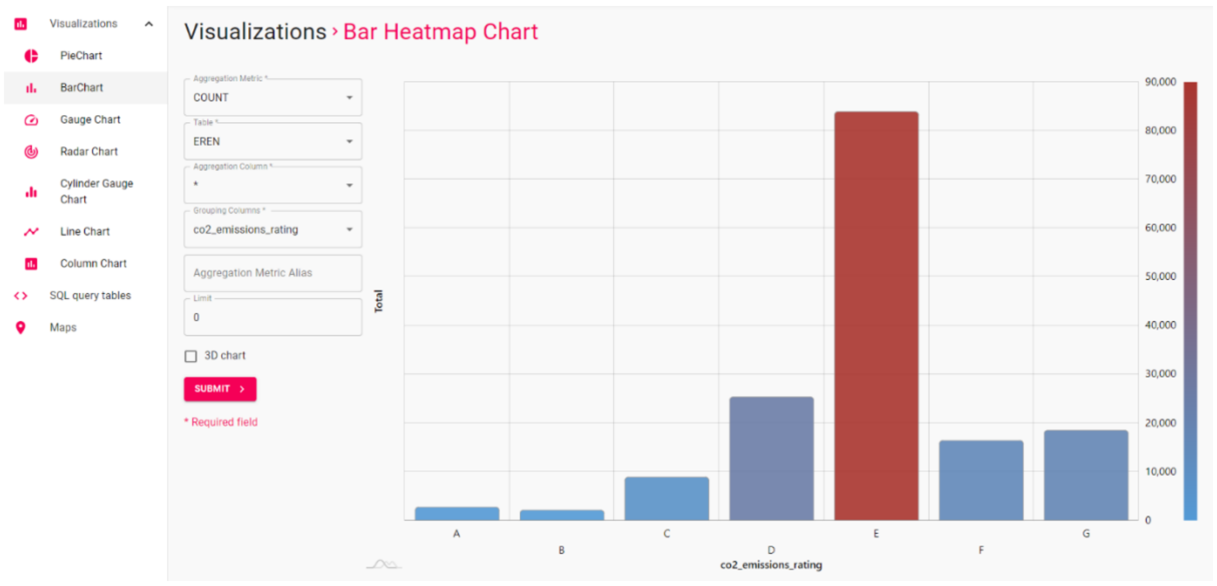
Table 5 presents the response of the returned data from the query above Table 4:

Table 5: *JSON* Response

```
[
    {total: 2699, co2_emissions_rating: "A"}
    {total: 2080, co2_emissions_rating: "B"}
    {total: 8871, co2_emissions_rating: "C"}
    {total: 25328, co2_emissions_rating: "D"}
    {total: 83899, co2_emissions_rating: "E"}
    {total: 16379, co2_emissions_rating: "F"}
    {total: 18482, co2_emissions_rating: "G"}
]
```

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Following Figure 8 depicts the bar charts that is formed with the returned data:



Aggregation Metric *: COUNT
Table *: EREN
Aggregation Column *: *
Grouping Columns *: co2_emissions_rating
Aggregation Metric Alias
Limit: 0
3D chart
* Required field

Figure 8: Bar Chart generated by the visualisation engine for Table 4 request

3.2 MATRYCS Analytics Building Services and integration with front-end environments

Several services will be developed within MATRYCS. These analytical building services will answer the needs specified in the pilots and has been captured in the user stories and specification (D2.1 – *State of the Art Analysis and Big Data Value Chain*). Here under the services list identified through the Name and the MATRYCS code (s = service + number) according to GA:

- Digital Twin (s.0.1)
- Geoclustering services as support to the Big Data vision [EURAC] (s.0.2)
- Energy Prediction (s.1.1)
- Building automation and Control (BAC) (s.1.2)
- KPIs Calculation (s. 1.3)
- Technical Building Management (TBM) (s1.4)
- Optimization for network operations (s1.5)
- Technologies catalogues services to support the design and development of building infrastructure (s2.1)
- ECM-based scenarios evaluation service (s.2.2)
- Support SECAPs decision-making (s.3.1)
- Support Energy Performance Certificate (EPC) harmonisation and checking (s.3.2)
- Services to support national and EU policy impacts assessment (s3.3)

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

- Services to support Energy Savings M&V towards improved Energy Performance Contracts (s.4.1)
- Services to support the financing of EU refurbishments (s.4.2)

The services will be developed to run independently as a docker container (virtual machine is also an option under evaluation). Together with the docker container (analytical service beck-end) each partner will develop a dedicated front-end using a common CSS framework. A common platform will orchestrate the user's entry to each service.

Each service will be developed as a docker container and it will contain everything needed to run a small piece of software. Each container includes all the code, its dependencies and even the operating system itself. This allows applications to run almost anywhere: a desktop computer, a traditional IT infrastructure or the cloud. (Figure 9).

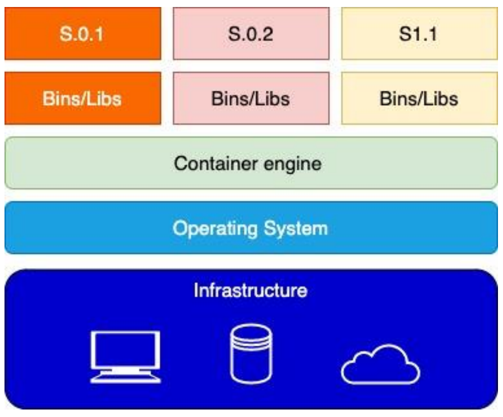


Figure 9: General Container infrastructure

Each application provides a specific front-end to the user in order to perform any analysis. Moreover, a general platform with a dedicated front-end will be developed to facilitate the choice of all services and to manage account security. The user after login will be able to access the chosen service based on the data provided, as shown in section 2.1.

3.3 Digital Building Twins

Digital Building Twin (DBT) is part of WP5 - "Analytics Building Services" and this section demonstrates an overview of the work carried out until now. DBT can be defined as a connected, digital representation of a physical building (district, city, region, etc.) and corresponding processes that are used to understand, predict and optimise performance in order to achieve a more cost-effective, straightforward and sustainable smart building. It brings together dynamic and static data from multiple sources in 2D/3D models and enables informed and effective decisions to be made. It provides real-time understanding of how a building is performing – enabling immediate adjustment to optimize efficiency and to provide data to improve the design and management of future buildings or districts.

There are several layer-based approaches to the architecture of a Digital Twin (DT). Starting with the core of the building's DT, three layers can be distinguished: a physical layer that corresponds to sensorisation (physical equipment that functions as data sources); a data layer in which all the available

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

data would be concentrated, (such as the data repositories and the real-time information); and finally the model layer, where the reasoning, behaviour, simulation and prediction models would be located (Figure 10).

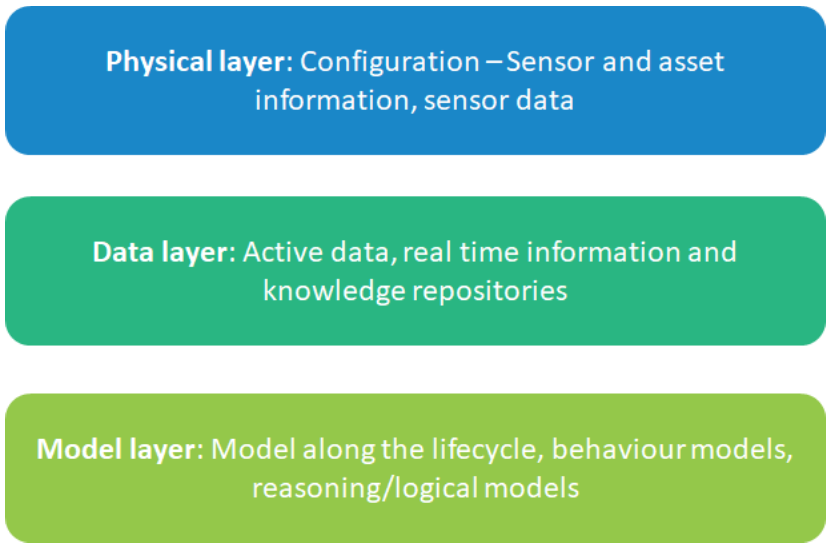

Physical layer: Configuration – Sensor and asset information, sensor data
Data layer: Active data, real time information and knowledge repositories
Model layer: Model along the lifecycle, behaviour models, reasoning/logical models

Figure 10: Digital Building Twin scheme

Other approaches expand the number of layers by adding a user interface layer or subdividing some of the previous layers (a cloud storage layer could be added in the data layer; an analysis layer could be included in the model layer; and the physical layer could also include communications, IoT). In MATRYCS project, these three main layers will be used and other layers will be included depending on the use cases and the specific requirements of each application.

This common model of DBT will be taken under consideration based on three layers at three different scales (building, district and region / nation) since the approach is different in each case (e.g. file formats, services). These scales correspond to the main typologies of the MATRYCS use cases.

The main objectives of the development of the DBT in MATRYCS are the following:

- The DBT will compile the relevant information to be used at three different scales to support the implementation of energy services.
- The scales considered are related to the main standard to be used in each case: (1) Building level (IFC), (2) District / City level (CityGML) and (3) Regional /National scale (INSPIRE data); which should be combined with the corresponding real and monitored data in each scale.
- The DT will support the deployment of other services by integrating all relevant information at each scale, and automatically generating models based on other data sources if they do not exist.
- As an overarching service, the related business cases could be applicable in every intermediate scale and be related to the stakeholders that deploy their activity at that level. In principle, all LSPs could benefit from the DBT service.

In MATRYCS project, DBTs will be considered at three different scales: building, district and regional / national.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

3.3.1 Digital Building Twin – Building level

At the building level scale, the following specific objectives are proposed:

- Create a coherent aggregation of data from multiple sources and link them to a digital model of the building to obtain an initial DT.
- Integrate the different analytics building services in the DBT answering the needs specified in the pilots and captured in the user stories and the specifications.
- Modelling the elements and behavior of the entities taking part in the DBT.
- Prepare/adapt the DBT to the required inputs and outputs, the required predictions and behavioral analysis, as well as the features to be applied for the ML and DL definitions.
- Use cases addressed (LSP1 & LSP2):
  - Use Case 1 [s0.1 DBT] (UC00_01: Digital Twin creation)
  - Use Case 2 [s0.1 DBT] (UC00_02: Services integration in Digital Twin).

As a first approach to the development of the DBT, the integration of sensor data and historical data in BIM models (using commonly accepted formats: IFC and RVT mainly) will be proceeded.

Sensor data can be integrated into Revit BIM models[8] with the Revit tool Dynamo[9]. It allows to create predefined scripts and execute them from Revit, asking the final users for a few necessary inputs only. First, the IFC file can be imported into Revit. Then, the scripts in Dynamo are created and executed through the Dynamo Player, placed in the Revit interface, to execute the following orders (Figure 11):

- To create a "sensor box" element.
- To define parameters that will represent the data in the "sensor box" element.
- Collect the sensor data directly from the excel/csv/other provided file and represent it into the previously defined parameters.
- To operate with the data. For instance, a schedule can be created and exported it to PDF.
- To export the model to an IFC file with the new data inputs into it.

