BD4NRG
Big Data for Next Generation Energy

D5.2:

Marketplace 1st Technology Release

WP5 – Marketplaces for realizing EU-level Energy Data Economy

Month 12

www.bd4nrg.eu

Disclaimer

Copyright Message

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613

Grant Agreement Number: 872613 Acronym: BD4NRG

Full Title	Big Data for Next Generation Energy		
Topic	DT-ICT-11-2019 Big data solutions for energy		
Funding scheme	H2020- IA: Innovation Action		
Start Date	January 2021	Duration	36
Project URL	http//www.bd4nrg.eu		
Project Coordinator	ENG		
Deliverable	D5.2 BD4NRG-MARKETPLACE - First Technology Release		
Work Package	WP5 Marketplaces for realizing EU-level Energy Data Economy		
Delivery Month (DoA)	10	Version	1.0
Actual Delivery Date	31/12/2021 (M12)		
Nature	Other	Dissemination Level	Public
Lead Beneficiary	Atos		
Authors	Elisa Herrmann [ATOS], Arturo Medela, [ATOS] Alexander Pastor [RWTH] Gabriel Iuhasz [TS], Caterina Sarno [ENG]		
Quality Reviewer(s):	Marzia Mammina and Caterina Sarno [ENG] Francesco Bellesini [EMOT]		
Keywords	BD4NRG, Marketplace, Functional Specification, DLT, Utilities, Sandboxes, VILLAS, MLaaS, Scenarios		

Preface

BD4NRG focuses on addressing emerging challenges in big data management for energy sector with an innovative open holistic solution for smart grid-tailored, near real time, energy-specific and AI-based open Big Data Analytics modular framework. The vision is to deliver holistic services for techno-economic optimal management of Electric Power and Energy Systems value chain. Services range from optimal risk assessment for energy efficiency investments planning, to optimized management of grid and non-grid owned assets, improved efficiency, and reliability of electricity networks operation, while at the same time contributing to achieve fair energy prices to the consumers and laying the foundations for an EU-level energy-tailored data sharing economy. The BD4NRG Toolbox will be implemented and validated in real-life pilots in 12 large-scale demo sites across 10 countries for:

- Increasing the efficiency and reliability of the electricity network **BD-4-NET**
- Optimising the management of assets connected to the grid **BD-4-DER**
- De-risking investments in energy efficiency and increasing the efficiency and comfort of buildings **BD-4-ENEF**

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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	RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN	**RWTH**	DE	
4	EUROPEAN DYNAMICS LUXEMBOURG SA	**ED**	LU	
5	INTERNATIONAL DATA SPACES EV	**IDSA**	DE	
6	EUROPEAN NETWORK OF TRANSMISSION SYSTEM OPERATORS FOR ELECTRICITY AISBL	**ENTSO-E**	BE	
7	PANEPISTIMIO DYTIKIS ATTIKIS	**UNIWA**	GR	
8	ATOS SPAIN SA	**ATOS**	ES	
9	FUNDACION CARTIF	**CARTIF**	ES	
10	UNIVERZA V LJUBLJANI	**UNILJ**	SL	
11	ENEL X SRL	**ENELX**	IT	
12	REN - REDE ELECTRICA NACIONAL SA	**REN**	PT	
13	CENTRO DE INVESTIGACAO EM ENERGIA REN - STATE GRID SA	**RDN**	PT	
14	UNINOVA-INSTITUTO DE DESENVOLVIMENTO DE NOVAS TECNOLOGIASASSOCIACAO	**UNINOVA**	PT	
15	ENERCOUTIM - ASSOCIACAO EMPRESARIALDE ENERGIA SOLAR DE ALCOUTIM	**ENERC**	PT	
16	FIWARE FOUNDATION EV	**FIWARE**	DE	
17	CENTRICA BUSINESS SOLUTIONS BELGIUM	**CENTRICA**	BE	
18	NEDERLANDSE ORGANISATIE VOOR TOEGEPAST NATUURWETENSCHAPPELIJK ONDERZOEK TNO	**TNO**	NL	
19	ASM TERNI SPA	**ASM**	IT	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

	Participant Name	Short Name	Country Code	Logo
20	VIDES INVESTICIJU FONDS SIA	**LEIF**	LV	
21	COMSENSUS, KOMUNIKACIJE IN SENZORIKA, DOO	**COMSENSUS**	SL	
22	HOLISTIC IKE	**HOLISTIC**	GR	
23	INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM	**IMEC**	BE	
24	TERRASIGNA SRL	**TS**	RO	
25	UBIMET GMBH	**UBIMET**	AT	
26	ELEKTRO LJUBLJANA PODJETJE ZADISTRIBUCIJO ELEKTRICNE ENERGIJE D.D.	**EKL**	SL	
27	BORZEN, OPERATER TRGA Z ELEKTRIKO, D.O.O.	**BORZEN**	SL	
28	AJUNTAMIENTO DE SANT CUGAT DEL VALLES	**AJSCV**	ES	
29	ELES, D.O.O., SISTEMSKI OPERATER PRENOSNEGA ELEKTROENERGETSKEGA OMREŽJA	**ELES**	SL	
30	E-LEX - STUDIO LEGALE	**ELEX**	IT	
31	OSMANGAZI ELEKTRIK DAGITIM ANONIM SIRKETI	**OEDAS**	TR	
32	VEOLIA SERVICIOS LECAM SOCIEDAD ANONIMA UNIPERSONAL	**VEOLIA**	ES	
33	STICHTING EG	**EGI**	NL	
34	CINTECH SOLUTIONS LTD	**CN**	CY	
35	EMOTION SRL	**EMOT**	IT	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Contents

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Figures

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Tables

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Executive Summary

The work described in this document (D5.2) was carried out in the framework of Work Package (WP5) – "Marketplaces for realizing EU-level Energy Data Economy". This report goes hand by hand with the initial iteration of the software solutions that shape the BD4NRG Marketplace; thus, it reflects the functional description of the structure and functionality of certain components relevant to build this Marketplace.

This report includes an initial view on the BD4NRG Marketplace that includes the services exposed, the resource catalogue offered, and the access layer to components that implements the actual services and resources. Afterwards, it delves into certain specifics related to Distributed Ledger Technology (DLT) and Smart Contracts and how they fit into the overall BD4NRG framework and also presents the first technology release of the Utilities Sandbox/SIMaaS (Simulation as a Service) module that will allow utilities and power network operators to access model-based simulated data on Hardware-In-the-Loop facilities. Finally, a view on the Machine Learning as a Service modules implementation is offered as well.

The process followed to comply with this objective organically involved partners not only involved in WP5 but also in technical work packages (namely WP3 – "Data Governance and Management" and WP4 – "Data Modelling & Open Modular AI-based edge-level Analytics Toolbox"), since there is a relevant impact of the modules developed there into the Marketplace. All in all, to support the integration activities which are a responsibility of WP2 – "System Requirements and Specifications", while following the common Architecture framework that will be definitely set in D2.6.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Acronyms and Abbreviations

Acronym	Description
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CI/CD	Continuous Integration / Continuous Deployment
DaaS	Data-as-a-Service
DAP	Data Access Policies
DB	DataBase
ERC	Ethereum Request for Comments
ETH	Ether
FPGA	Field Programmable Gate Array
FR	Functional Requirements
GPGU	General-Purpose Graphics Processing Unit
GPL	GNU General Public License
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol
IC	Infrastructure Components
IdM	Identity Manager
JSON	JavaScript Object Notation
ML	Machine Learning
MLaaS	Machine Learning as a Service
MoSCoW	M - Must have S - Should have C - Could have W - Won't have
MQTT	Message Queuing Telemetry Transport

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

MVC	Model/View/Controller
NFR	Non Functional Requirements
OIDC	OpenId Connect
P2P	Peer-To-Peer
QoS	Quality of Service
REST	REpresentational State Transfer
SIMaaS	Simulation as a Service
SPA	Single-Page Applications
SSL	Secure Secure Socket Layer
TLS	Transport Layer Security
UDP	User Datagram Protocol
UI	User Interface
VPN	Virtual Private Network
WP	Work Package

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

1 Introduction

In reference to the BD4NRG global solution, the WP5 covers the specification, implementation, and integration of the components that support the BD4NRG Marketplace. With this, WP5 will provide the infrastructure to offer stakeholders of the energy sector, tools, resources, and services as tradable assets of the BD4NRG platform. The Marketplace is the visible environment and entry point to the utilities and features that BD4NRG provides, not only in the context of WP5 but also in WP3 and WP4.

In this context, deliverable D5.2 covers a short description of the First Technology Release of the BD4NRG Marketplace that integrates BD4NRG resources, services, and tools. The following figure depicts the Marketplace architecture including the WP3 and WP4 components that expose their functionality and resources through the Marketplace.

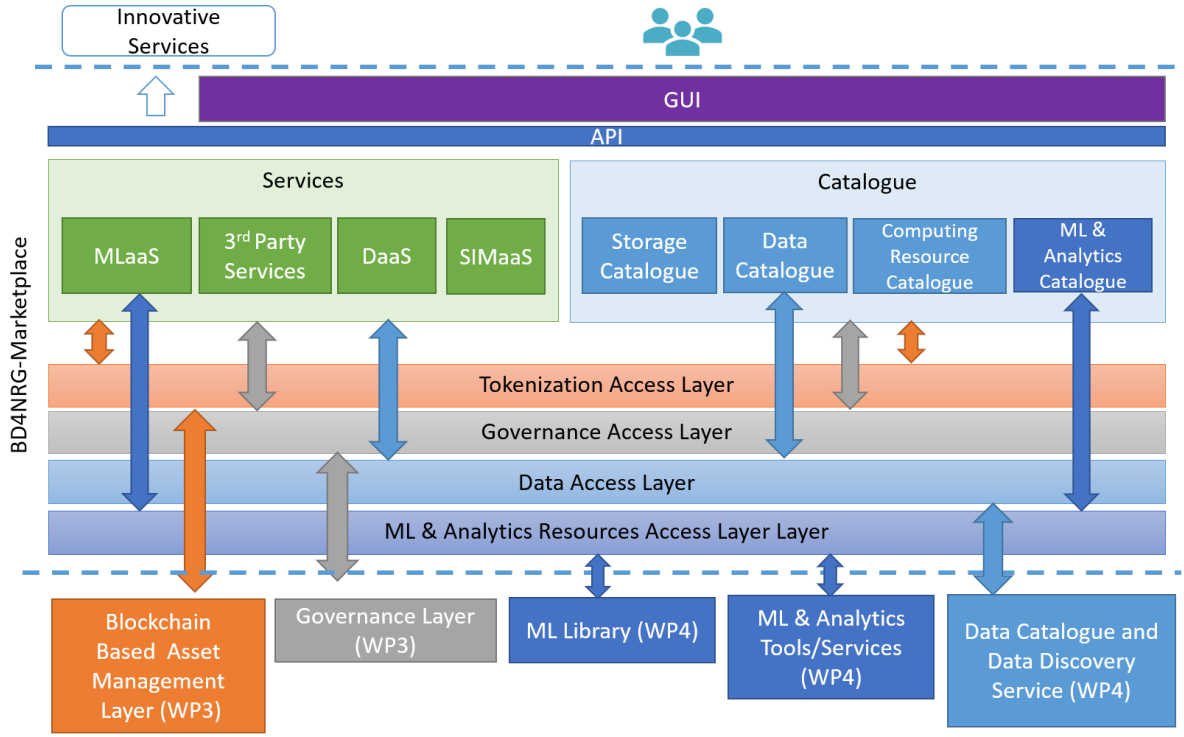

Figure 1: BD4NRG- Marketplace Architecture.

This D5.2 document comprises the following chapters:

- **Chapter 2** shorty presents first technology release of the BD4NRG-Marketplace that includes the services exposed, the resource catalogue offered, and the access layer to components that implements the actual services and resources. Also, the Marketplace includes the Open Application Programming Interface (API) that exposes the main functionality and the Graphic User Interface (GUI) to allow users to interact with the Marketplace through a web application.
- **Chapter 3** discusses how trustable and secure mechanisms for digital assets monetisation will be implemented relying on smart contracts, exploiting the distributed ledger technology offered by the blockchain platform provided on WP3.
- **Chapter** 4 presents the first technology release of the Utilities Sandbox/SIMaaS (Simulation as a Service) module that will allow utilities and power network operators to access model-based simulated data on Hardware-In-the-Loop facilities. The module functionality will be integrated

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

in the Marketplace by means of the API and the Marketplace will provide the GUI to its functionality.

- **Chapter 5** briefly introduces the MLaaS – ML/DL Library component that will provide pre-trained ML models that may be deployed on demand.

Therewith, this document is targeting at a short description of the first technology release of the BD4NRG-Marketplace key ecosystem elements.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2 BD4NRG-Marketplace

2.1 Architecture

The BD4NRG- Marketplace is conceived an entry point to energy stakeholders to BD4NRG functionality and resources. The architecture of the Marketplace is conceived as a web application that integrates the functionality of a set of loosely coupled, collaborating services and tools. The services and tools provided by WP3, WP4 and WP5 are developed and deployed as independent services and expose their functionality through a REST API. The BD4NRG Marketplace provides a unified GUI and a component that coordinates the interaction with the services and tools. Also, the Marketplace expose their own API to allow machine to machine interaction among application and innovative services with the BD4NRG environment. The following Figure 2 shows the diagram components, the interfaces, and connections between them. From the GUI represented as a web application to the services like the Machine Learning as a Service (MLaaS), SIMaaS, 3rd party services and Data-as-a-Service (DaaS) that will be part of the Marketplace, going through the so-called Catalogue Module which comprises the Data Catalogue and Discovery, the Storage Catalogue, the Computing Resource Catalogue, and the ML and Analytics Catalogue. On the other hand, the Service Integration Layer includes the functionality to access services and tools provided by BD4NRG framework; they are the Tokenization Access Layer, the Governance Access Layer, the Data Access Layer and Discovery Access Layer, and the ML and Analytics Resources Access Layer.

The functionality identified in this first technology release will be extended and refined in the later technology releases, according. to the progress in the definition and specification of the different tools and services developed in the other work packages and considering the requirements and specifications of the pilots.

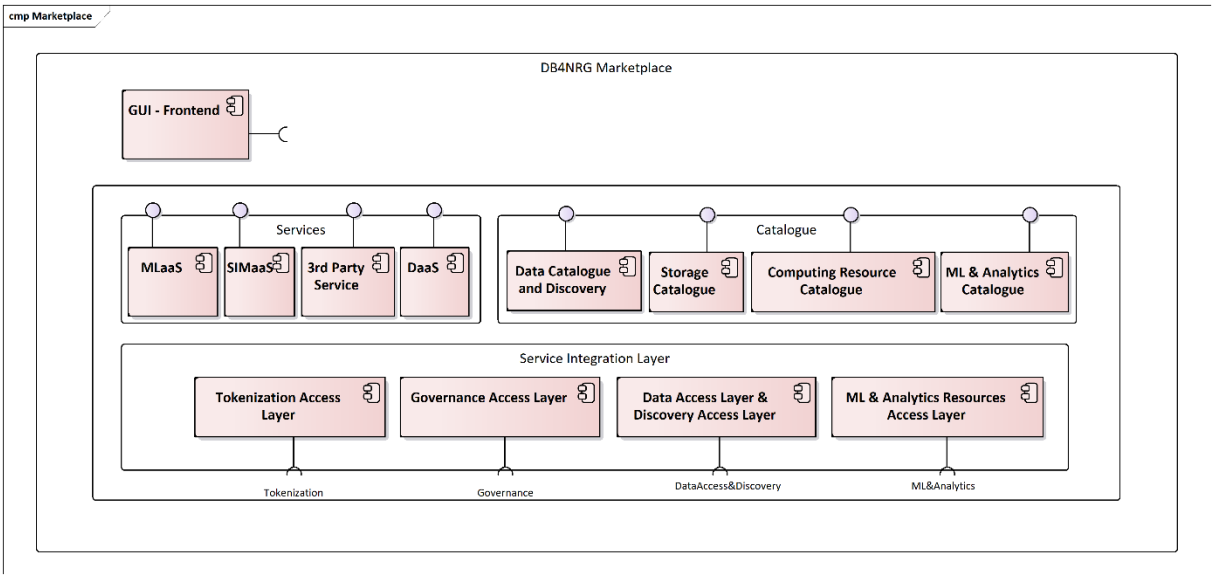

Figure 2: Marketplace Component Diagram.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.2 Implementation and Deployment Technologies

2.2.1 BD4NRG- Marketplace GUI - Web Application

The BD4NRG frontend application is implemented using the Angular[1] framework to build dynamic and single-page applications (SPAs), and to do so uses Hypertext Markup Language (HTML) and TypeScript[2]. The application is based on the frontend of Agora[3] Marketplace reusing some of the tools and technologies.

The engineering of an Angular application depends on Angular components that gather associated functional sets of code, coordinated into modules. The code aims to be modular, reusable, and efficient, and thus service providers can be inserted into components as dependencies.

The application generates dynamic web pages using data services exposed by the BD4NRG-Marketplace core component. It uses Angular 8.0 and the Nebular customizable User Interface (UI) library. The library *Angular-oauth2-oidc* provides support for OAuth 2 and OpenId Connect (OIDC) that eases the implementation of login implicit flows. The following table shows the list of dependencies.

Table 1: Web Application Library List. (Package.json).

```
{
  "name": "bd4nrg_frontend",
  "version": "1.0.0",
  "description": "bd4nrg Marketplace Application",
  "author": "Atos",
  "homepage": " https://www.bd4nrg.eu/”
  "license": "SEE LICENSE IN LICENSE",
  "scripts": {
    "ng": "ng",
    "config": "ts-node set-env.ts",
    "prestart": "npm run config",
     "start": "node --max_old_space_size=4096 ./node_modules/@angular/cli/bin/ng serve --
watch",
    "prebuild": "npm run config",
    "build": "node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng build",
    "build:prod": "node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng build --
prod --aot",
    "test": "ng test",
    "test:coverage": "rimraf coverage && npm run test -- --code-coverage",
    "lint": "ng lint",
    "lint:fix": "ng lint ngx-admin-demo --fix",
    "lint:styles": "stylelint ./src/**/*.scss",
    "lint:ci": "npm run lint && npm run lint:styles",
    "pree2e": "webdriver-manager update --standalone false --gecko false",
    "e2e": "ng e2e"
  },
  "dependencies": {
    "@agm/core": "^1.0.0-beta.5",
    "@angular-builders/dev-server": "7.3.1",
    "@angular/animations": "^8.0.0",
    "@angular/cdk": "^8.0.0",
    "@angular/common": "^8.0.0",
    "@angular/compiler": "^8.0.0",
```

[1] https://angular.io/
[2] https://www.typescriptlang.org/
[3] https://booklet.atosresearch.eu/agora

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

```
    "@angular/core": "^8.0.0",
    "@angular/forms": "^8.0.0",
    "@angular/platform-browser": "^8.0.0",
    "@angular/platform-browser-dynamic": "^8.0.0",
    "@angular/router": "^8.0.0",
    "@asymmetrik/ngx-leaflet": "3.0.1",
    "@nebular/auth": "^4.4.0",
    "@nebular/eva-icons": "^4.4.0",
    "@nebular/security": "^4.4.0",
    "@nebular/theme": "^4.4.0",
    "@swimlane/ngx-charts": "^10.0.0",
    "@types/jquery": "^3.3.30",
    "angular-oauth2-oidc": "^8.0.4",
    "angular-tree-component": "7.2.0",
    "angular2-chartjs": "0.4.1",
    "angular2-toaster": "^7.0.0",
    "bootstrap": "4.3.1",
    "chart.js": "^2.8.0",
    "ckeditor": "4.7.3",
    "classlist.js": "1.1.20150312",
    "core-js": "2.5.1",
    "echarts": "^4.0.2",
    "eva-icons": "^1.1.0",
    "font-awesome": "^4.7.0",
    "heatmap.js": "^2.0.5",
    "intl": "1.2.5",
    "ionicons": "2.0.1",
    "jquery": "^3.4.1",
    "leaflet": "1.2.0",
    "leaflet-heatmap": "^1.0.0",
    "lodash": "^4.17.15",
    "moment": "^2.24.0",
    "nebular-icons": "1.1.0",
    "ng2-ckeditor": "^1.2.2",
    "ng2-completer": "2.0.8",
    "ng2-smart-table": "1.3.5",
    "ngx-echarts": "^4.0.1",
    "node-sass": "^4.13.1",
    "normalize.css": "6.0.0",
    "pace-js": "1.0.2",
    "roboto-fontface": "0.8.0",
    "rxjs": "6.5.2",
    "rxjs-compat": "6.3.0",
    "socicon": "3.0.5",
    "tinymce": "4.5.7",
    "tslib": "^1.9.0",
    "typeface-exo": "0.0.22",
    "web-animations-js": "github:angular/web-animations-js#release_pr208",
    "zone.js": "~0.9.1"
  },
  "devDependencies": {
    "@angular-builders/custom-webpack": "8.1.0",
    "@angular-devkit/build-angular": "0.803.24",
    "@angular/cli": "^8.0.2",
    "@angular/compiler-cli": "^8.0.0",
    "@angular/language-service": "8.0.0",
    "@compodoc/compodoc": "^1.1.10",
    "@fortawesome/fontawesome-free": "^5.2.0",
    "@types/d3-color": "1.0.5",
    "@types/echarts": "^4.1.9",
    "@types/jasmine": "2.5.54",
    "@types/jasminewd2": "2.0.3",
    "@types/leaflet": "1.2.3",
```


BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

```
    "@types/node": "6.0.90",
    "codelyzer": "^5.0.1",
    "conventional-changelog-cli": "1.3.4",
    "copy-webpack-plugin": "^5.1.1",
    "file-loader": "^4.1.0",
    "html-elements-webpack-plugin": "^2.0.0",
    "html-webpack-plugin": "^3.2.0",
    "husky": "0.13.3",
    "jasmine-core": "2.6.4",
    "jasmine-spec-reporter": "4.1.1",
    "json-loader": "^0.5.7",
    "karma": "1.7.1",
    "karma-chrome-launcher": "2.1.1",
    "karma-cli": "1.0.1",
    "karma-coverage-istanbul-reporter": "1.3.0",
    "karma-jasmine": "1.1.0",
    "karma-jasmine-html-reporter": "0.2.2",
    "npm-run-all": "4.0.2",
    "protractor": "5.1.2",
    "rimraf": "2.6.1",
    "stylelint": "7.13.0",
    "ts-node": "3.2.2",
    "tslint": "^5.7.0",
    "tslint-language-service": "^0.9.9",
    "typescript": "3.4.5",
    "url-loader": "^2.1.0"
  }
}
```

2.2.2 BD4NRG-Marketplace Core Component and REST API

The core components of the BD4NRG-Marketplace and the API have been implemented using the Sails[4] framework version 1.4.

The sails framework is built on Node.js and it is a lightweight server-side technology that permits developers to prepare scalable network applications in JavaScript and utilizes Express[5] for handling HTTP requests and wraps socket.io for managing WebSockets. It allows to create Model/View/Controller (MVC) node.js applications and to support data-driven APIs with a scalable, service-oriented architecture.

The sails framework provides “models” to represent a set of structured data. A “controller” contains public methods called “action” methods which are charted to the endpoints that lever incoming requests, retrieves required data from the models, and gives back suitable responses. The so-called services are stateless libraries of functions that may be used from anywhere like controller actions, from inside other services, or in custom model methods. These services can be employed to retrieve data from a third-party API like the Governance Services, the Identity Management services, the Tokenization services, the Data Access and Discovery Services, and the ML and Analytics services. These services utilize the libraries created through the OpenAPI[6] specification of the actual services.

[4] https://sailsjs.com/
[5] https://expressjs.com/
[6] https://swagger.io/specification/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Policies let to limit certain activities to logged-in users depending on role permissions and reject undesirable access.

The application "models" information is stored in a mongoDB[7] database, attributes correspond to fields, and records correspond to document using the sails-mongo library (version 1.2).

The implementation includes the Mocha[8] and Chai[9] library for unit and integration testing, and nyc[10] for coverage tests. The following table shows the list of dependencies used in the Marketplace core component.

Table 2: Marketplace core component Library List. (Package.json).

```
{
  "name": "bd4nrg_backend",
  "private": true,
  "version": "0.0.1",
  "description": "bd4nrg API",
  "keywords": [],
  "dependencies": {
    "@sailshq/lodash": "^3.10.3",
    "bcrypt": "5.0.1",
    "crypto-js": "4.0.0",
    "ejs": "2.3.4",
    "expect": "26.6.2",
    "generate-password": "1.6.0",
    "idm_api": "file:lib/idm_api",
    "include-all": "^4.0.3",
    "is-coordinates": "^2.0.2",
    "mime-types": "^2.1.30",
    "mongodb": "^3.6.5",
    "nested-pop": "^0.1.4",
    "ngeohash": "0.6.3",
    "node-forge": "0.10.0",
    "object-sizeof": "^1.2.0",
    "sails": "^1.4.0",
    "sails-hook-apianalytics": "^2.0.3",
    "sails-hook-email": "^0.12.1",
    "sails-hook-organics": "^2.0.0",
    "sails-hook-orm": "^3.0.2",
    "sails-hook-sockets": "^2.0.0",
    "sails-hook-swagger-generator": "^3.3.0",
    "sails-mongo": "1.2.0",
    "sails-service-mailer": "^1.1.0",
    "should": "^13.2.3",
    "skipper": "^0.9.0-4",
    "superagent": "^6.1.0",
    "uuid": "^3.4.0",
    "validator": "^13.5.2",
    "winston": "^3.3.3"
  },
  "scripts": {
    "debug": "node debug app.js",
    "start": "node app.js",
      "test":    "nyc    --reporter=html    --reporter=text    node_modules/mocha/bin/_mocha
test/bootstrap.test.js test/**/**/*.test.js"
```

[7] https://www.mongodb.com/
[8] https://mochajs.org/
[9] https://www.chaijs.com/
[10] https://istanbul.js.org/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

```
  },
  "main": "app.js",
  "repository": {
    "type": "git",
    "url": "git://scm.atosresearch.eu/ari/bd4nrg-backend.git"
  },
  "author": "a601149",
  "license": "",
  "devDependencies": {
    "@sailshq/upgrade": "1.0.9",
    "eslint": "7.24.0",
    "grunt": "1.3.0",
    "htmlhint": "0.14.2",
    "lesshint": "6.3.7",
    "mocha": "7.4.0",
    "chai": "4.0.0",
    "nyc": "10.1.0",
    "sails-hook-grunt": "4.0.1"
  }
}
```

2.2.3 Deployment

The Marketplace is developed using agile methodologies and DevOps practices implementing continuous integration/continuous deployment (CI/CD) to build, test and deploy features automatically with almost no manual intervention and more frequently.

The CI/CD implementation is achieved using various tools, frameworks, and systems like Gitlab[11] for the code repository and version control based in git, gitLab-CI[12] as the continuous integration tool built-in GitLab, Docker[13] for packaging and distributing the applications, NexusOSS[14] for storing docker images, Kubernetes[15] to deploy the applications in the Kubernetes clusters and Rancher[16] form managing and operating the clusters and applications. The tools cover the entire CI/CD process including the planning, coding, building, testing, releasing, deploying, operating, and measuring of the software (Figure 3).

[11] https://about.gitlab.com/
[12] https://docs.gitlab.com/ee/ci/
[13] https://www.docker.com/
[14] https://www.sonatype.com/products/repository-oss
[15] https://kubernetes.io/
[16] https://rancher.com/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

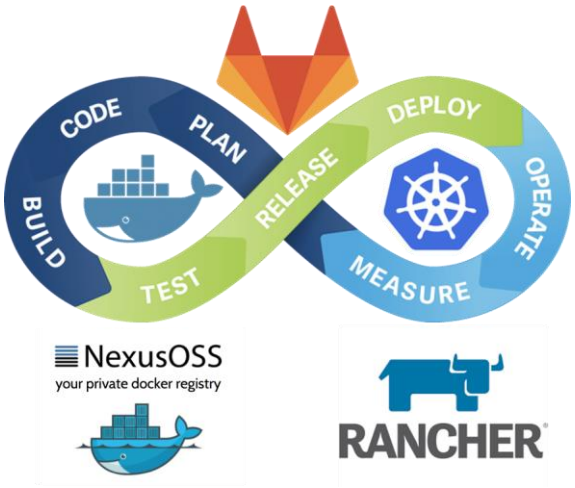

Figure 3: CI/CD process and Tools.

The GitLab repositories that supports the Marketplace core components are:

- Backend [1]
- Frontend [2]
- Environment [3]

The backend component repository contains the Marketplace core component and the REST API, the frontend repository contains the BD4NRG-Marketplace Web Application, the environment repository stores the configuration of the Kubernetes services, workloads, and storage resources.

The backend and the web application repositories have a pipeline with stages: *build*, *style_check*, *test*, *test_clean* and *Deploy* (Figure 4).

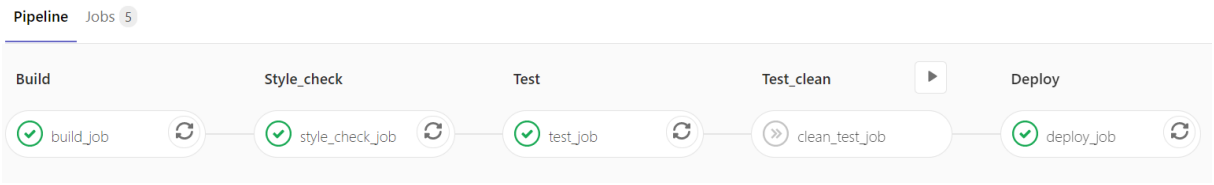

Figure 4: Pipeline Stages of the BD4NRG-Marketplace Components.

The backend repository contains a docker file that packs up the backend component to run the application as follows:

```
ROM node:15.3.0
WORKDIR /backend
RUN npm -g install sails
RUN npm install -g pm2
COPY package*.json .
RUN npm install
COPY . .
ADD run.sh /backend/run.sh
ADD style_check.sh /backend/style_check.sh
RUN chmod a+x /backend/run.sh
RUN chmod a+x /backend/test.sh
RUN chmod a+x /backend/style_check.sh
EXPOSE 1337
CMD [“./run.sh”]
```

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

The gitlab-ci.yml file configures the scripts to be executed by the GitLab Runner to build, test, and deploy the applications in a Kubernetes cluster managed by the Rancher platform.