[8] https://www.bimobject.com/en/product?sort=trending
[9] https://dynamobim.org/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

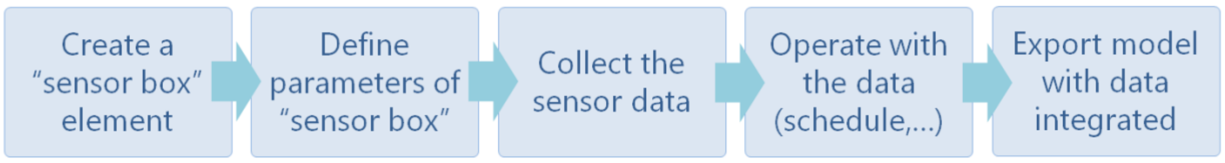

Create a "sensor box" element
Define parameters of "sensor box"
Collect the sensor data
Operate with the data (schedule,...)
Export model with data integrated

Figure 11: Revit Tool Dynamo data flow

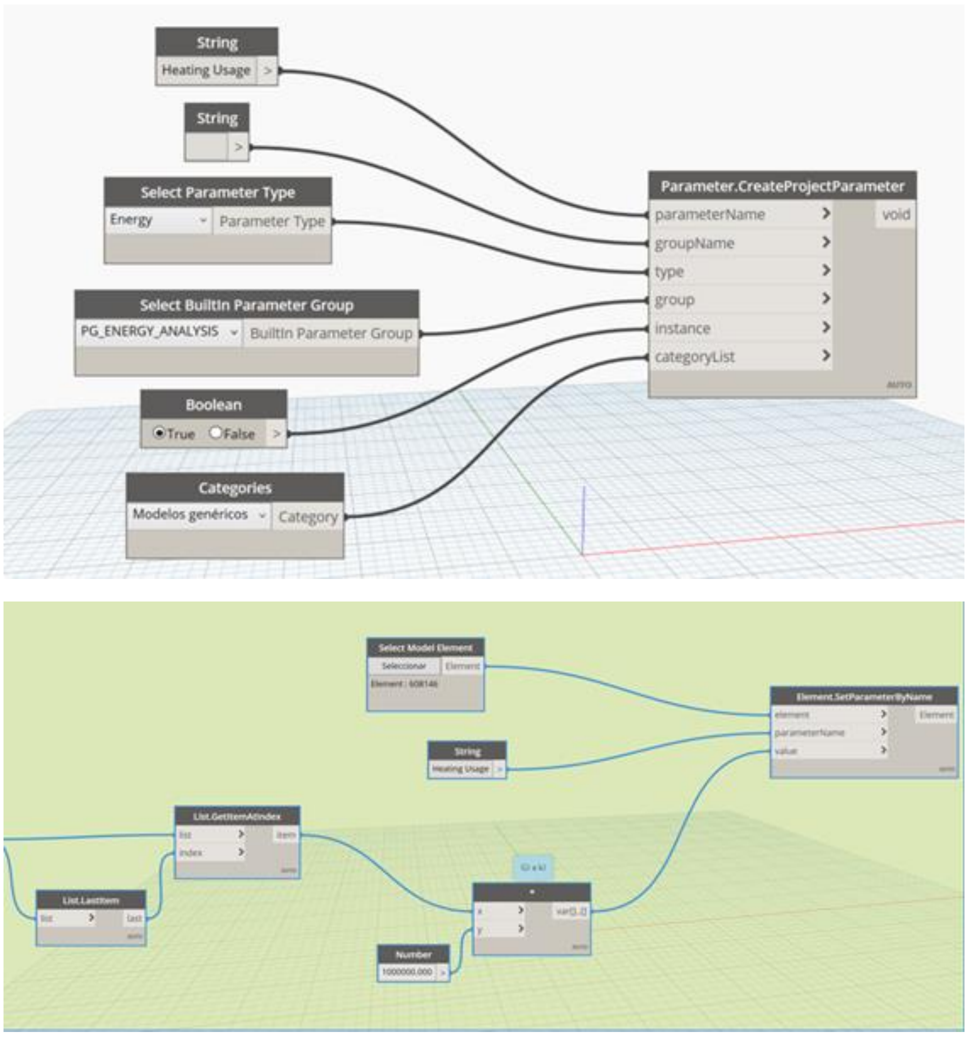


Figure 12: Examples of Revit/Dynamo scripts for integrating sensor data into BIM files

Regarding the visualisation part, the use and programming of several open-source BIM tools will be evaluated. The tools that are considered in the first instance that could be useful are: Xbim Xplorer[10] and BIMvision[11] (Figure 12).

The Xbim[12] toolkit is a .NET open-source software development BIM toolkit that supports the BuildingSmart Data Model (aka the Industry Foundation Classes IFC).

[10] https://docs.xbim.net/downloads/xbimxplorer.html
[11] https://bimvision.eu/
[12] https://docs.xbim.net/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Xbim allows .NET developers to read, create and view Building Information (BIM) Models in the IFC format. There is full support for geometric, topological operations and visualisation. In addition, Xbim supports bi-directional translation between IFC and COBie formats. Core libraries for data manipulation are all written in C#, core of geometry engine is written in C++.

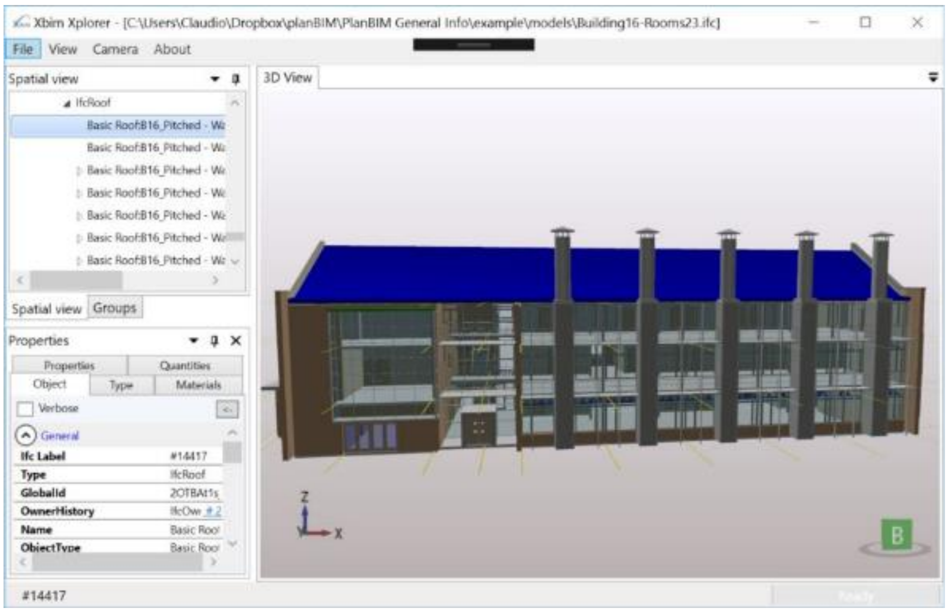


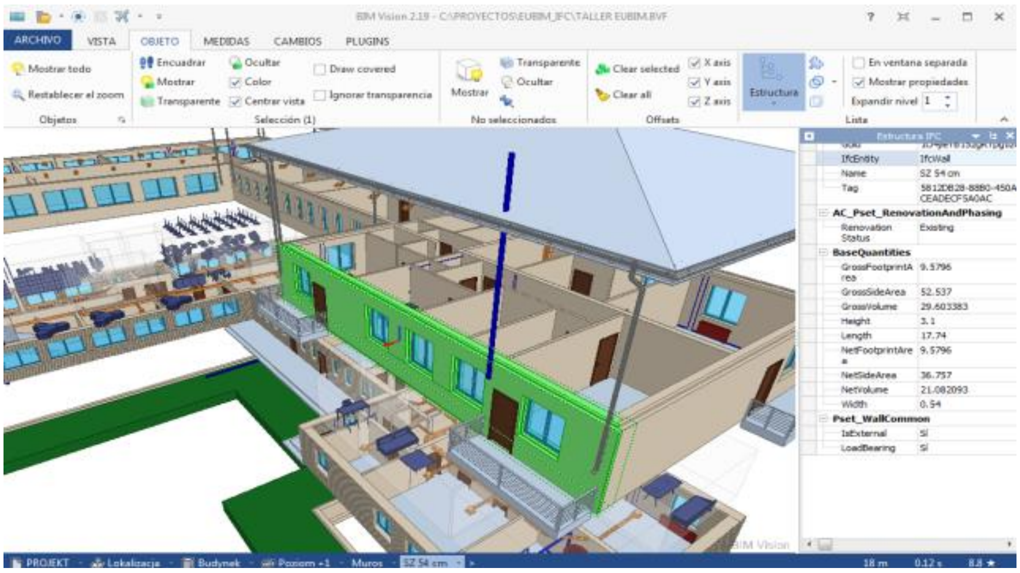
Figure 13: Xbim explorer sample interface

BIMvision is a freeware IFC model viewer (Figure 13). It allows to view the virtual models coming from CAD systems like Revit[13], Archicad[14], BricsCAD BIM[15], Advance[16], DDS-CAD[17], Tekla[18], Nemetschek VectorWorks[19] and others without necessity of having commercial licenses of these systems or having each of particular vendor's viewer. BIMvision visualises the BIM models created in IFC format 2×3 and 4.0.

It supports a plugin system that can be extended by way of other plugins, in order to develop specific viewing or data processing extension.

[13] https://www.autodesk.com/products/revit/overview
[14] https://graphisoft.com/solutions/archicad
[15] https://www.bricsys.com/en-intl/bricscad-bim
[16] https://www.graitec.com/advance-bim-designers-2020-is-here/
[17] https://www.dds-cad.net/
[18] https://www.tekla.com/
[19] https://www.vectorworks.net/en-GB

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158



IfcEntity	IfcWall
Name	SZ 54 cm
Tag	5812DB28-8880-450A CEADECF5A0AC
AC_Pset_RenovationAndPhasing	
Renovation Status	Existing
BaseQuantities	
GrossFootprintArea	9.5796
GrossSideArea	52.537
GrossVolume	29.603383
Height	3.1
Length	17.74
NetFootprintArea	9.5796
NetSideArea	36.757
NetVolume	21.082093
Width	0.54
Pset_WallCommon	
IsExternal	Sí
LoadBearing	Sí

Figure 14: BIM vision sample interface

It is considered that the use of these tools could complement the visualisation envisaged in the MATRYCS Framework without relying on commercial software such as Revit or ArchiCAD. In any case, the final decision to use these tools will depend on the results obtained in the first iteration of the development and the ease of integration in the MATRYCS platform.

3.3.2 Digital Building Twin – District level

The objective of this part is the modelling and integration of geolocated data at district level to develop simplified CityGML[20] building models that helps in the estimation of the energy demand improving the accuracy of the calculations and characterization of buildings.

The following specific objectives will be achieved:

- To identify and create processing methods to characterize the geographical distribution of buildings heights including algorithms to collect data.
- To evaluate the relevant data that helps building characterization.
- To develop a standardised DBT model at district level.

The datasets necessary to build the DBT at district level are:

- Buildings from cadastres, if are available for the pilot. Building attributes are also useful to characterise demand estimations.
- Buildings from OpenStreetMaps (OSM), if cadastral data are not available.
- LiDAR[21] data to improve height calculation. These data need to be pre-processed to identified data that were classified in a wrong class.
- Copernicus data from "Urban Atlas" to identify changes and/or heights if LiDAR data are not available.

[20] http://www.citygml.org/
[21] https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/what-is-lidar-.htm

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

At this first stage, some challenges are identified such as data availability and their accuracy. Depending on the available data (cadastres or OSM), the geometries and attributes are different and can affect the final results at district level. Depending on the district size, data could be large and pre-processing steps will be necessary to avoid less accurate results.

In the next stage, these issues will be addressed by identifying available data in each pilot; by evaluating CityGML requirements and integration methods for different datasets; and by defining pre-processing methods if necessary.

3.3.3 Digital Building Twin – Regional level

The target of the DBT in the regional level is to create a coherent aggregation of data from multiple sources, in order to obtain a real version of the energy performance of the building in a given region.

The usefulness of this energy DT could be tested mainly in decision-making processes, where it would be necessary to have a clear idea about the usage and consumptions of energy, its geographical distribution, and what would happen if an external factor modified the situation, at regional level.

This service will be used as basis for the calculation in other service: mainly S1.3 (KPIs calculation) and S3.2 (EPC harmonization and compliance). But also, this DT will be fed by the results of these services and from other sources.

The use cases will be centered in the LSP9: The Use Case 1 (UC09_01, EPCs data error checker and validator) and Use Case 2 (UC09_02, Visualisation of estimated EPC) for the s3.2 service, as well as the Use Case 1 and 2 for the s0.1 service considering Digital Twin (UC00_01, UC00_02, Services integration in Digital Twin).

Currently the following sets of data are under process:

- **Spanish cadastre:**
  - Relevant data for Castilla y León region is collected (2248x4 files: BU, Bupart, CP and CZ files - 43.02GB).
  - Uploading to the platform.
  - There is an online service of not protected data, but up to date, the information cannot be downloaded in batch.
- **OpenStreetMaps**:
  - Maps that can be consulted online.
  - Possibility to download all the data (directly or using API).
- **Regional Spanish EPC:**
  - Public EPCs offered by Castilla y León. Available in web.
  - EPC XML repository of Castilla y León. This information is not public (it belongs to the regional administration). Currently, the process of collecting it is taken place. This is around 99.000 files (17,8 GB).

Up to date, some potential hurdles have been detected for this topic:

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

- The incomplete data might be not uniformly distributed, for example, the OSM data does not show information on the year of construction and use of the buildings, or there is a lack of relevant information in the cadastre data for some areas. Therefore, the analyses could offer results not 100% valid on certain cases.
- Some data sources might not be available in certain European countries, and the alternatives should be feasible.
- The large quantity of data will require the usage of Big Data methodologies in order to handle and store them. Therefore, it will be necessary to pay attention to the processing of data, so there would be relatively small files for general reports.

In the next stage these issues will be addressed:

- Analyse the datasets: ongoing (for example, deciding if cadastral data can be complemented with data taken from the not protected data of the cadastre);
- Decide the data model to use for DBT;
- Geometrical part and data to be added from the results.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

4 MATRYCS Toolbox Platform as a Service Layer

In the scope of the MATRYCS project, a Virtual Workbench at the PaaS level will be built. Its goal is to combine various data sets, third party services, ML models and storage resources into an environment suited for new services creation. It will enable in a wide range of potential stakeholders such us SMEs, developers and innovators to use the provided set of tools for the design and development of new services for the building sector.

The Virtual Workbench will provide a user-friendly UI to help in building new services that rely on Big Data and ML models created in the MATRYCS project. Besides Training Models Library, the Virtual Workbench allows access to different data sources. The ability to have both data and ML models easily accessible under a single environment will facilitate the creation and testing of potential new services at scale and reduce the time required for developers and SMEs to create them.

Knowage Suite[22] is chosen as an open-source suite that can provide all the functionalities needed for the Virtual Workbench Module. Knowage Suite is equipped to meet all the needs of advanced analyses and management of Big Data, traditional sources, trained ML models, together with business intelligence suite. Knowage Suite will provide multidimensional data analysis, charting, query builder and data management tools, with which it will target a wide set of energy stakeholders. Special attention will be on the management, reuse and configuration of selected ML models from the library of MATRYCS trained models.

During this first stage of the project, the Knowage Suite has been installed on-premises into a dedicated VM using Docker technologies as described in the official installation manual[23]. The Knowage open-source code is available on the dedicated GitHub repository[24].

4.1 Building Data and Services Innovation Hub

The Virtual Workbench's aim is to be at the disposal of the stakeholders that are interested in creating new services for the building sector. To this purpose, the Building Data and Services Innovation HUB will provide a set of functionalities to facilitate the creation of such services quickly and easily through an intuitive user interface. The project datasets and the pre-trained models that are the results of the Data feed module and Model Training Module will be provided to the Virtual Workbench for analysis and reuse.

The main functionalities that will be made available within this module are:

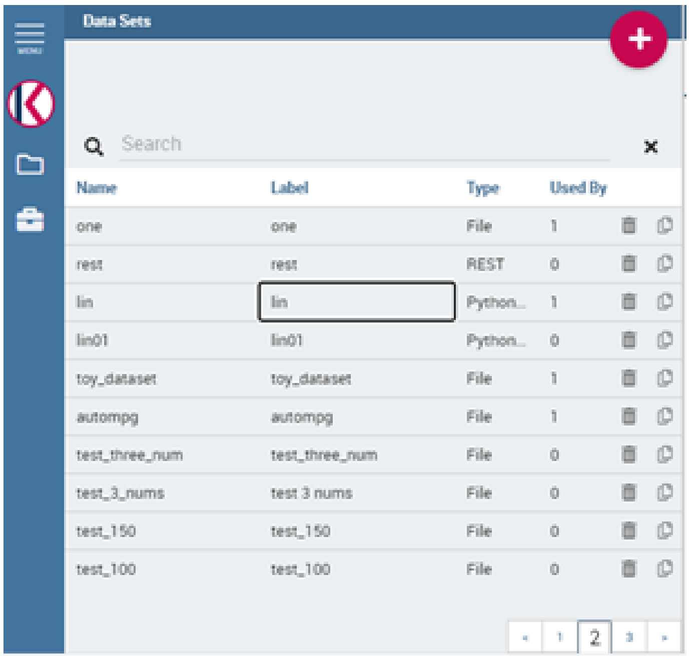
- a catalogue of ML models among those that will be developed, trained and evaluated within the project;
- a catalogue of data sets among the ones provided by the Data Feed module which will be useful for the execution of ML models;
- a catalogue of functions through which it will be possible to define functions/services resulting from appropriate use of ML models and data.

[22] Knowage suite web site: https://www.knowage-suite.com/site/
[23] Knowage GitHub repository: https://hub.docker.com/r/knowagelabs/knowage-server-docker/
[24] https://github.com/KnowageLabs/Knowage-Server

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

The data catalogue will provide specific functionality to access the data sets provided by the Data Feed Module. It will also be possible to define custom data sets for special needs through the creation of datasets from comma-separated files or through specific python scripts. In this first phase of the project, direct access to the Data Feed module was tested in order to retrieve the first project datasets.

At M9, after some performance tests related to Data Feed Module activities, a new technological solution based on the document-oriented database MongoDB[25] instead of the columnar database ScyllaDB[26], is being evaluated to improve the performance of this module. For this reason, the impact of using MongoDB as data source in the *Building Data and Services Innovation HUB* will be evaluated during the 2nd technology release. Some functionalities of this module are transversal to the Data analytics module and will be better detailed in the next paragraph.



Name	Label	Type	Used By
one	one	File	1
rest	rest	REST	0
lin	lin	Python...	1
lin01	lin01	Python...	0
toy_dataset	toy_dataset	File	1
autompg	autompg	File	1
test_three_num	test_three_num	File	0
test_3_nums	test 3 nums	File	0
test_150	test_150	File	0
test_100	test_100	File	0

Figure 15: Data Sets list

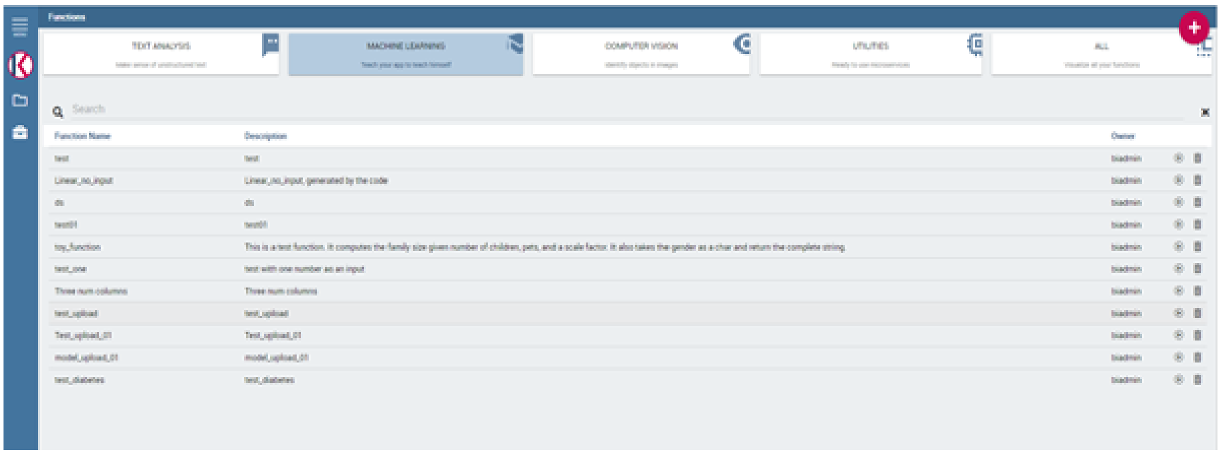
The ML model catalogue (Figure 15) will provide a hierarchical view of the ML models available within the Virtual Workbench. It will provide information about the characteristics of each ML model through specific attributes and metadata that will highlight the: use, type, short description, accuracy and performance, usage of the model, format of the input and related output. All this information will help end-users and developers to choose the appropriate ML model and work with it correctly and most efficiently. Through this functionality, it will also be possible to load new ML models, remove obsolete ones and, through an appropriate link, reach the function catalogue to see where the ML model is used and create new functions/services.

Besides the data catalogue and the ML model catalogue, one of the most important parts of this module is the function catalogue which will use both catalogues. Up to now, the ML models are saved in a dedicated directory (Figure 16) of the server where the Virtual Workbench runs in order to be used. By this module, it is possible to define functions for different purposes and goals. It is viable to write Python code to be run in the Knowage suite's functions. By means of this catalogue, it will be possible to define different functions thus prepare the basic environment for the Data Analytics module that will permit to

[25] https://www.mongodb.com/
[26] https://www.scylladb.com/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

analyse the functions' results through the construction of specific cockpits that will be better described in section 4.2.


Functions
TEXT ANALYSIS
Make sense of unstructured text
MACHINE LEARNING
Teach your app to teach himself
COMPUTER VISION
Identify objects in images
UTILITIES
Ready to use microservices
ALL
Visualize all your functions
Search

Function Name	Description	Owner
test	test	biadmin
Linear_no_input	Linear_no_input, generated by the code	biadmin
ds	ds	biadmin
test01	test01	biadmin
toy_function	This is a test function. It computes the family size given number of children, pets, and a scale factor. It also takes the gender as a char and return the complete string.	biadmin
test_one	test with one number as an input	biadmin
Three num columns	Three num columns	biadmin
test_upload	test_upload	biadmin
Test_upload_01	Test_upload_01	biadmin
model_upload_01	model_upload_01	biadmin
test_diabetes	test_diabetes	biadmin

Figure 16: Function catalogue

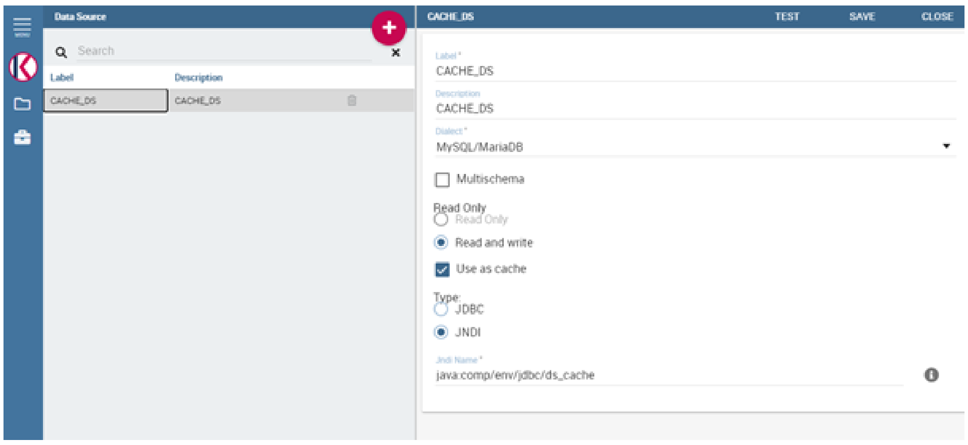
4.2 Virtual Workbench - Data Analytics Environment

The Virtual Workbench - Data Analytics environment will allow users (e.g. analysts, data scientist) to perform freeform queries and data analytics on the assets which are accessible to the platform (models and data). The data processing actions that are defined by queries are used to present received data graphically on the well-defined user interface. To show the results of the queries and functions execution the Knowage Suite visualisations and reports engine will be exploited. It can represent data in different formats, including different kinds of dynamic charts, tables, HTML code, etc. These representations are chosen and created based on the MATRYCS project needs.

Knowage Suite enables exploration, organisation and processing of MATRYCS datasets provided by the Data Feed module. To access data, data sources should be defined in the Data source functionality in which the database connections must be configured to use Business Intelligence tools properly. A connection can be configured as JNDI [27]or JDBC[28]. At this stage of the project, the connection with Data Feed Module and ScyllaDB have been tested together with some custom Python data sets utilized for testing purposes.

[27] JNDI: https://en.wikipedia.org/wiki/Java_Naming_and_Directory_Interface
[28] JDBC: https://en.wikipedia.org/wiki/Java_Database_Connectivity

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158


Data Source

Search

Label	Description
CACHE_DS	CACHE_DS

CACHE_DS TEST SAVE CLOSE

Label *
CACHE_DS

Description
CACHE_DS

Dialect *
MySQL/MariaDB

Multischema

Read Only
Read Only
Read and write

Use as cache

Type
JDBC
JNDI

Jndi Name *
java:comp/env/jdbc/ds_cache

Figure 17: Example of cache data source configuration.

Querying of datasets enables dynamical change of data sets received by changing parameters in the SQL query. Knowage users can create datasets by uploading an XLS or CSV file. The dataset could be created from the Python script, where datasets are defined through Pandas Data Frames functionality, for example. Moreover, particular APIs can be queried, and datasets can be received from them. To analyse the data and give the option to work with these, the Data Analytic environment provides the users with the functionality to create their own Cockpits with the possibility to intuitively utilize different widgets such as tabular widgets, different charts, images, text, HTML, and particularly, Python widget.