The web application repository contains a docker file that packs up the frontend application to run as the following snippet of code shows:

```
FROM node:10.15.3

RUN printf "deb http://archive.debian.org/debian/ jessie main\ndeb-src http://archive.debian.org/debian/ jessie main\ndeb http://security.debian.org jessie/updates main\ndeb-src http://security.debian.org jessie/updates main" > /etc/apt/sources.list

RUN apt-get update

RUN apt-get install -y npm

RUN apt-get install -y nginx

WORKDIR /frontend

ADD . /frontend/

RUN npm install

RUN npm audit fix

RUN npm i @angular-devkit/build-angular@0.803.24

RUN npm run build:prod

ADD ./bd4nrg.conf /etc/nginx/sites-available/bd4nrg.conf

RUN cp -r /frontend/dist/* /usr/share/nginx/html

RUN ln -s /etc/nginx/sites-available/bd4nrg.conf /etc/nginx/sites-enabled/bd4nrg.conf

# Forward request logs to Docker log collector

RUN ln -sf /dev/stdout /var/log/nginx/access.log \

&& ln -sf /dev/stderr /var/log/nginx/error.log

EXPOSE 80 443

CMD ["nginx", "-g", "daemon off;"]
```

Additionally, an Ingress[17] load balancer exposes Hypertext Transfer Protocol Secure (HTTPS) routes from outside the cluster to services within the cluster, traffic load balancing, Secure Socket Layer (SSL) / Transport Layer Security (TLS) termination, allowing traffic routing, and offer name-based virtual hosting. This configuration allows the applications and services to scale up on demand.

2.3 Functional Specification

This section aims at describing the functionality of the BD4NRG Marketplace that includes the composition of the functionality of the tools and services of the BD4NRG framework integrated to provide a single-entry point to the energy stakeholders. The functionalities described comprises the integration of the functionality of the life cycle of the actors involved in the BD4NRG framework, the services functionality, the catalogue functionality, and the access layer functionality to access the BD4NRG services.

As hinted before, the functionality identified in this first technology release will be extended and refined in the later technology releases according to the progress in the definition and specification of the different tools and services.

[17] https://www.nginx.com/resources/glossary/kubernetes-ingress-controller/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.1 User Management

This component will allow users, actors, and stakeholders of the BD4NRG platform to register and login in the Marketplace application with the required roles and permissions to manage and query their resources and to configure and consume the provided services. The user management functionality will be provided by the Identity Management Service deployed in the context of WP3. The roles identified in the BD4NRG framework are for example: Data Providers and Data Consumers, Service Providers and Service Consumers. The entire list of the BD4NRG user roles will be completed from the specification of the BD4NRG tools and services and the analysis of the pilot's requirements.

2.3.1.1 Requirements

Requirement #	Description
UM_R1	The Marketplace module shall manage the users and roles provided by the Identity Manager (IdM).
UM_R2	The Marketplace shall login a BD4NRG in the Marketplace application.
UM_R3	The Marketplace shall provide the profile information of the user.
UM_R4	The Marketplace shall validate credentials through the IdM system.
UM_R5	The Marketplace shall support user registration and approval process.

2.3.1.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [18]
UM-US1	1	Data Provider	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Provider User and Register an Application.	I have received the confirmation of a verified account from the Identity manager component.	Must
UM-US2	1	Data Consumer	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Consumer	I have received the confirmation of a verified account from the Identity	Must

[18] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

				User Register an Application.	manager component.	
UM-US3	1	Service Provider	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Service Provider User and Register an Application.	I have received the confirmation of a verified account from the Identity manager component.	Must
UM-US4	1	Service Consumer	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Service Consumer User Register an Application.	I have received the confirmation of a verified account from the Identity manager component.	Must
UM-US5	1	User	Check my user account information and permissions.	I can manage my user account and check on the permissions granted	I have received the confirmation of my account information from the Identity manager component.	Must
UM-US6	1	User	Validate my credentials to login the Marketplace.	I can access the Marketplace functionality.	I have received an authentication token from the Identity manager component.	Must

2.3.1.3 Required Interfaces

Interfaces that the component requires from the BD4NRG Identity Manager.

Interface #	Description
UM_IN_1	IdM user creation REST API
UM_IN_2	IdM user management REST API
UM_IN_3	IdM user credential validation REST API

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.2 Utilities Sandboxes/SIMaaS

The BD4NRG Marketplace integrates the functionality of the Utilities Sandboxes/SIMaaS described in sections 4 of this document. The Utilities Sandboxes component exposes its functionality through REST API and the Marketplace component uses that API to integrate the functionality providing a unified GUI through the web application.

2.3.2.1 Requirements

Requirement #	Description
SIM_R1	The Marketplace module shall connect to the SIMaaS component to provide the utilities sandboxes functionality.
SIM_R2	The Marketplace module shall allow users connect to the SIMaaS component to request the functionality
SIM_R3	The Marketplace module shall provide a GUI to configure and monitor the SIMaaS assets

2.3.2.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW[19]
SIM-US1	1	Service Consumer	Configure a simulated data testbed on BD4NRG Marketplace.	I may have access to operational experimentation scenarios.	I have received the confirmation from the SIMaaS component that the testbed is configured and running.	Must
SIM-US2	1	Service Consumer	Have access to simulated data from the configured testbed on BD4NRG Marketplace.	I may access real life smart grid operational data without disrupting the real infrastructure	I receive the simulated data from the SIMaaS configured testbed.	Must

[19] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.2.3 Required Interfaces

Interfaces that the component requires from the BD4NRG SIMaaS component.

Interface #	Description
SIM_IN_1	Testbed configuration REST API
SIM_IN_2	Testbed monitoring REST API

2.3.3 DaaS

The BD4NRG Marketplace integrates data sharing and data consuming functionality of provided by the data services and governance services developed in WP3. Also, the data catalogue and the data discovery service provided in the context of WP4 will be integrated to configure the functionality to share and consume data. The DaaS functionality mainly includes the configuration of the user data assets to be shared with other users that can discover them and request for data consuming permissions. The Marketplace shall provide a unified GUI to provide the data sharing configuration functionality to the data provider users and the data accounting and data permissions to the data consumer user. The actual data sharing and data sharing configuration and functionality are provided by the respective services through REST API and the Marketplace component uses that API to integrate the functionality providing a unified GUI through the web application.

2.3.3.1 Requirements

Requirement #	Description
DS_R1	The Marketplace module shall provide the visualizations of the data catalogue.
DS_R2	The Marketplace module shall provide the visualization of data discovery request and response.
DS_R3	The Marketplace shall provide data providers users the functionality to configure a data sharing
DS_R4	The Marketplace shall provide data consumer users the functionality to request access permission to data assets.
DS_R5	The Marketplace module shall data providers users the functionality to grant permissions of their data assets to data consumer users.
DS_R6	The Marketplace shall provide the users accounting information about data shared and consumed

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.3.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [20]
DS-US1	1	Data Consumer	Search the available data of the data catalogue.	I can create or improve my services.	I have received data assets catalogue with their metadata.	Must
DS-US2	1	Data Consumer	Discover new data assets according to some metadata specification.	I can create or improve my services.	I have received new data assets information according to my search criteria	Must
DS-US3	1	Data Provider	Configure my data assets to be shared with data consumers.	I can share my data according to my needs and data availability.	I have received the confirmation of the data sharing configuration.	Must
DS-US4	1	Data Consumer	Request for permission to access shared data.	I can receive shared data according to its description and data sharing configuration.	I have received the confirmation of the granted permissions over data.	Must
DS-US5	1	Data Provider	Grant access permission to the data consumers.	I can share my data without losing the control over my data assets.	I have received the confirmation of the granted permissions.	Must
DS-US6	1	Data Provider	Get information about the accounting of the shared data.	I can monitor the shared data.	I have received the accounting information over data assets and data consumers.	Must

[20] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

DS-US7	1	Data Consumer	Get information about the accounting of the consumed data.	I can monitor the consumed data.	I have received the accounting information over data assets from data providers.	Must

2.3.3.3 Required Interfaces

Interfaces that the component requires from the BD4NRG tools and services components.

Interface #	Description
DS_IN_1	Discovery Service REST API
DS_IN_2	Data Catalogue REST API
DS_IN_3	Data Sharing Configuration REST API
DS_IN_4	Data Asset REST API
DS_IN_5	Data Governance REST API
DS_IN_6	Tokenization REST API

2.3.4 Resources Catalogue

The BD4NRG Marketplace integrates the user's storage and computing resource in a resource catalogue for shareable assets. The functionality mainly includes the functionality to configuration of the user to be shared with other users that can discover them and request for access permissions. The Marketplace shall provide a unified GUI to provide the assets sharing configuration functionality to the asset provider users and the data accounting and data permissions to the asset consumer user. The actual asset sharing and asset sharing configuration and functionality are provided by the respective services through REST API and the Marketplace component uses that API to integrate the functionality providing a unified GUI through the web application.

2.3.4.1 Requirements

Requirement #	Description
RS_R1	The Marketplace module shall provide the visualizations of the assets catalogue.
RS_R2	The Marketplace module shall provide the visualization of assets discovery request and response.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Requirement #	Description
RS_R3	The Marketplace shall provide assets providers users the functionality to configure an asset sharing.
RS_R4	The Marketplace shall provide assets consumer users the functionality to request usage permission to assets.
RS_R5	The Marketplace module shall assets providers users the functionality to grant permissions of their assets -to- assets consumer users.
RS_R6	The Marketplace shall provide the users accounting information about assets shared and consumed

2.3.4.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [21]
RS-US1	1	Asset Consumer	Search the available assets of the assets catalogue.	I can create or improve my services.	I have received assets catalogue with their metadata.	Must
RS-US2	1	Asset Consumer	Discover new assets according to some metadata specification.	I can create or improve my services.	I have received new assets information according to my search criteria	Must
RS-US3	1	Asset Provider	Configure my assets to be shared with asset consumers.	I can share my assets according to my needs and availability.	I have received the confirmation of the asset sharing configuration.	Must

[21] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

RS-US4	1	Asset Consumer	Request for permission to access shared assets.	I can receive shared asset according to its description and sharing configuration.	I have received the confirmation of the granted permissions over the assets.	Must
RS-US5	1	Asset Provider	Grant access permission to the asset consumers.	I can share my assets without losing the control over them.	I have received the confirmation of the granted permissions.	Must
RS-US6	1	Asset Provider	Get information about the accounting of the shared assets.	I can monitor the shared assets.	I have received the accounting information over assets and data consumers.	Must
RS-US7	1	Asset Consumer	Get information about the accounting of the consumed assets.	I can monitor the consumed assets.	I have received the accounting information over assets from data providers.	Must

2.3.4.3 Required Interfaces

Interfaces that the component requires from the BD4NRG framework components.