Besides tables, charts are crucial in the representation of data and data characteristics. One of the characteristics of the Cockpit is to enable users to create analytical documents using different widgets and defining associations among them, so when data are selected in one widget, they are automatically updated in the rest of the widgets. On the other hand, the different widgets can work with different datasets and in that way with different data sources and also with the defined functions showing their results. The following images (Figure 18 and Figure 19) show examples of different types of widgets that can be used to analyse data within the Virtual Workbench.

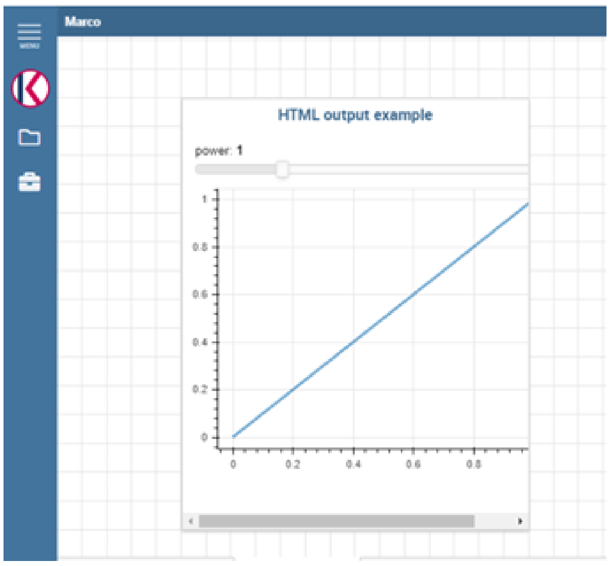
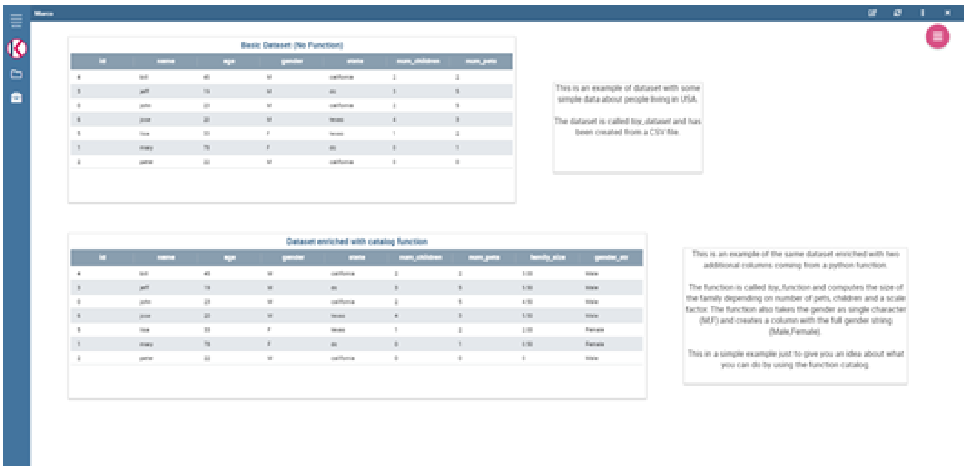


Figure 18: Example of HTML widget

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158



Basic Dataset (No Function)

id	name	age	gender	state	num_children	num_pets
4	bill	45	M	california	2	2
3	jeff	19	M	dc	3	5
0	john	23	M	california	2	5
6	jose	20	M	texas	4	3
5	lisa	33	F	texas	1	2
1	mary	78	F	dc	0	1
2	jane	22	M	california	0	0

This is an example of dataset with some simple data about people living in USA.

The dataset is called toy_dataset and has been created from a CSV file.

Dataset enriched with catalog function

id	name	age	gender	state	num_children	num_pets	family_size	gender_str
4	bill	45	M	california	2	2	3.00	Male
3	jeff	19	M	dc	3	5	5.50	Male
0	john	23	M	california	2	5	4.50	Male
6	jose	20	M	texas	4	3	5.50	Male
5	lisa	33	F	texas	1	2	2.00	Female
1	mary	78	F	dc	0	1	1.50	Female
2	jane	22	M	california	0	0	0	Male

This is an example of the same dataset enriched with two additional columns coming from a python function.

The function is called toy_function and computes the size of the family depending on number of pets, children and a scale factor. The function also takes the gender as single character (M,F) and creates a column with the full gender string (Male,Female).

This in a simple example just to give you an idea about what you can do by using the function catalog.

Figure 19: Example of Table widget

The Virtual Workbench has a Python standalone web service in order to enable running Python scripts and receiving results from the Python environment. Results of analysis can be shown in these widgets, which can be grouped into multiple Cockpits. The Python widget facilitates the direct usage of Python scripts in the Cockpit to develop advanced custom analytics of chosen datasets.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

5 MATRYCS Toolbox Infrastructure as a Service Layer

5.1 Cloud Infrastructure and Service Orchestrator

Within the MATRYCS-ANALYTICS layer, a set of analytics is provided, including the in IaaS. At this level, the platform will package its services in container-based workflows that could be flexibly deployed over cloud infrastructures through a service orchestrator. The IaaS model will make possible to dynamically scale applications according to real needs, based on a light virtualisation.

At this stage, different cloud provider options are being evaluated. In order to provide an overview of the actual needs to be addressed, a set of requirements is being compiled, including information on:

- Number of CPU cores
- Minimum processor speed (GHz)
- Amount of RAM per CPU CORE (GB)
- Local disk (GB)
- Number of VM instances
- Number of public IP
- Volume (GB)

This information is under evaluation, considering a triple perspective:

- **Components' requirements**: The needs of the MATRYCS-GOVERNANCE layer in order to implement WP3 modules.
- **Services requirements**: Details of cloud resources are collected for Cloud Computing and cloud container computing, considering each service separately.
- **Storage requirements**. The data sources to be used are being collected and in process of integration. Depending on the size of the historical information and the information to be updated and its frequency, different alternatives could be considered.

The last element that is being analysed is the feasibility of the different alternative services in terms of costs, and considering the resources allocated for it (Task 3.4 - Data Storage and High Performance, Distributed Query Engine).

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

6 Continuous Integration and Testing

The task 4.6 – *Continuous Integration & Testing* includes all the activities regarding the integration of MATRYCS components and the Testing Plan that should be applied over MATRYCS Toolbox. The Integration and Testing Plan facilitates the design and planning of the platform's functionalities and enhance the offering of the platform with the introduction of new functionalities and fixing the issues of current features.

Continuous Integration & Testing activities that are reported in this 1st technology release include the research conducted for Source Code Management principles that are applied in MATRYCS code base (Github, Git branching methodology, code commits, merge requests), the definition of technologies to be used as an Automation Server (Jenkins) and the definition of Testing activities for MATRYCS Toolbox. More specifically the Testing Plan defined for MATRYCS Toolbox includes Unit Testing for testing each MATRYCS component functionality; Integration Testing for testing the MATRYCS components connections; User Acceptance Testing for evaluating in high level format the usage of MATRYCS Toolbox (using acceptance cards, end users' feedback); Requirement Coverage for evaluating if MATRYCS Toolbox covers the MATRYCS Technical Requirements and KPIs; and the definition of Error Tracking Applications that would be integrated with MATRYCS components.

Regarding the Integration and Testing plan, the methodology that has been followed in MATRYCS project is based on the Systematic Test and Evaluation Process (STEP) approach [4]. The main concept of this approach is that an overall Testing Plan is designed in detail, taking into account that testing in the first stages of the MATRYCS Toolbox development will eliminate errors and bugs. Consequently, most of the issues could be resolved before they even occur during the initial testing stages. To this end, the goal of this approach is the design of the testing framework as well as the corresponding use cases tests to take place in early stages.

The design of the testing framework will also take into account the specific objectives and requirements of the project for creating a unified framework and open solution for building sector stakeholders, without anticipating the software design process to be completed.

The testing approach for the MATRYCS toolbox following the principles of STEP approach includes the testing of individual components (unit testing) implemented in the early stages of the project. After the unit testing, the integration tests will be realised. Similarly, these integration tests will be implemented before the first release of the integrated MATRYCS platform, using simulated environments and interconnections, as necessary.

6.1 Source Code Management

Source Code Management (SCM) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. It is also synonymous with Version Control. Continuous Integration relies on Version Control system. Every change applied to the code base must be safely stored in a dedicated Version Control system. Once the code is version controlled it can be accessed by the CI/CD tools.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

6.1.1 GitHub

The source code management in MATRYCS is achieved by using GitHub[29], which is a web-based GIT repository manager with wiki and issue tracking features, using an open-source license. GitHub also includes code reviews, issue tracking, wikis and a variety of other features, including integrations for tools such as Slack[30], LDAP[31], JIRA[32] and Jenkins[33], as well as a complete API. The MATRYCS Github private Repository link is this https://github.com/Matrycs. The repository is private because the MATRYCS code base contains information such as usernames and passwords for crucial infrastructure after the MATRYCS 2nd technology release the code base will be public, where the code base will be finalised and the information has been encrypted.

6.1.2 Git Branching Methodology

There are several reasons to use a methodology or workflow:

〉 It allows releasing new features quickly and often.

〉 It separates the work in progress without affecting software stability in the integration process.

〉 It promotes agility and the developers do not lose time solving unneeded conflicts.

The Git Branching Methodology is helpful in order to comply with the following goals:

〉 Code conflicts and integration problems are discovered as soon as possible. It is definitely better to fix small features or bugfix often and receiving feedback than to fix large problems less often.

〉 All the technical team uses this scheme in a daily basis, so the workflow must be clear, simple and easy to apply.

The GIT branching model proposed is the GitFlow workflow which is based on keeping the master branch flawless after passing the integration tests.

Instead of a single main branch, this workflow uses two (or more) branches to record the history of the project. The Main (master) branch stores the official release history, and the Develop branch(es) serves as an integration branch for features. It is also convenient to tag all commits in the main branch with a version number.

The GitFlow branching model is based in creating a Develop branch in order to integrate all the features implemented during the development cycle. All the issues extracted generate a new branch starting from the Develop branch with the name "feature-number-issue-DoesThis" (two words description). Consequently, all the merges are done in the Develop branch.

The integration process can be done via Merge requests. The changes are made on the Develop branch and after assuring their quality and functionality, the Develop branch is merged with the Master branch. As a result, the later branch includes all the latest changes that are functional. It is preferable to work in small features and integrate to the Master branch often. In this way, it is possible to face the conflicts

[29] https://github.com/

[30] https://slack.com/intl/en-gr/

[31] https://ldap.com/

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

[33] https://www.jenkins.io/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

that could arise in advance. Consequently, it is feasible to generate releases from the Master branch at any given time. The release names should follow the pattern "v0.X"

In order to fulfil this objective, the Main branch is only updated from the Develop branch by the administrator, who has the responsibility to manually merge the two branches, using the tools that GitLab provides. The result must be a fully clean merge, with qualitative code which passes all the tests that have been specified.

Figure 20 presents the GitFlow Workflow schema.

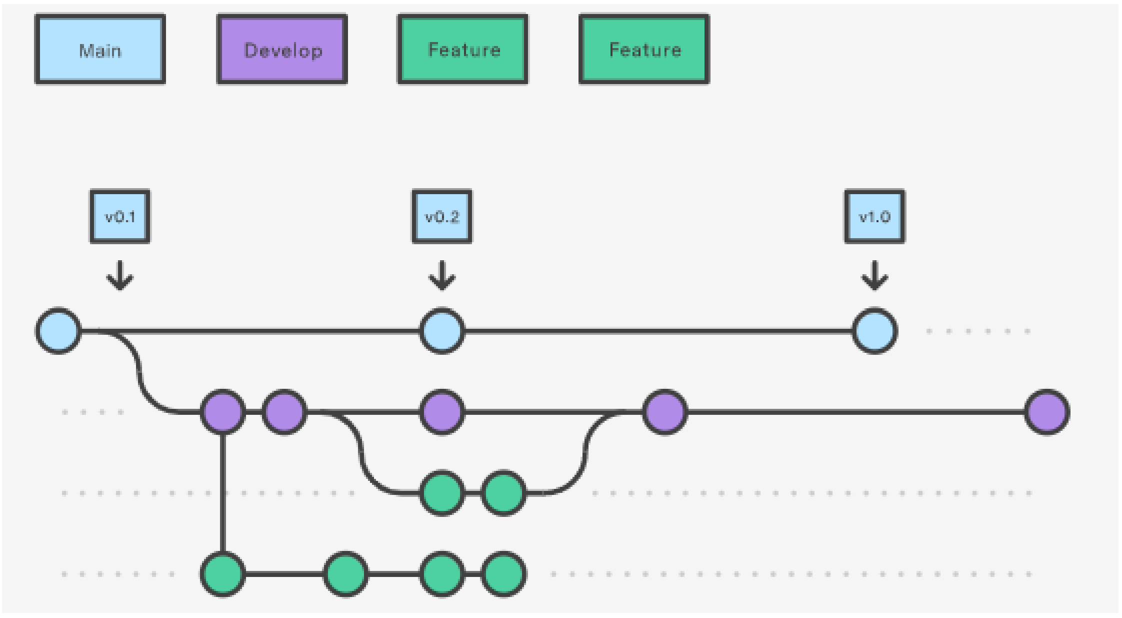


Figure 20: GitFlow Workflow schema

6.1.3 Features and Repository branches

The process to create a new branch is the following:

- When a new feature or issue in GitHub is created and it is assigned to one person (for instance the feature is called), the person assigned synchronises his/her Develop branch accordingly and creates a new branch from Develop, following the naming convention **feature-99-DoesThis.**
- In order to create a branch, firstly the Develop branch is synchronized:
  - git checkout develop
  - git pull
  - git checkout -b feature-99-DoesThis
  - git push -u origin feature-99-DoesThis
- In order to access in the local to the branch previously created by someone:
  - git fetch
  - git branch -r
  - git checkout feature-XX-DoesThis

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

6.1.4 Code commits

The rule of thumb is to commit often and with useful messages. When a merge request is raised, the process of code review is easy if there is a small number of well-commented commits. The commit messages should be aligned with the following rules:

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- Capitalise the subject line
- Do not end the subject line with a period
- Use the imperative mood in the subject line
- Wrap the body at 72 characters
- Use the body to explain what and why VS how

In the commit message, a solved issue is marked by using the character "#", with keywords like "Fix", "Resolve" or "Close".

6.1.5 Merge Requests

Merge requests allow to exchange source code changes and collaborate with other developers in the MATRYCS project. The process of Merge Requests in GitHub is as follows:

- The developer creates a branch with the name of the feature that must be developed:

  git checkout -b feature-XX-DoesThis

- The developer starts writing code and commits changes often, accompanied with clear messages:

  git commit -m "My feature is ready"

- The developer pushes the branch to the remote GitHub branch:

  git push origin feature-XX-DoesThis

- The developer reviews the code on commits page to check if it is functional.
- The developer creates a Merge Request clearly describing the functionality/pending issues.
- The administrator reviews the code, comments, and discusses it if needed and finally merges it to the develop branch.

6.2 Automation Server

In this section, an analysis of possible technologies for enabling CI/CD functionalities is presented. Travis[34] and Jenkins[35] have been identified as possible technologies for their specific characteristics and

34 Travis: https://www.travis-ci.com/

35 Jenkins: https://www.jenkins.io/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

both will be described and compared, so as to identify the best solution for the MATRYCS project.

Travis

Travis is a continuous integration service that allows building and testing software projects hosted on GitHub and Bitbucket. Travis CI with a cloud-hosted code does not require any installation, but an account on GitHub or Bitbucket and owner permissions for the project hosted there. Due to its tight integration with GitHub, it allows developers to independently test pull requests and branches, test results can be easily monitored on the GitHub UI.

Travis supports parallel testing and consequently the developers can speed up the testing phase by executing multiple builds in parallel, across different VMs.

In terms of Apps and Plugins, Travis has support for around 20 programming languages, and it can be integrated with tools such Slack, Email and others for notifications.

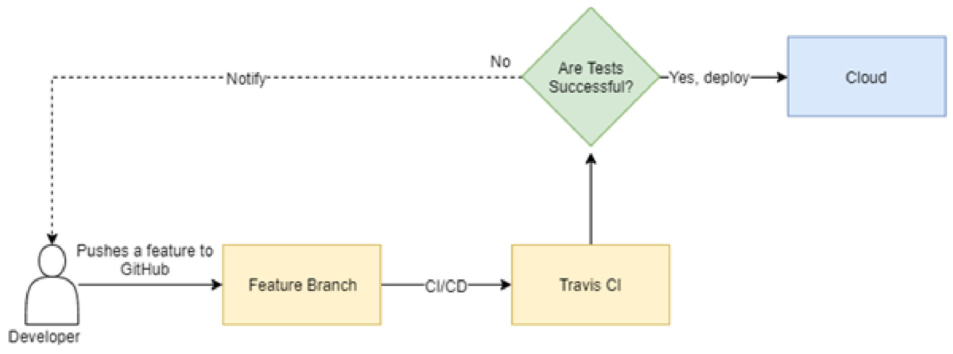


Figure 21: Travis workflow example

In Figure 21 a simple Travis workflow is presented. First step of enabling Travis is adding the *.travis.yml* file to the root directory of the repository. Pushing the feature or making changes to the repository triggers the Travis pipeline where it usually builds the software and runs all the automated tests. If the tests are successful, the software is being deployed to a specified place. Both successful and unsuccessful tests will result in notifying the team about its results. More detail about the Travis CI can be found at the official web site[36]:

Jenkins

Jenkins is an open-source automation server and one of the most used technology for CI/CD. It offers an automatized approach to the whole software development lifecycle including building, testing and deploying, which provides full CI/CD functionalities. Jenkins is written in Java which makes it work on almost every platform there is.

One of the main reasons for using Jenkins is its flexibility and how highly configurable it is. It is easy to install and configure, it has a huge number of plugins that supports and get supported by Jenkins as well. It is also extensible (extended via its plugin architecture), therefore providing nearly infinite possibilities for what it can do. It can easily distribute work across multiple machines, helping in faster builds, tests and deployments across multiple platforms.

[36] https://www.travis-ci.com/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

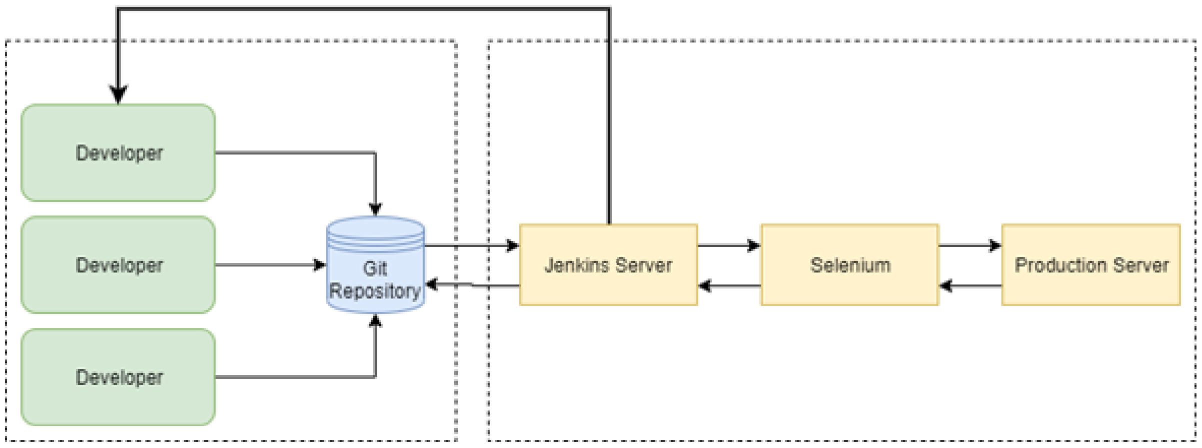


Figure 22: Jenkins workflow example

In the context of Figure 22, first, a developer commits the code to the repository. Jenkins is checking the repository at regular intervals for changes. Then, it recognised any changes and it will draw them by starting preparing a new build. If the build is successful, it deploys it to the test server, otherwise it notifies the team about its failure. After testing, Jenkins generates feedback and notifies the team about build and test results. If the build is successful and all tests are passed, the application is deployed to the production server. Jenkins will continue to monitor the code repository for changes, and the whole process above keeps on repeating.

Another good example of deploying an application would be Docker. After Jenkins copies the code repository, it would build a Docker Image based on the Dockerfile, followed by instantiating a container with the application code where the tests would be executed. If the tests are successful, the image is pushed to the Docker Trusted Registry. This example is shown in Figure 23.

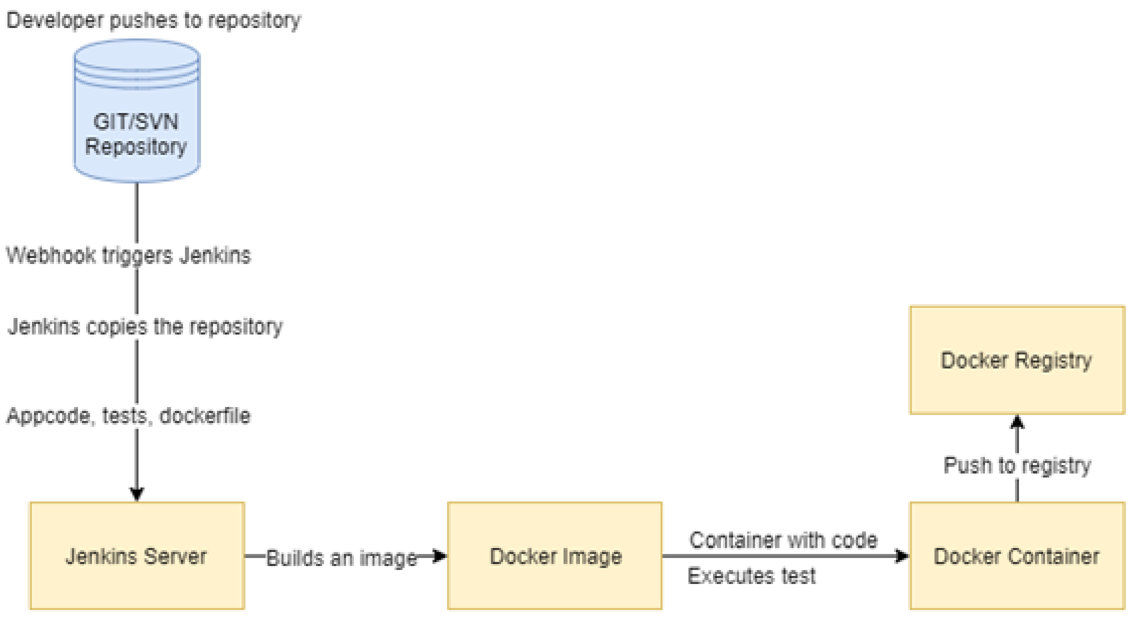


Figure 23: Jenkins example with Docker

Jenkins follow Master-Slave architecture to manage distribution. Main Jenkins server is the Master and its job is to handle scheduling build jobs; distribute builds and jobs to the slaves for execution; record

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

and present the results; and also work on jobs.

They communicate through TCP/IP protocol and its architecture is shown in Figure 24. More detail about the Jenkins CI can be found at the official website[37].

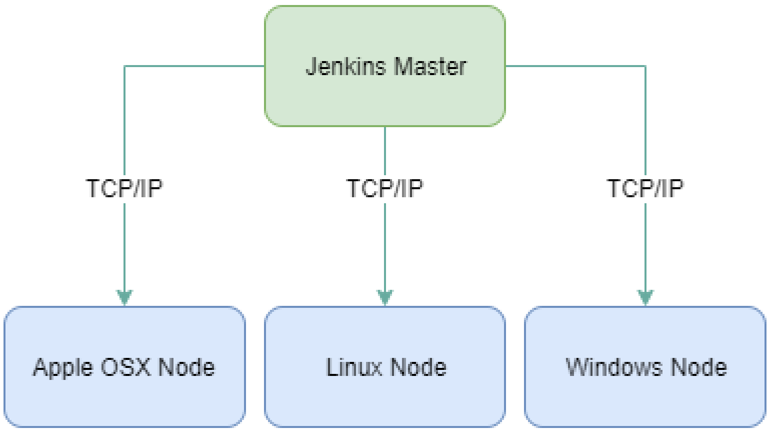


Figure 24: Jenkins's master-slave architecture

Due to high scalability of the MATRYCS project and a wide variety of technologies that it uses, the better solution would be the use of Jenkins. Even with a possibility of using new technologies in a future, Jenkins is more likely to gain certain plugins and functionalities regarding that new technology, which makes it way easier to integrate and configure in the already existing system. Below (Table 6) a comparison between the two analysed technologies based on specific identified features.