Interface #	Description
RS_IN_1	Assets Discovery Service REST API
RS_IN_2	Assets Catalogue REST API
RS_IN_3	Assets Sharing Configuration REST API
RS_IN_4	Asset REST API
RS_IN_5	Data Governance REST API
RS_IN_6	Tokenization REST API

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.5 ML and Analytics Tools/Services Catalogue

The BD4NRG Marketplace shall integrate ML and Analytics tools and services developed in WP4 and WP5. The DaaS functionality mainly includes the configuration of the of the tools and services to be used by the service consumer users. The Marketplace shall provide a unified GUI to provide the service/tool configuration functionality to the service consumer users. The service consumer user shall get accounting data about the service/tool utilization. The ML and Analytics service configuration and functionality are provided by the respective services through REST API and the Marketplace component will use that API to integrate the functionality providing a unified GUI through the web application. In the future technology release specification, a more detailed and extended functionality will be provided according to the services specification level provided by the respective WPs.

2.3.5.1 Requirements

Requirement #	Description
ML_R1	The Marketplace shall provide the visualizations of the ML and Analytics tools and service catalogue.
ML_R2	The Marketplace shall provide service consumer users the functionality to configure ML and Analytics tools and service.
ML_R4	The Marketplace shall provide the users accounting information about ML and Analytics tools and service consumed

2.3.5.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [22]
ML-US1	1	Service Consumer	Search the available ML and Analytics tools and service in the catalogue.	I can create or improve my services.	I have received ML and Analytics tools and service catalogue with their metadata.	Must

[22] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

ML-US2	1	Service Consumer	Configure the ML or Analytics services.	I can create or improve my services.	I have received the confirmation and the information of the configured service.	Must
ML-US3	1	Service Consumer	Download the ML or Analytics tool.	I can deploy the services on premises.	I have received the ML or Analytics Toolbox and the configuration information.	Must
ML-US4	1	Service Consumer	Get information about the accounting of the ML or Analytics services.	I can monitor the ML or Analytics services usage.	I have received the accounting information of the ML or Analytics services consumed.	Must

2.3.5.3 Required Interfaces

Interfaces that the component requires from the BD4NRG ML and Analytics services and tools components.

Interface #	Description
ML_IN_1	ML and Analytics services and tools Catalogue REST API
ML_IN_2	ML and Analytics services and tools Configuration REST API
ML_IN_3	ML and Analytics services and tools REST API
ML_IN_4	Data Governance REST API
ML_IN_5	Tokenization REST API

2.3.6 Marketplace Web Application

The web application of the BD4NRG Marketplace is the GUI that integrates all the functionality for the interaction of the users and stakeholders with the tools and services available in the BD4NRG platform.

The application provides the GUI of the functionality described in the User Management section, the Utilities Sandboxes, the DaaS, the Resources Catalogue, and the ML and Analytics Tools and Services section. To provide the information required the web application uses the Marketplace REST API.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.3.6.1 Required Interfaces

Interfaces that the component requires from the Marketplace core component:

Interface #	Description
WA_IN_1	IdM user creation REST API
WA_IN_2	IdM user management REST API
WA_IN_3	IdM user credential validation REST API
WA_IN_4	MP Testbed configuration REST API
WA_IN_5	MP Testbed monitoring REST API
WA_IN_6	MP Discovery Service REST API
WA_IN_7	MP Data Catalogue REST API
WA_IN_8	MP Data Sharing Configuration REST API
WA_IN_9	MP Data Asset REST API
WA_IN_10	MP Governance REST API
WA_IN_11	MP Accounting REST API
WA_IN_12	MP Assets Discovery Service REST API
WA_IN_13	MP Assets Catalogue REST API
WA_IN_14	MP Assets Sharing Configuration REST API
WA_IN_15	MP Asset REST API
WA_IN_16	MP ML and Analytics services and tools Catalogue REST API
WA_IN_17	MP ML and Analytics services and tools Configuration REST API
WA_IN_18	MP ML and Analytics services and tools REST API

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3 DLT and Smart Contracts for Multi-sided Configurable Flexible Market Platforms

3.1 Architecture: Blockchain-Based Asset Management Layer

The BD4NRG Marketplace integrates the Blockchain-Based Asset Management Layer, which will be implemented by the ENG partner. This layer involves the DLT/blockchain and smart contracts technologies which are able to coordinate the decentralized data exchange as well as the peer-to-peer (P2P) decentralized financial mechanisms. The use of the blockchain architecture does not start from scratch but leverages on the ongoing work carried out by ENG in the H2020 PlatONE project [4] .

The blockchain infrastructure makes possible the use of the smart contracts, which can be defined as tools able to automatically execute transactions under certain conditions without requiring intermediary entities. They are often associated with Ethereum [5], a blockchain that was designed to accommodate smart contracts without restriction to any particular platform.

Blockchain technology as well as smart contracts play a key role in the BD4NRG Marketplace since they bring new schemas of coordination among customers, such as Peer-To-Peer (P2P) trading and guarantee a transparent unmodifiable data management and sharing. The smart contracts will be used for managing the services provided by the market platform, such as the MLaaS, the SIaaS, the 3rd party service, and the DaaS. The users will have the opportunity to buy and sell services for each asset offered by the Marketplace and, in the meantime, the validity checks provided by the smart contracts will ensure that the market session rules are not violated.

Within the BD4NRG Marketplace, the Blockchain Based Asset Management Layer (Figure 5) will be in charge of tokenization functionality to create a utility monetary and/or non-monetary reward/incentive metric/function. The tokens will be defined in a specific smart contract and assigned to users in exchange of the services provided. Moreover, this technology allows to make the policy for the token assignment completely customizable. The remuneration process could be implemented with the usage of Ethereum Request for Comments (ERC) 20, also known as (ERC-20), tokens as a way to reward users involved in the market (see paragraph 3.1.1).

The Blockchain-Based Asset Management Layer can comprise a web blockchain dashboard that allows the participants to monitor the transactions, including a visual tool for data integrity verification; the users would have the opportunity to check their own market operations through a Blockchain Explorer.

The Layer encompasses two different modules: Data handling and Digital assets tokenization. Data handling module provides the management of the data, including the sharing of the data coming from the Energy Assets Data Access Layer. In this case, the blockchain technology is used for data certification and integrity and for data interoperability, using standard communication protocols and following the data security and data privacy best practices. Indeed, the module will implement the Data Access Policies (DAP) for sharing the data to all the relevant stakeholders in a secure and transparent way. Digital assets tokenization module is related to the digital assets trading, able to exploit the tokenization model. In the blockchain ecosystem, the platform is able to “tokenize” the

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

settlement outcomes enabling a token-based remuneration process that the users can exploit for payments. Both modules rely on the blockchain smart contracts for trustiness.

In summary, the described characteristics enable a blockchain-driven Marketplace to ensure the market transactions certification, tracking the registration and validation of market data. The participants, data consumers and data providers, will have the opportunity to sell and buy the services offered by the platform through a fully transparent settlement based on tokenization.

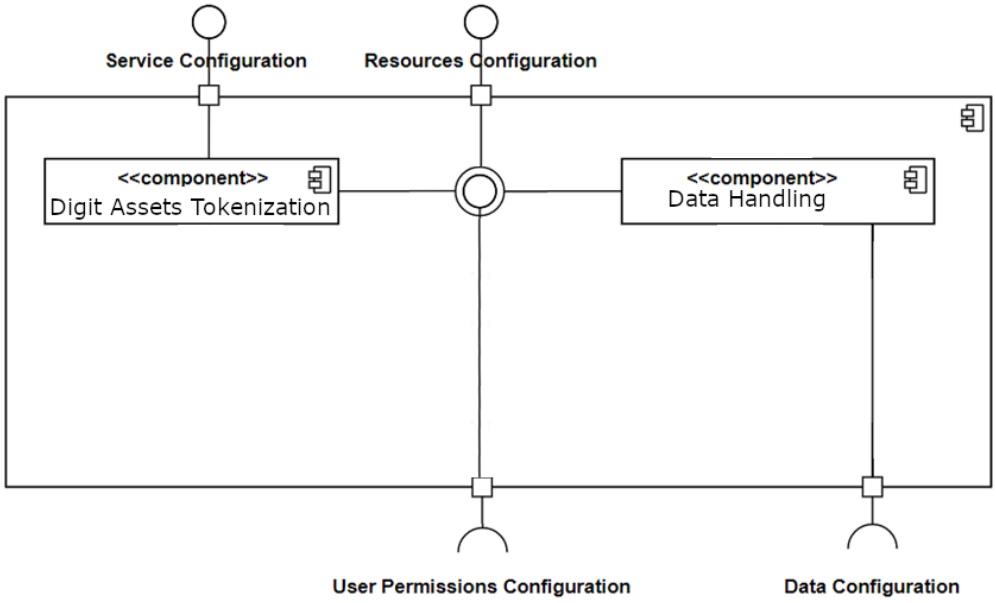

Figure 5: Blockchain Based Asset Management Layer Component Diagram

3.1.1 Market Transactions

Blockchain-Based Asset Management Layer will include Ethereum blockchain nodes and smart contract services, which can directly manage market transactions among two or more actors on a blockchain, being completely self-enforcing.

Ethereum, like all cryptocurrencies, works based on a blockchain network and it is considered the most popular after Bitcoin. However, while Bitcoin was created as an alternative to national currencies, Ethereum was intended as a platform to facilitate immutable contracts and applications via its own currency. In the Ethereum system, the transactions are rewarded with cryptocurrency tokens called Ether (ETH) or with customised tokens.

Currently there are two main categories of customized tokens on Ethereum: ERC-20 [6] for fungible tokens that are not unique and interchangeable, and then, can have a value like a currency note; ERC-721 [7] for non-fungible tokens, which are unique and can be used for collectible goods.

3.1.2 Data Management

The system covers data security and privacy aspects following the best practices for protecting data from unauthorized access and data corruption during its lifecycle. Beyond data tokenization and

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

encryption, data security involves some management practices able to protect data across all platforms.

All the data are registered at the level of an individual user and then stored as immutable transactions. This allows the provision of two important features: data immutability and data provenance. Blockchain technology can guarantee the immutability of data records once those records have come into the system. The transactions are stored in data structures able to ensure the provenance property by authorizing their tracking back until the moment of their registration in the blockchain.

The distributed ledger is a collection of blocks, linked back using hash pointers. Each block stores a set of valid transactions on the registered digital assets. If a change appears in the previous registered nodes, it will produce an inconsistency since the hash pointer of that block has been modified. In case of necessity of changing the content of the previous block and obtaining a consistent updated data structure, all the following blocks will need to be re-hashed and re-linked.

Since the blocks are always added at the head of the chain, the data structure is called append-only type. This offers a noteworthy advantage of preserving historical information, including the order in which the transactions are registered. Moreover, the probability of changing the value of the implemented asset in a block by an attacker decreases with the number of blocks following that block in the append-only linked list. To ensure the ownership of the data, the user provides the signature over the transfer transaction showing the ownership of the asset, in this way the transfer is authenticated and validated.

3.2 Functional Specification

3.2.1 Requirements

While the functional requirements (FR) (Table 3) are defined from the use cases and, more in general, from the project goals, the non-functional requirements (NFR) can be deducted from the literature in relation to the system functionalities. Indeed, they can be defined as the quality attributes of an architecture that defines how the system should behave.

Some typical NFR are:

- reliability: system availability to the end user at any given time;
- efficiency: response time, transit delay, latency;
- performance;
- scalability: the system is available to more users and covers wider areas;
- expandability: the system can be expanded with new types of service;
- interoperability: the system is able to interact with other external systems;
- security;
- privacy;
- maintainability;
- resilience.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Table 3: Functional requirements

Requirement #	Description
FR_1	The Marketplace shall provide tokenization system for the settlement through Smart Contracts functionalities.
FR_2	The Marketplace shall register all the data transactions.
FR_3	The Marketplace shall allow participants to verify all the market data registered in the blockchain.
FR_4	The Marketplace shall expose REST APIs for collecting services requests.

3.2.2 User Stories

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW[23]
	1	Service Consumer	Log in the Marketplace	I may access the BD4NRG platform	I have not received any error message	Must
	2	Service Consumer	Access to the Marketplace functionalities	I can consult the services provided by the platform	I have received an authentication	Must
	3	Service Consumer	Read information of the services provided by the platform	I can select the service I need	I have received an authentication	Must
	4	Service Consumer	Buy a service	I can use that service	I have received the confirmation of the successful market transaction	Must
	5	Service Provider	Log in the Marketplace	I may access the BD4NRG platform	I have not received any error message	Must
	6	Service Provider	Receive the notification of the market transaction	I can check my transactions	The tokens are transferred from the Service Consumer wallet to my wallet	Must

[23] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3.2.3 Interfaces

All the REST APIs exposed by the Blockchain based asset management layer implement an authentication mechanism based on Oauth2.0 [8] over an HTTPS connection. Table 4 reports the list of APIs exposed.

Table 4: REST APIs list

Name	Url	Method	Parameters	Responses
Get Data	/scd/getData	GET	In body: query: JSON Object with the query	Success (200) requestedData Error (500) Error Message - *String*
Validate Data	/bap/validateData	GET	In body: data: JSON Object with data to be validated	Success (200) Success Message: String (OK for valid data, KO for invalid data) Error (500) Error Message - *String*
Get Data	/scd/getData	GET	In body: query: JSON Object with the query	Success (200) requestedData Error (500) Error Message - *String*
Validate Data	/bap/validateData	GET	In body: data: JSON Object with data to be validated	Success (200) blockchainTransactions Error (500) Error Message - *String*

Table 5 shows features and descriptions of the standard APIs for managing tokens.

Table 5: Standard API for tokens management

Name	Description	URL	Method	Parameters	Responses
transfer	Transfer tokens from the owner's wallet to another one	/v1/api/transfer	POST	In body: address: wallet address - String	Success (200) – Confirmation Message - *String*

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Name	Description	URL	Method	Parameters	Responses
				value: transfer value - Number	Error (500) Error Message – *String*
getBalance	Token Balance of a specific Wallet (you must be the owner of the wallet or a superadmin)	/v1/api/balance/:address	GET	In params: address: wallet address - String	Success (200) – Token Balance – Number Error (500) Error Message –*String*

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4 Utilities Sandboxes

In this section, we describe the Utilities Sandbox/SIMaaS module based on VILLAS provided by RWTH. Firstly, the architecture with the inner subcomponents is presented. Subsequently, all relevant implementation and deployment technologies are listed. Furthermore, the functional specification section depicts the feature set concisely. Its subsections section characterizes the requirements, user stories and interfaces the component exposes so that the module integrates with the Marketplace.

As the module will be based on VILLAS, please refer to the following sources for additional information:

- FEIN Website [9]
- Lab exercises [10]
- Git doc source [11]
- VILLAS concept - Slide share slides [12]

4.1 Architecture

In this section the architecture of RWTH's SIMaaS module is described following the premise sketched in Figure 5 below.

It depicts an abstract functional architecture comprising Data Gateway, Control Components and Web Interface Components. Subsequently, two concrete example deployments are discussed.

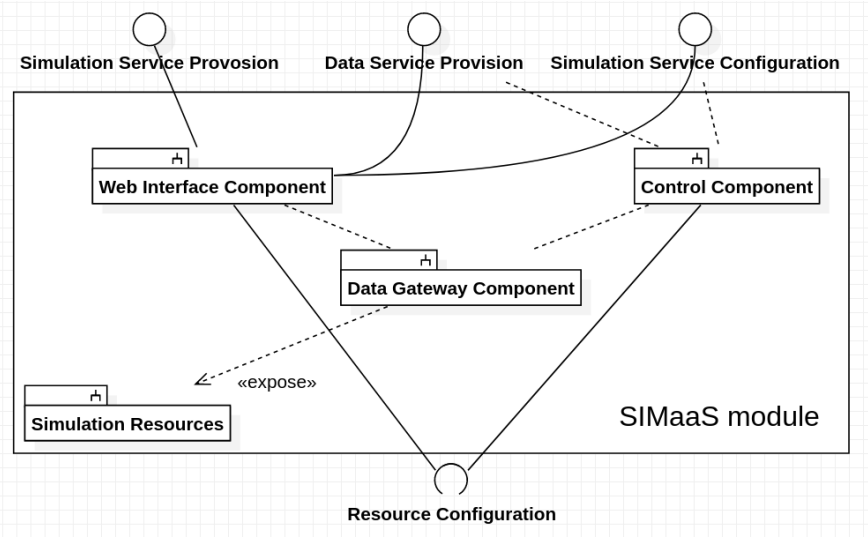

Figure 6:Abstract architecture of the provided module

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

The Data Gateway Component is a modular gateway for simulation data. It offers a wide selection of interface types such as among many others broker-based messaging protocols such as AMQP and MQTT, web protocols like WebSockets and FIWARE NGSI, as well as IEC 61850. Other features include the collection of QoS statistics such as packet loss, latency jitter. Furthermore, network emulation to model geographically dispersed co-simulation is offered.

The Control Component is a set of scripts to control the various parts that constitute the co-simulation: simulators, gateway nodes and data bases. It offers a unified API to control simulation hardware from different vendors. It makes use of AMQP to implement remote control akin to JSON-RPC.

The Web Interface Component is a GUI that allows the administration of the simulators, simulations and models. It is tightly integrated with the Control Component in that it is used to control co-simulations and query the simulator status.

Figure 7 and Figure 8 depict slightly **simplified example** configurations. The first example shows how a DPSim instance is run on a Kubernetes cluster. The next paragraphs are a walkthrough of these examples. Classes that will be mentioned are explained in further sections. Each block represents either a Docker container, Kubernetes pod or hardware node. The numbers in the top right-hand corner indicate the number of instances of these blocks in the example deployment. The labels on the connections are the communication protocols between the blocks.

A user opens the browser on his local machine and accesses the VILLASweb frontend (which could be integrated into or linked to in the Marketplace). The web frontend then asks the backend which components, scenarios and users are available. The web backend controls the infrastructure components through VILLAScontrol. This control is implemented indirectly through an AMQP broker to which the VILLASweb backend subscribes. Purely virtual simulations (= not involving remote lab hardware) are run in simulation pods that contain three containers: the simulator itself (in this case DPsim), VILLAScontrol to be able to control and read the status of the simulation and VILLASnode to be able to livestream simulation data to the web frontend through VILLASrelay. Additionally, VILLAScontrol also takes care of fetching the simulation models and storing the results in the S3 Object Store.

Another special instance of VILLAScontrol labelled as Kubernetes manager that take care of creating and disposing of simulation pods at the user's behest.