Table 6: Jenkins and Travis comparison

Feature	Travis	Jenkins
Customisation	Less	More
Online Documentation	Yes, blogs and official community	Yes, support forum, events and more
Ease of use	Focus on usability and functionality, good GUI	Customization via plugins
Setup and installation	No installation required	Easy
Business model	Free (open-source)	Free (open-source)
Apps & Plugins	Fewer things available	Over 1500 plugins
Cloud Integrations	Built-in support for AWS, Azure, Google Cloud, etc.	Amazon EC2, VMWare vSphere, Google Cloud, Slack, etc. through Jenkins Plugins

[37] https://www.jenkins.io/doc/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Feature	Travis	Jenkins
Configuration	Customisation through .travis.yml (YAML) file	Completely customisable

6.3 Toolbox Testing

In this sub-section, the methodologies and tools used during the MATRYCS Toolbox Testing Plan are presented. The approaches that are followed for the Unit Testing, Integration Testing as well as User Acceptance Testing are described in the following sub-sections. Moreover, Sentry[38] and Redmine[39] that are used for error tracking are presented at the end of this chapter.

6.3.1 Unit Testing

The main objective of this sub-section is to describe the applied unit tests in the MATRYCS toolbox integrated framework. This description cannot be exhaustive, given the fact that the software development is still in progress. During the 2nd technology release, where MATRYCS components will be more mature, Unit Testing procedures will be enriched. Unit tests are the tool to examine the functionality of individual components across the platform [6]. In the case of the MATRYCS Toolbox integrated framework, the quality of each component/layer developed in the corresponding WP/Task is guaranteed by unit tests. An appropriate unit test is applied to each part of code and each component, having no dependencies with other parts of code. Thus, the developer of each layer is testing the functionality of each component before the integration into the platform. These unit tests will be the initial ones of a big group of many tests that will be implemented for individual components and will run during the integration tests.

6.3.2 Integration Testing

Testing is still the primary means for quality assurance. A lot of testing techniques that have been proposed in the past, mainly focus on module testing. However, in practice, integration testing is often the most time consuming and expensive part of the testing [7].

After the development of all the individual components is finished, they are ready to be integrated into a system which will then be tested in order to check if it works seamlessly and if it meets the specifications that have been set.

This kind of testing is referred to as Integration Testing [8], as it tests the integration of the separate units into the overall system. Specifically, Integration Testing's aim is to diagnose design errors of the system or the system's units' specifications, as well as the interaction and the interfacing between them.

As long as the different units of the system are tested in combinations, eventually all the units comprising a process will be tested together. The discovered errors during the integration testing are mainly related to the interfaces between them, as all units have already been tested separately during the unit testing

[38] https://sentry.io/welcome/

[39] https://www.redmine.org/

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

portion of the testing process.

There are various strategies to perform Integration Testing. The most common ones are listed and briefly explained below:

- The **Top-Down Approach** (Figure 25) achieves step-by-step verification of the interfaces among components that operate under a common control strategy. This control strategy dictates the order of development, integration and testing. Top-down integration interleaves component scope testing and integration of a system's components. Table 7 presents the advantages and disadvantages of this approach.

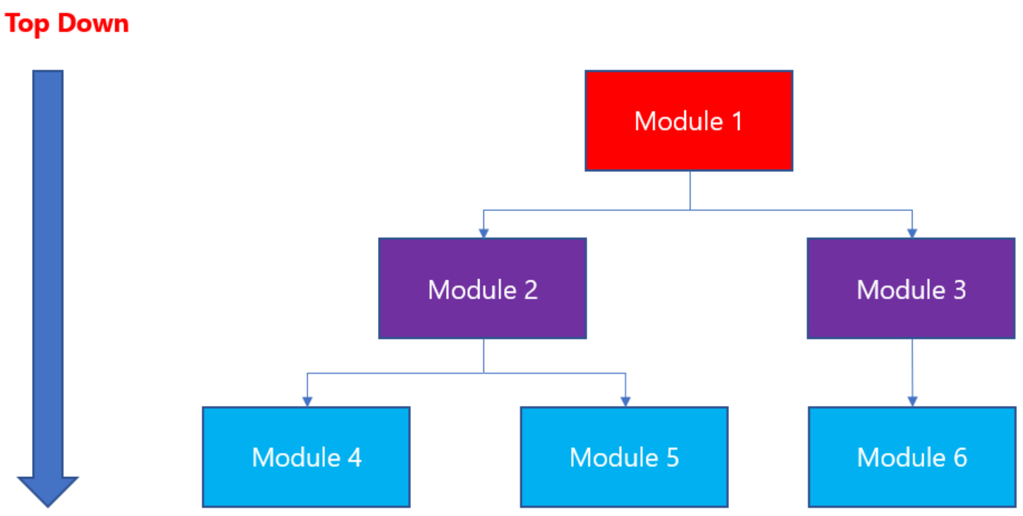


Figure 25: Top-Down approach for Integration Testing

Table 7: Advantages & Disadvantages of the Top-Down Approach

Advantages	Disadvantages
Fault localization is easier	Modules at a lower level are tested inadequately
Possibility to obtain an early prototype	
Critical Modules are tested on priority; major design flaws could be found and fixed first	

- The **Bottom-Up Approach** (Figure 26) achieves step-by-step verification of the interfaces between tightly coupled components. It interleaves component scope testing and integration of system components. Components with the least number of dependencies are tested at first. Table 8 presents the advantages and disadvantages of this approach.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

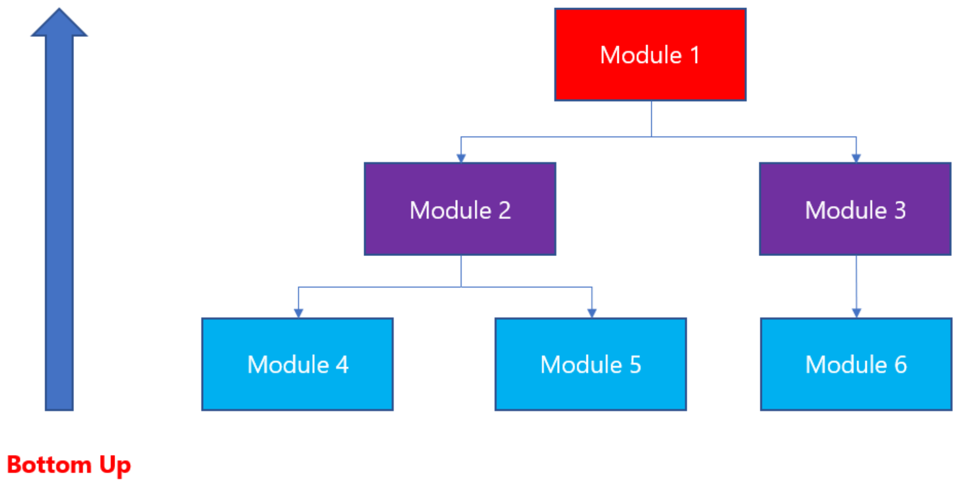


Figure 26: Bottom-Up approach for Integration Testing

Table 8: Advantages & Disadvantages of Bottom-UP approach

Advantages	Disadvantages
Fault localization is easier	Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects
No time is wasted waiting for all modules to be developed	An early prototype is not possible

- The **Hybrid** Approach (Figure 27) uses a combination of the Top-Down and the Bottom-Up approaches and performs testing with functional data along with control flow paths. Firstly, the inputs for functions are integrated in the Bottom-Up pattern discussed above. The outputs for each function are then integrated in the Top-Down manner.

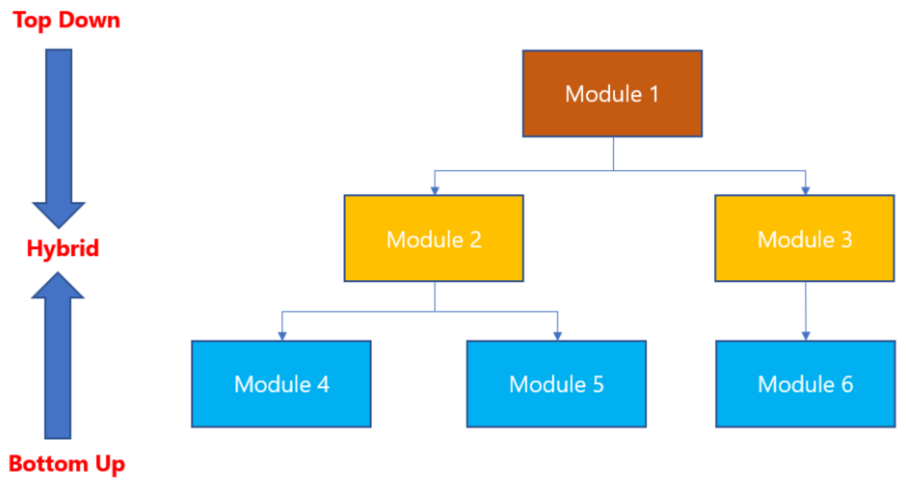


Figure 27: Hybrid approach for Integration Testing

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

For MATRYCS, the Hybrid Approach was chosen, as it allows multitarget testing to occur in parallel. This approach may require potential software components and subcomponents to be simulated for the sake of testing before they become operational, but overall, it allows for a greater test coverage of the global framework.

6.3.3 User Acceptance Testing

Errors in requirements' specifications have been identified as a major contributor to costly software project failures. Consequently, it would be highly beneficial if information systems' developers could verify requirements by predicting workplace acceptance of a new system based on user evaluations of its specifications measured during the earliest stages of the development project, ideally before building a working prototype [9]. Acceptance tests are high-level tests created by business customers, ideally in collaboration with analysts, end users, developers and testers. The language used in these tests is related to business domain. In each iteration, a set of user stories is validated in terms of completeness and entirety.

Acceptance cards have been created before the development so that the developers can understand clearly what they have to execute. Furthermore, acceptance cards can be created during the iteration planning. It is often the case that acceptance tests include more than one story (not implemented in the same iteration) and there are plenty of ways to test them. A well-known approach is to simulate the case in which external interfaces or data from other stories, with low probability to happen in the same iteration (since these stories have lower business priority), are co-existing. A user story is considered completed after all the acceptance tests have passed.

Regarding the MATRYCS Toolbox, the needs of user acceptance that will be implemented has been derive from the requirements and specifications of the Use Cases described in D2.2 - *Technical & Security Specification (April 2021)*. Taking into account these requirements and specifications, the MATRYCS Toolbox will be able to meet the needs as well as the specific scenarios of the use cases.

6.3.4 Requirement Coverage

The Integration and Testing Plan is also taking into account the coverage of the requirements of the MATRYCS project. These requirements have been shaped based on the Use Case scenarios, which have already been finalised and presented in D2.2 - *Technical & Security Specification* (April 2021).

The following table (Table 9) demonstrates the MATRYCS Technical Requirements reported in D2.2

Table 9: List of MATRYCS Technical Requirements

ID	Technical Requirement Description
TR050	The MATRYCS platform must provide system modularity, providing the independent operation of its modules
TR051	The MATRYCS platform must guarantee system recoverability
TR052	The MATRYCS platform must provide the possibility for future functionalities, ensuring system scalability and extensibility

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

ID	Technical Requirement Description
TR053	The MATRYCS platform will follow an incremental approach which facilitates the design and planning off the platform's functionalities
TR054	Each MATRYCS development iteration its focused on fixing and addressing the issues of the previous iteration but at the same time to provide on extending the list of offerings of the platform with new functionalities
TR055	MATRYCS Components Integration and formulation of the final platform
TR056	MATRYCS platform functional, integration and stress testing procedures
TR057	The MATRYCS platform must support software implementation in many programming languages

Effort will be put towards ensuring the requirements are satisfied, taking into account the main principles of Software Quality (Presence of defects, exhaustive testing is not possible, early testing, defect clustering, pesticide paradox, testing is context dependent, absence of errors fallacy). In short, this entails the following:

- Functional Sustainability
  - Functional Completeness
  - Functional Correctness
  - Functional Appropriateness
- Performance Efficiency
  - Time Behaviour
  - Resource Utilisation
  - Capacity
- Compatibility
  - Co-existence
  - Interoperability
- Usability
  - Learnability
  - Operability
  - User Error Protection
  - User Interface Aesthetics
  - Accessibility
- Reliability

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

    - Maturity
    - Availability
    - Fault Tolerance
    - Recoverability
- Security
    - Confidentiality
    - Integrity
    - Non-repudiation
    - Authenticity
    - Accountability
- Maintainability
    - Modularity
    - Reusability
    - Analysability
    - Modifiability
    - Testability
- Portability
    - Modafiability
    - Installability
    - Replacebility

Adhering to all these principles, with an individual focus on the requirements, as expressed through the use case scenarios, the resulting platform will guarantee the quality of its individual components and the set itself.