The second example with the OPAL-RT simulator is quite similar, the only difference being that in the second case dedicated simulation hardware and two accompanying containers are run outside of the cluster in a remote lab.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

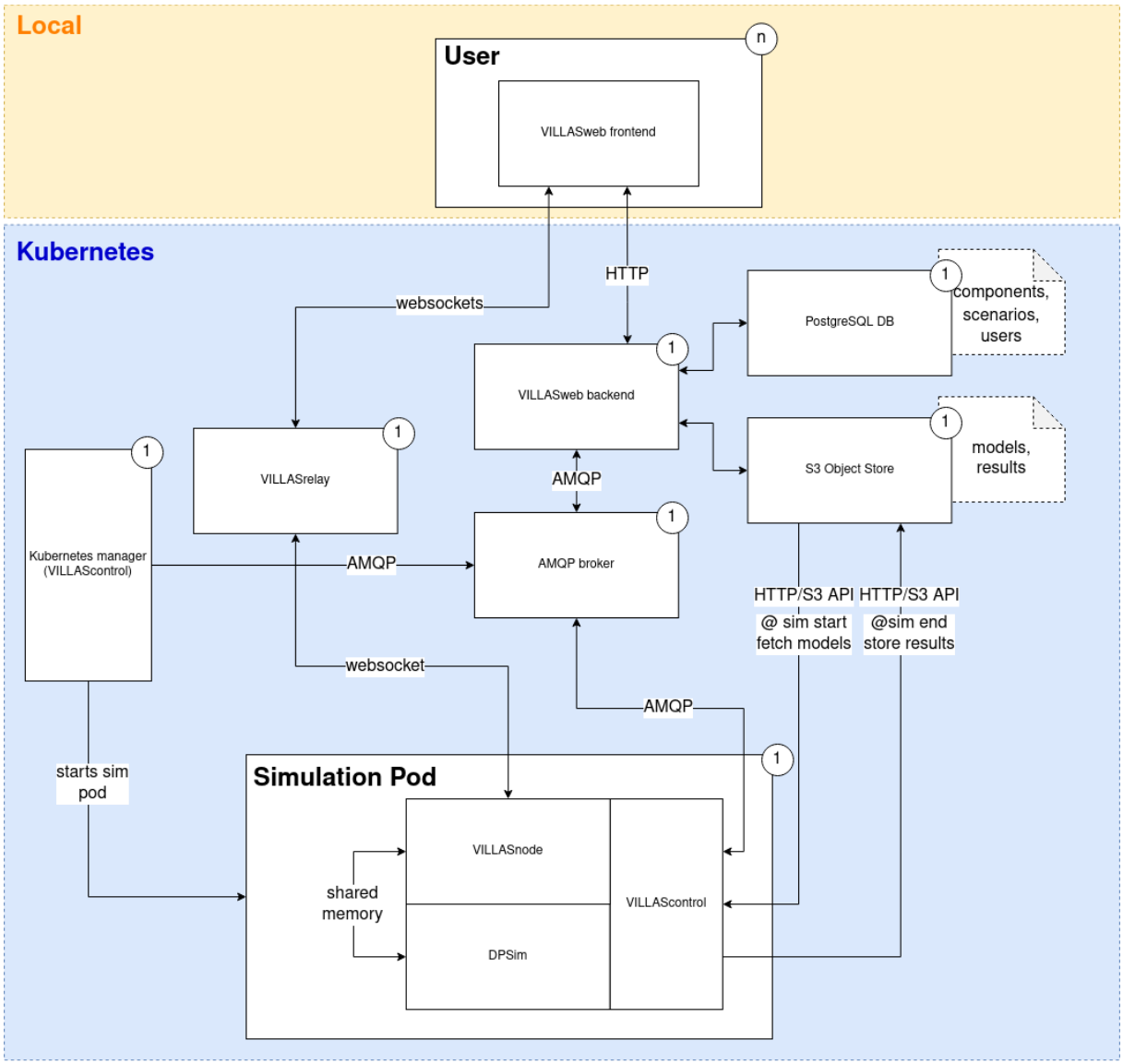

Figure 7: Example Deployment 1: a single simulation running in the Kubernetes cluster.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

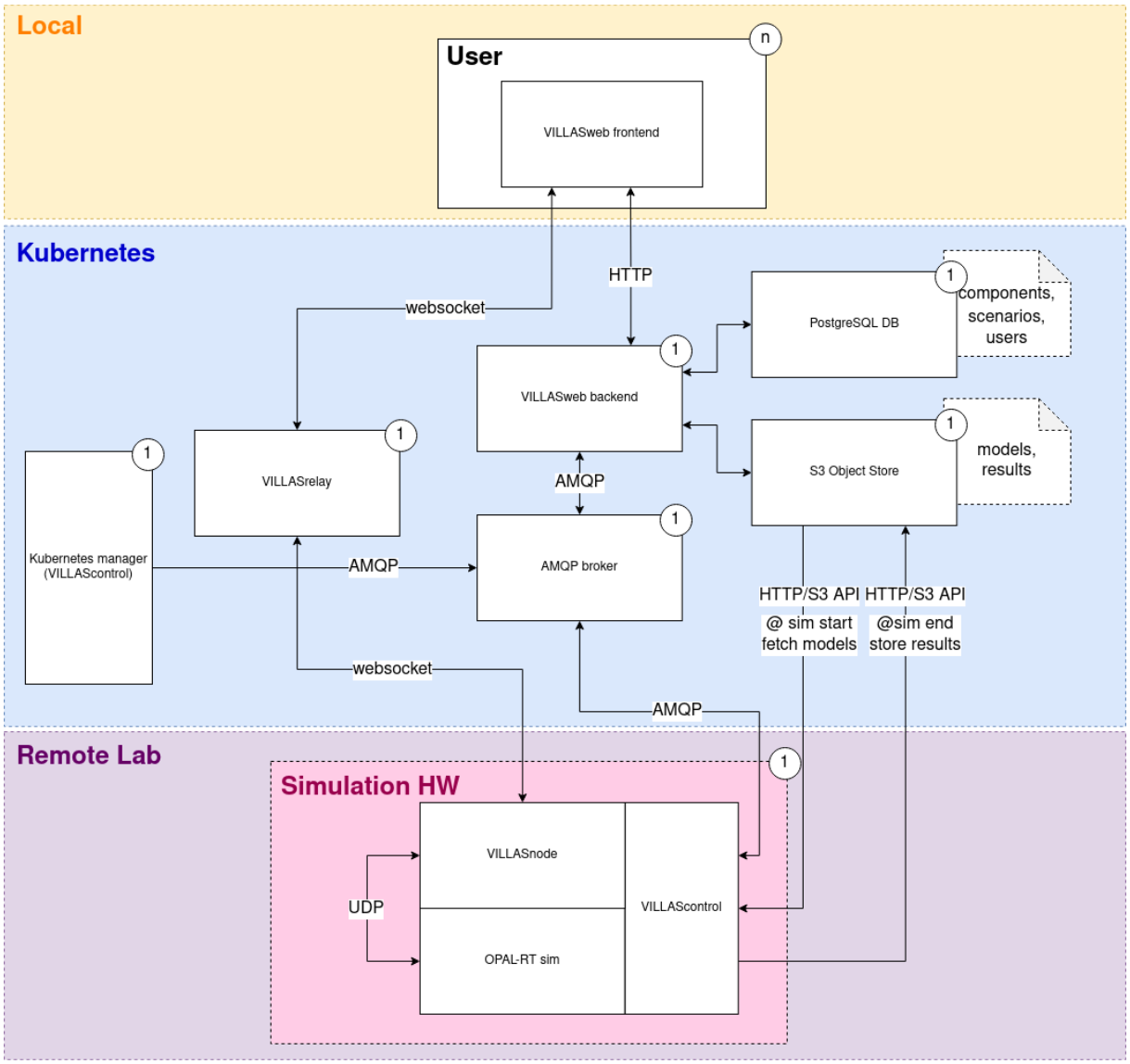

Figure 8: Example Deployment 2: a single simulation is running remotely on lab hardware.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4.2 Implementation and Deployment Technologies

The components are independent of each other and can thus be run/deployed independent of each other. The recommended deployment is through a Kubernetes Helm chart for components that are not directly tied to physical simulation hardware.

The Data Gateway Component (VILLASnode) is implemented in C++. It can be deployed via Docker, installed as an RPM package for Redhat-based Linux distributions, as bootable live Linux, compiled from source, and be used with Kubernetes (Helm chart). For further information please consult the installation guide.

The Web Interface Component (VILLASweb) comprises a frontend and a backend subcomponent. The web frontend is written in Javascript/React.js and connects to the Data Gateway Component (VILLASnode) and to VILLASweb-backend-go which is implemented in go.

The Control Component is a set of object-oriented Python scripts.

4.3 Functional Specification

Most parts of this specification are already implemented. This section is by no means exhaustive, it should only provide an overview of the framework and show how it should be extended. Please refer to the following links for more information:

- VILLAS Data Gateway Component [13]
- VILLAS Web Interface Component [14]
- VILLAS Control Component [15]

4.3.1 Data Gateway Component

This component consists of a server daemon and number of client modules. The server acts as a gateway to forward simulation data from one client to another while collecting Quality of Service (QoS) statistics and handling encryption or tunnelling through Virtual Private Networks (VPNs).

Clients/Node Types

There are two types of clients: socket type clients that support communication over network links (UDP, raw IP, raw Ethernet frames) and specialized clients connecting to simulation equipment that is directly connected to the server or even running on it. Examples for this are the FPGA node type, which directly fetches and pushes data to a PCIe card, and the file node type, which logs or replays simulation data from hard disk.

Here, in Table 6, is the list of currently supported nodes:

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Table 6: Currently supported clients of the Data Gateway Component.

Type	Network Emulation	Read	Write	Vectorize	Status
amqp	no	yes	yes	unlimited	stable
can	no	yes	yes	?	beta
comedi	no	yes	yes	unlimited	beta
ethercat	no	yes	yes	?	alpha
exec	no	yes	yes	unlimited	stable
file	no	yes	yes	unlimited	stable
fpga	no	yes	yes	?	beta
iec61850-8-1	no	yes	yes	1	alpha
iec61850-9-2	no	yes	yes	1	beta
infiniband	no	yes	yes	unlimited	beta
influxdb	no	no	yes	unlimited	stable
kafka	yes	yes	yes	unlimited	stable
loopback	no	yes	yes	unlimited	stable
mqtt	no	yes	yes	unlimited	stable
nanomsg	yes	yes	yes	unlimited	stable
ngsi	no	yes	yes	unlimited	stable
opal	no	yes	yes	1	untested
redis	no	yes	yes	unlimited	stable
rtp	yes	yes	yes	?	beta
shmem	no	yes	yes	unlimited	stable
signal	no	yes	no	1	stable

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Type	Network Emulation	Read	Write	Vectorize	Status
socket	yes	yes	yes	unlimited	stable
stats	no	yes	no	1	stable
temper	no	yes	no	unlimited	stable
test-rtt	no	yes	yes	unlimited	stable
uldaq	no	yes	no	unlimited	stable
websocket	no	yes	yes	unlimited	stable
zeromq	yes	yes	yes	unlimited	stable

Additionally, a number of third-party clients are supported: OPAL_RT Asynchronous Process, RTDS GTNET-SKT, OPAL-RT HYPERSIM, NI LabView, MATLAB, ML507 GTFPGA UDP Interface, Python

Note: new clients/equipment should be implemented as a new node-type as part of VILLASnode. Using a dedicated client-application which communicates via the 'socket' type is deprecated because it leads to code duplication.