6.3.5 Error Tracking Application

Sentry

The testing process is fundamental for building robust applications, since the seamless and consistent operation of the services is ensured. For monitoring the errors and bugs generated by the MATRYCS Toolbox, Sentry has been installed in NTUA premises and will be moved to the MATRYCS cloud infrastructure once available to the consortium.

Sentry [10] is a powerful open-source application monitoring platform that can be connected with external components for error monitoring and bug fixing. Sentry facilitates the work of developers because it helps in the diagnosis, fixing and optimization of their code.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

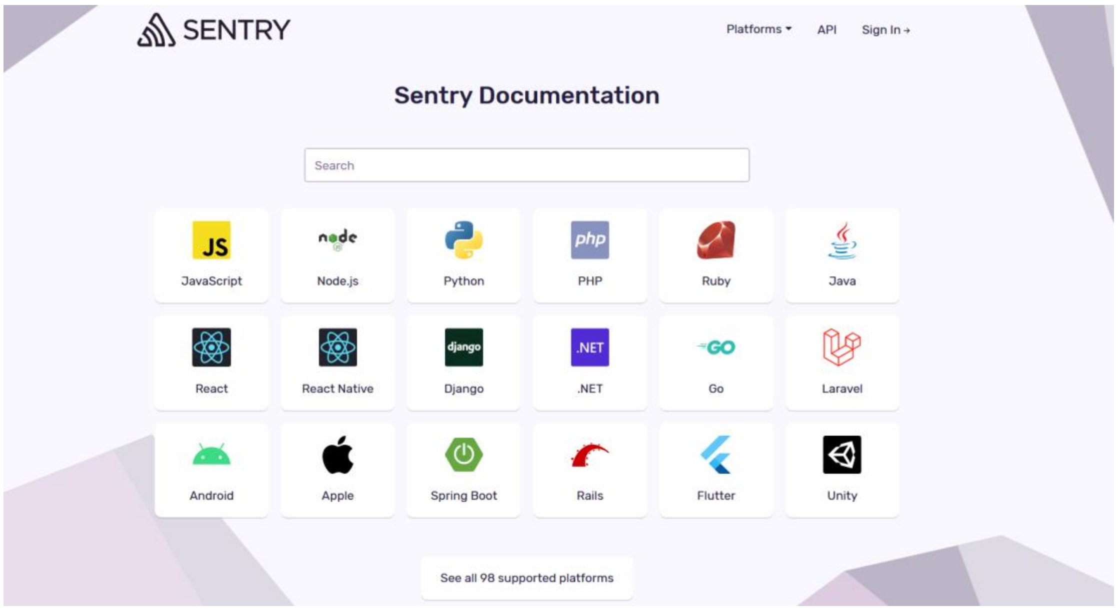


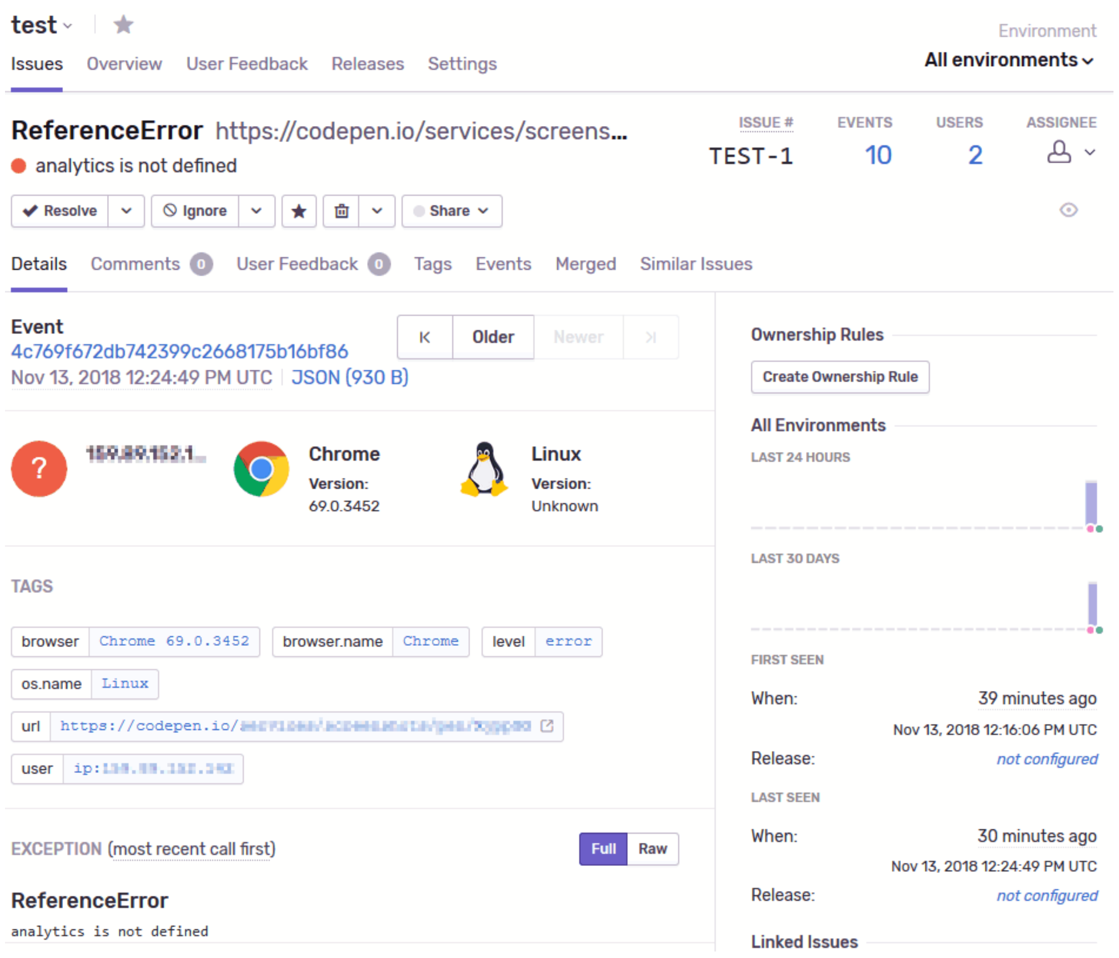
Figure 28: Supported Frameworks and Languages

As the above figure depicts (Figure 28), Sentry supports a wide variety of frameworks, such as Python, React, React Native, Django, Java, etc.

After the connection of Sentry with a component, when an error is raised (Figure 29), Sentry captures that error and provides a UI with the following contextual information:

- The error type and error message.
- The number of times that an error has been occurred.
- The number of unique users been affected by this error.
- The part of the code that raised the error.
- The steps of user that lead to the error.
- Metadata about the error.
- The commits might be responsible for the generation of the error.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158


test
Issues Overview User Feedback Releases Settings
Environment
All environments

ReferenceError https://codepen.io/services/screens...
analytics is not defined

ISSUE #	EVENTS	USERS	ASSIGNEE
TEST-1	10	2	

Resolve Ignore Share

Details Comments 0 User Feedback 0 Tags Events Merged Similar Issues

Event
4c769f672db742399c2668175b16bf86
Nov 13, 2018 12:24:49 PM UTC | JSON (930 B)

Older Newer

[redacted]
Chrome
Version:
69.0.3452
Linux
Version:
Unknown

TAGS

browser	Chrome 69.0.3452
browser.name	Chrome
level	error
os.name	Linux
url	https://codepen.io/[redacted]
user	ip:[redacted]

EXCEPTION (most recent call first)
Full Raw

ReferenceError
analytics is not defined

Ownership Rules
Create Ownership Rule

All Environments
LAST 24 HOURS
LAST 30 DAYS

FIRST SEEN
When: 39 minutes ago
Nov 13, 2018 12:16:06 PM UTC
Release: *not configured*

LAST SEEN
When: 30 minutes ago
Nov 13, 2018 12:24:49 PM UTC
Release: *not configured*

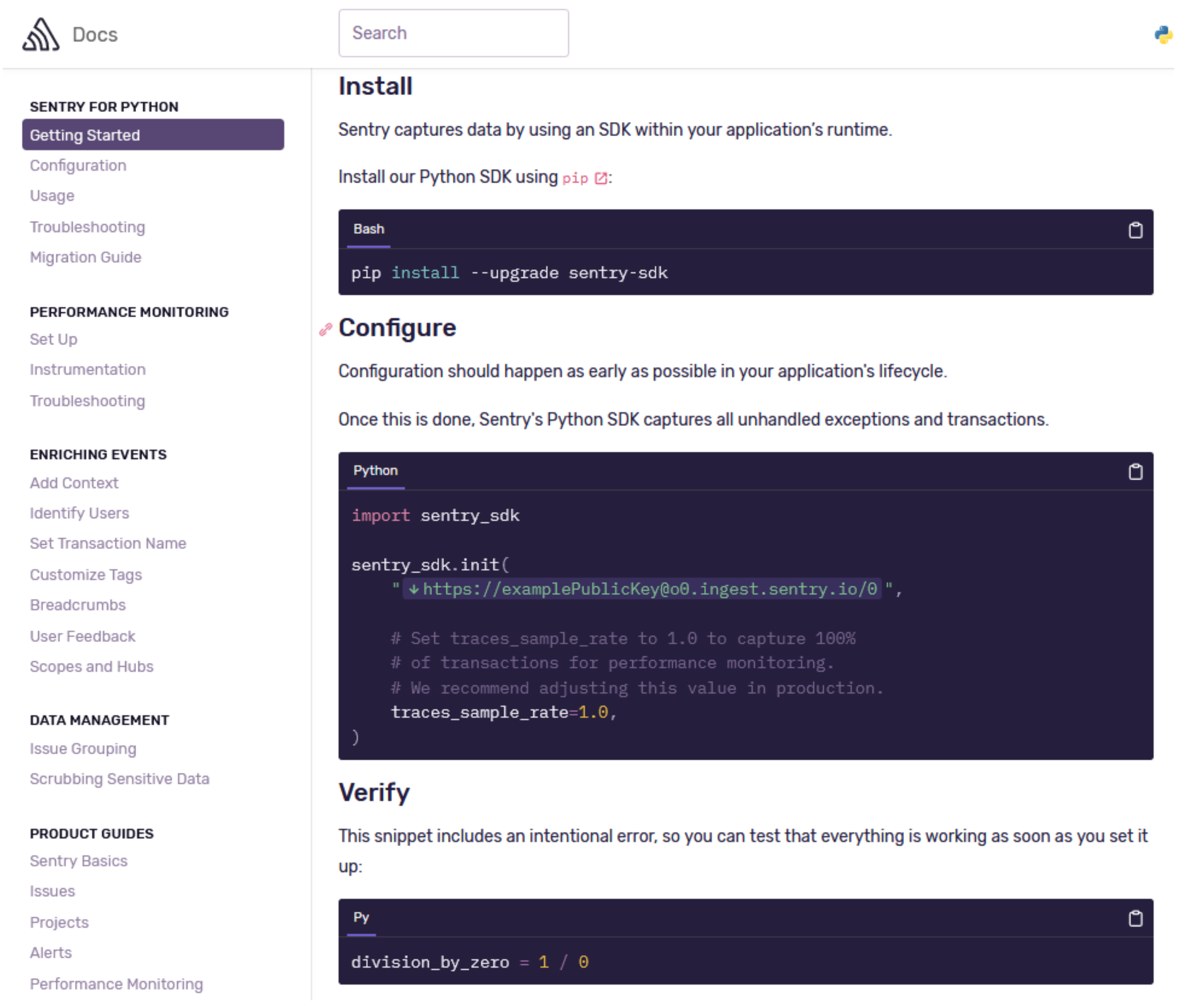
Linked Issues

Figure 29: Example of Error after integration testing with Sentry

In addition, Sentry can be connected with GitHub, Jira, Slack, GitLab, etc., facilitating the communication among the developers about the errors and bugs. Moreover, an error can be assigned to a developer and be marked as resolved after the developer has implemented a solution.

Sentry captures data by using an SDK within the application's runtime. The process of installing Sentry in Python or any other framework or technology is easy and well-documented. For instance, the installation in Python is taking place as depicted below (Figure 30). After the installation, the configuration should happen as early as possible in the application's runtime. Therefore, the following figure demonstrate the process that should be followed in order to integrate Sentry with a Python project for error tracking.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158



Install

Sentry captures data by using an SDK within your application's runtime.

Install our Python SDK using pip:

Bash

```
pip install --upgrade sentry-sdk
```

Configure

Configuration should happen as early as possible in your application's lifecycle.

Once this is done, Sentry's Python SDK captures all unhandled exceptions and transactions.

Python

```
import sentry_sdk

sentry_sdk.init(
    "https://examplePublicKey@o0.ingest.sentry.io/0",

    # Set traces_sample_rate to 1.0 to capture 100%
    # of transactions for performance monitoring.
    # We recommend adjusting this value in production.
    traces_sample_rate=1.0,
)
```

Verify

This snippet includes an intentional error, so you can test that everything is working as soon as you set it up:

Py

```
division_by_zero = 1 / 0
```

Figure 30: Installation of Sentry in Python project

Similarly, the integration of Sentry in a Python Project can be implemented as the figure above depicts (Figure 30).

Redmine

Redmine is a flexible open-source project management and issue tracking web tool. Written using the Ruby on Rails[40] framework, it is cross platform and cross database system. Redmine is open source and released under the terms of the GNU General Public License v2[41].

Redmine provides a lot of functionalities such as project support, issue tracking wiki, SCM integration, forums, news, feeds, etc. It also possible to install plugins to integrate with other tools like Jenkins and for Agile Management. Redmine provides a list of plugins to integrate with other tools. One interesting benefit from using Redmine is that as Project Management tool it is possible to gather all the results coming from other tools such as Jenkins and Sonar using add-on plugins.

Redmine allows users to manage and monitor multiple projects and associated sub-projects. Also, there are wikis, forums, time tracking and flexible role-based access control per project.

[40] https://rubyonrails.org/
[41] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

Redmine has been successfully installed in NTUA premises and it can be accessed at the following address: http://MATRYCS.epu.ntua.gr:3000/.

The following figure (Figure 31) demonstrates the last activities on Redmine

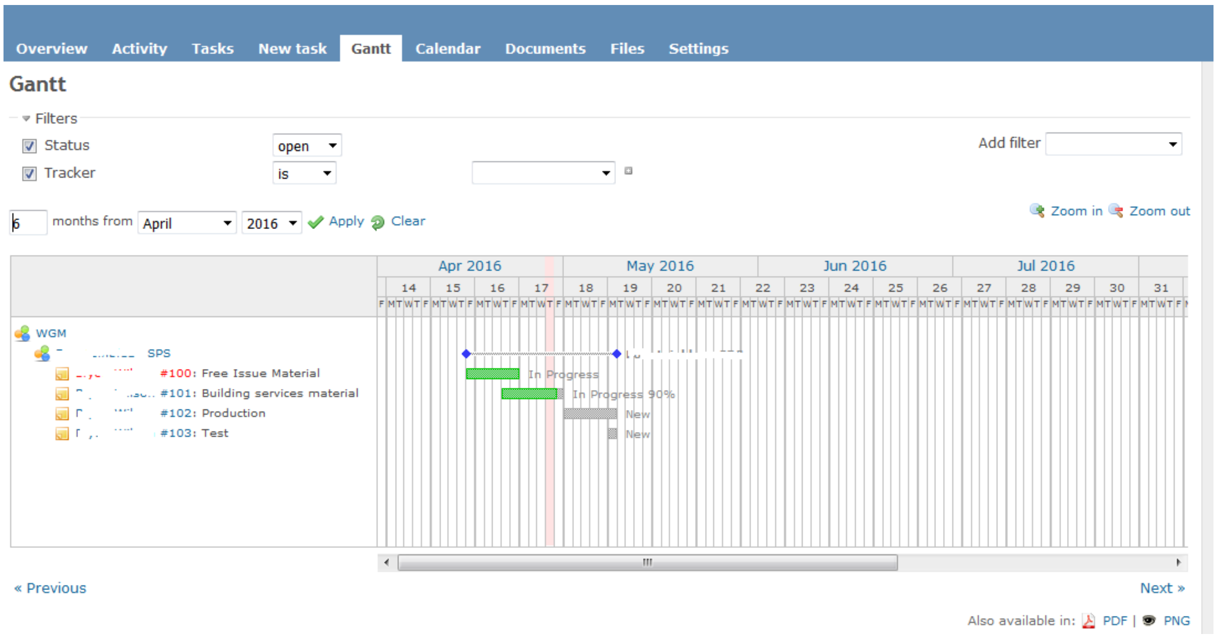


Figure 31: Redmine's Gantt diagram for issues and bugs

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

7 MATRYCS Toolbox Integration on M11

The MATRYCS Toolbox is mainly composed by three main services of Cloud Computing: SaaS, PaaS and IaaS. The code of the MATRYCS Analytics and Open Cloud based Data Analytics Toolbox 1st technology is continuously committed on this Github Repository https://github.com/Matrycs.

The **Analytics Building Applications** together with **Visualisations and Reports Engine** and **Digital Building Twins** are the three pillars of the SaaS in the MATRYCS Architecture. The integration of these services with the MATRYCS-PROCESSING and MATRYC-GOVERNANCE (PaaS) is summarized in the Figure 32.

The security framework handles the authentication and access of different users to a common platform where all services will be available. The security framework will guarantee the possibility of using different analytical services according to the data provided and complying with the EC regulations on Data Protection (GDPR).

Each single service uses the data or analysis of MATRYCS-PROCESSING and MATRYC-GOVERNANCE in three main ways:

- Direct access to building database.
- Connection to specific frameworks where different ML algorithms can run generating data and analysis to be shown in a dedicated front-end.
- Using the reasoning engine.

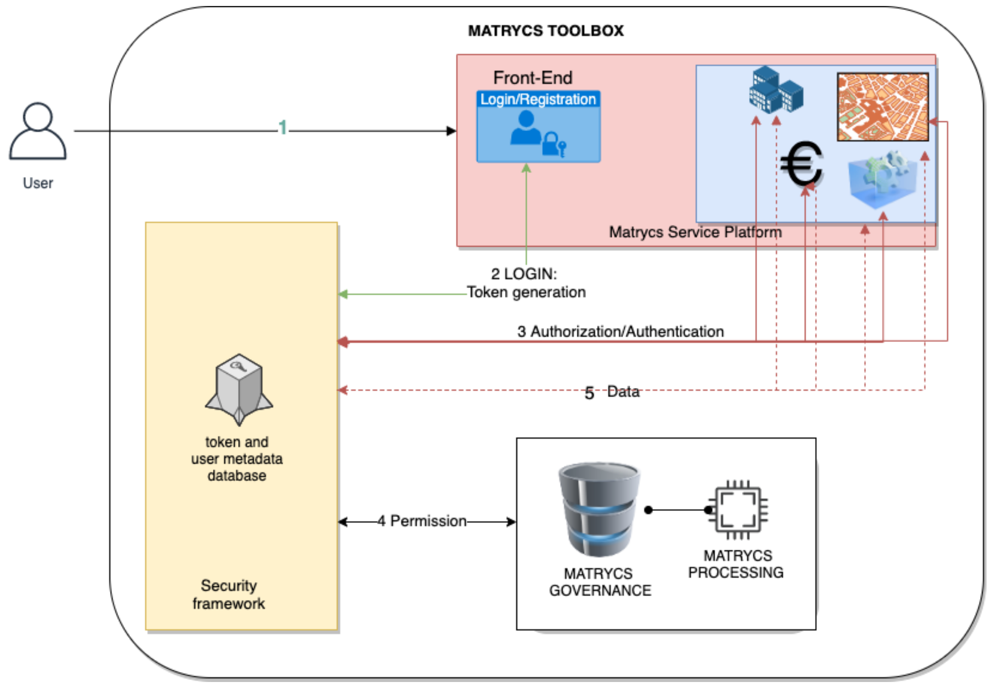


Figure 32: MATRYCS Toolbox Components Interconnections

According to the type of service one of the three types of connection will be set-up. For example, considering the development of the geoclustering tool, the application of a clustering algorithm will be referred by a specific serving framework (BENTO ML) running inside the MATRYCS-PROCESSING. In all cases, an API will be provided to manage the exchange of information between the analytical services and the MATRYCS-PROCESSING and MATRYC-GOVERNANCE.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

8 Future Steps

As the 1st technology release has been described and analysed for the MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox, the main technologies that were used and the limitations that have arisen are well understood and lead to the future activities for an efficient and successful 2nd technology release.

Below, the future activities are listed:

MATRYCS Toolbox Software as a Service Layer

- Visualisation Engine extension.
- Addition of more datasets to MATRYCS Storage.
- User authorisation implementation.
- User profile mock-up and implementation. This step will include features analysis and the user profile's page will be provided (stored queries, datasets uploading for model training, etc.).
- Integration with the End-to-End security framework.
- Development of each analytical service so as to run independently.
- Based on the results of the first iteration of the created Digital Twins, a final decision will be made considering the usage of various tools (Revit, Archicad, BricsCAD BIM, etc) and their encapsulation in the MATRYCS Framework.
- Identify available data in each LSP.
- Evaluate CityGML requirements and integration methods for different datasets.
- Analysis finalization of the available datasets.
- Decide the data model to use for digital building twin.

MATRYCS Toolbox Software as a Platform Layer

- Integration with the End-to-End Security Framework.
- Integration with the other MATRYCS-PROCESSING components (such as Data Feed Module, Model Development Module, Serving & Evaluation Framework).
- Proceed from MongoDB testing phase to fully integration of Virtual Workbench with MongoDB.

MATRYCS Toolbox Infrastructure as a Service Layer

- Migration to EGI-Ace cloud (once positively evaluated by EGI ACE project).

Continuous Integration and Testing

- Proceed to one-to-one meetings with Task leaders (T4.1, T4.2, T4.3, T4.4, T4.5, T5.1, T5.7) in order to adapt MATRYCS Framework components.
- Bilateral meetings for the functional testing of the components.
- Evaluate the testing plan for the MATRYCS Framework 2nd technology release.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

The 2nd technology release of the MATRYCS Toolbox will be available on July 2022 and described in detail in D4.5 - *MATRYCS-ANALYTICS and Open Cloud-based Data Analytics Toolbox (2nd technology release)*.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158

References

[1] G. Kulkarni, "Cloud Computing-Software as Service-1." Accessed: Jul. 08, 2021. [Online]. Available: https://www.academia.edu/6172744/Cloud_Computing_Software_as_Service_1.

[2] "A Complete Guide to SaaS - DZone Cloud." https://dzone.com/articles/what-is-saas-a-complete-guide (accessed Jul. 08, 2021).

[3] C. C. Chou, C. T. Chiang, P. Y. Wu, C. P. Chu, and C. Y. Lin, "Spatiotemporal analysis and visualization of power consumption data integrated with building information models for energy savings," *Resour. Conserv. Recycl.*, vol. 123, pp. 219–229, Aug. 2017, doi: 10.1016/J.RESCONREC.2016.03.008.

[4] N. A. Sulaiman, M. Kassim, and S. Saaidin, "Systematic Test and Evaluation Process (STEP) approach on Shared Banking Services (SBS) System identification," *ICETC 2010 - 2010 2nd Int. Conf. Educ. Technol. Comput.*, vol. 5, 2010, doi: 10.1109/ICETC.2010.5529778.

[5] A. Dwaraki, S. Seetharaman, S. Natarajan, and T. Wolf, "GitFlow: Flow revision management for software-defined networks," *Symp. Softw. Defin. Netw. Res. SOSR 2015*, Jun. 2015, doi: 10.1145/2774993.2775064.

[6] P. Runeson, "A survey of unit testing practices," *IEEE Softw.*, vol. 23, no. 4, pp. 22–29, 2006, doi: 10.1109/MS.2006.91.

[7] W. T. Tsai, X. Bai, R. Paul, W. Shao, and V. Agarwal, "End-to-end integration testing design," *Proc. - IEEE Comput. Soc. Int. Comput. Softw. Appl. Conf.*, pp. 166–171, 2001, doi: 10.1109/CMPSAC.2001.960613.

[8] "Integration Testing: What is, Types, Top Down & Bottom Up Example." https://www.guru99.com/integration-testing.html (accessed Jul. 08, 2021).

[9] F. D. Davis and V. Venkatesh, "Toward preprototype user acceptance testing of new information systems: Implications for software project management," *IEEE Trans. Eng. Manag.*, vol. 51, no. 1, pp. 31–46, Feb. 2004, doi: 10.1109/TEM.2003.822468.

[10] S. E. Chodrow and M. G. Gouda, "The Sentry System," pp. 230–237, 1994.

The MATRYCS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no.101000158