Command Line Interface Tools

There are a number of command line tools that serve various purposes. In order to understand their functionality, we first discuss some terminology:

- A **super node** is an instance of the VILLASnode daemon which runs on a physical machine.
- A **node** is an interface to a local or remote simulator, file, database etc. A **super node** consists of one or more **nodes**. A **node** acts as a sink **and** as a source of **samples**.
- A **sample** is an array of floating point or integer values with an associated timestamp and sequence number.
- A **path** connects one or more **nodes** within a **super node** and forwards **samples** from one or more **source nodes** to one or more **destination nodes**.
- A **hook** manipulates or filters **samples** which are processed by a **path**.

The following command line tools are available (see Table 7):

Table 7: Command Line Tools of the Data Gateway Component.

Tool	Purpose
villas node	The main VILLASnode daemon.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Tool	Purpose
villas signal	A signal generator for testing and training purposes.
villas pipe	Send / receive **samples** to / from **nodes** via standard IO streams.
villas hook	Filter or manipulate **samples** provided via standard IO streams.
villas compare	Compare multiple files/streams containing sample data.
villas conf2json	Convert libconfig-style formatted configuration files to JSON format.
villas convert	Convert sample data files/streams between different supported [formats](@ref node-format-types).
villas graph	Generate a graphical representation of the VILLASnode configuration file with Graphviz.

Further information about the options of these tools as well as hook types, payload formats etc. can be found in the documentation.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4.3.2 Web Interface Component [14]

In the following we briefly describe the data model (cf. Figure 9). The Web Interface Component data model has the following classes: Users, Infrastructure Components and Scenarios, Component Configurations, Signals, Files, Dashboards, Widgets and Results.

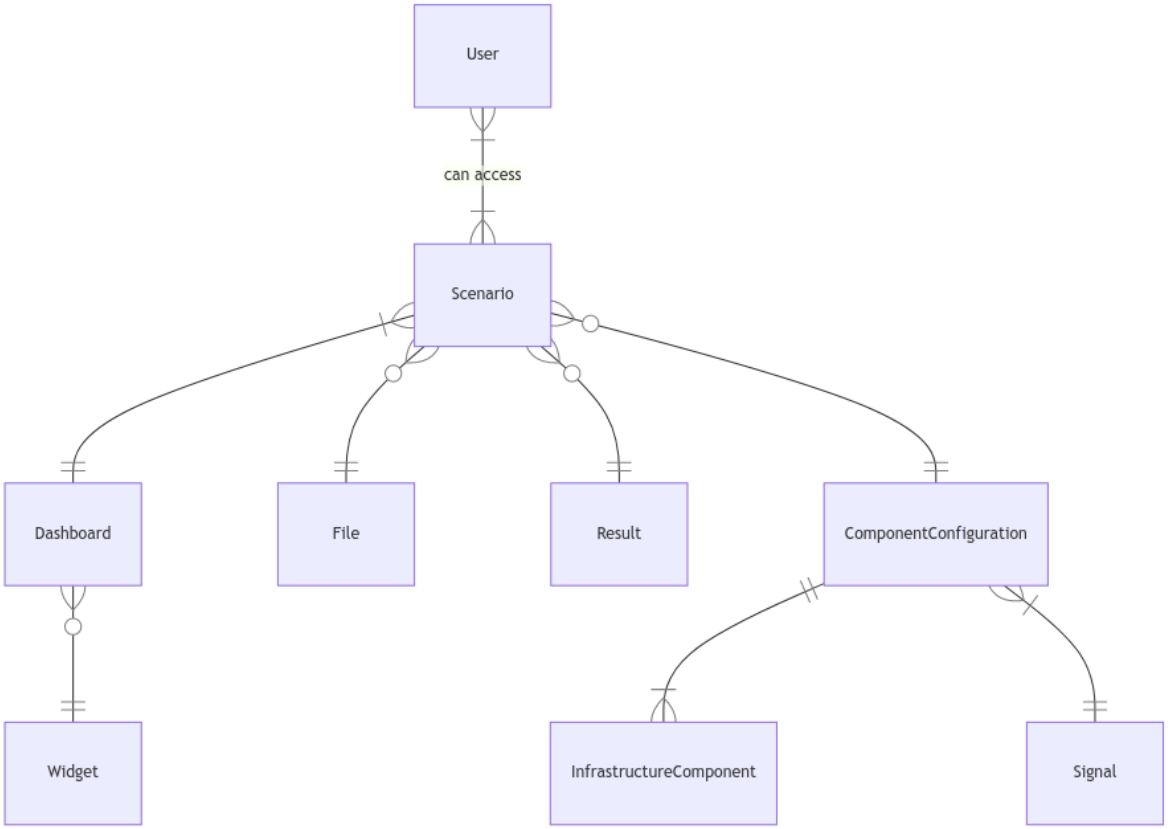

Figure 9: Web Interface Component data model.

Users

There are three kinds of users (all authenticate in VILLASweb with username and password):

- Guests have only read access and cannot modify anything
- Users have access to scenarios, can see available infrastructure components and modify their accounts except for their role.
- Admin users have full access, including changing the role of existing users and creating new users. Only admins can add or modify infrastructure components.

Infrastructure Components

Infrastructure Components map to research infrastructure: they are simulators, gateways, database etc.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Scenarios

- A collection of component configurations, dashboards, and files for a specific experiment (compare Figure 10).
- Users can have access to multiple scenarios.
- Users can be added to and removed from scenarios.

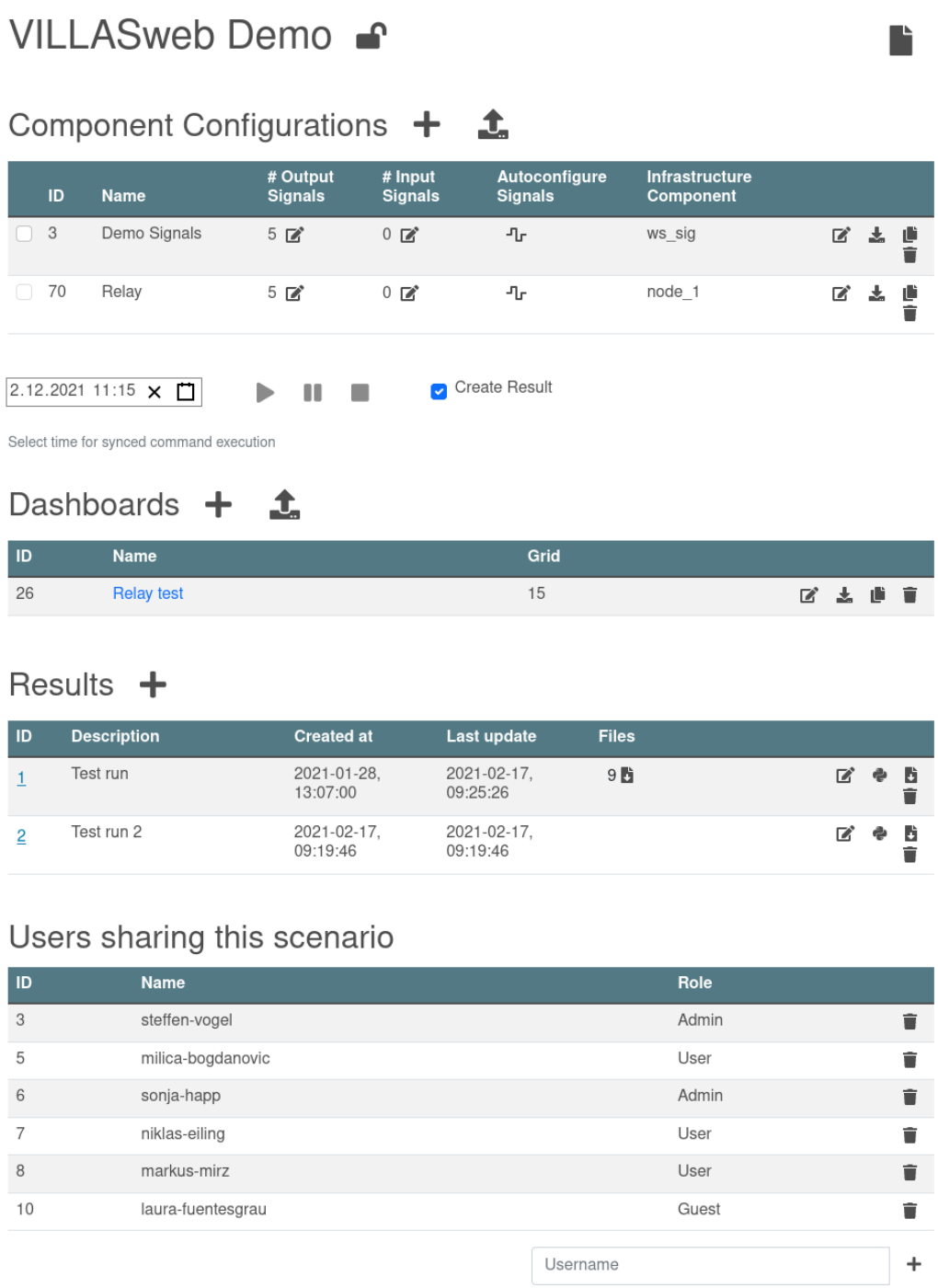

Figure 10: Example Scenario. From the scenario page Component Configurations, Dashboards, Results and Users can be viewed and edited.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Component Configurations and Signals

- Configure an infrastructure component for the use in a specific scenario (see Figure 10).
- Input signals: Signals that can be modified in VILLASweb
- Output signals: Signals that can be visualized on dashboards of VILLASweb
- Parameters: Additional configuration parameters of the infrastructure component
- Signals are the actual live data that is displayed or modified through VILLASweb dashboards

Figure 11: In the Result section of the Dashboard the Component Configuration used for the conducted experiment can be viewed.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

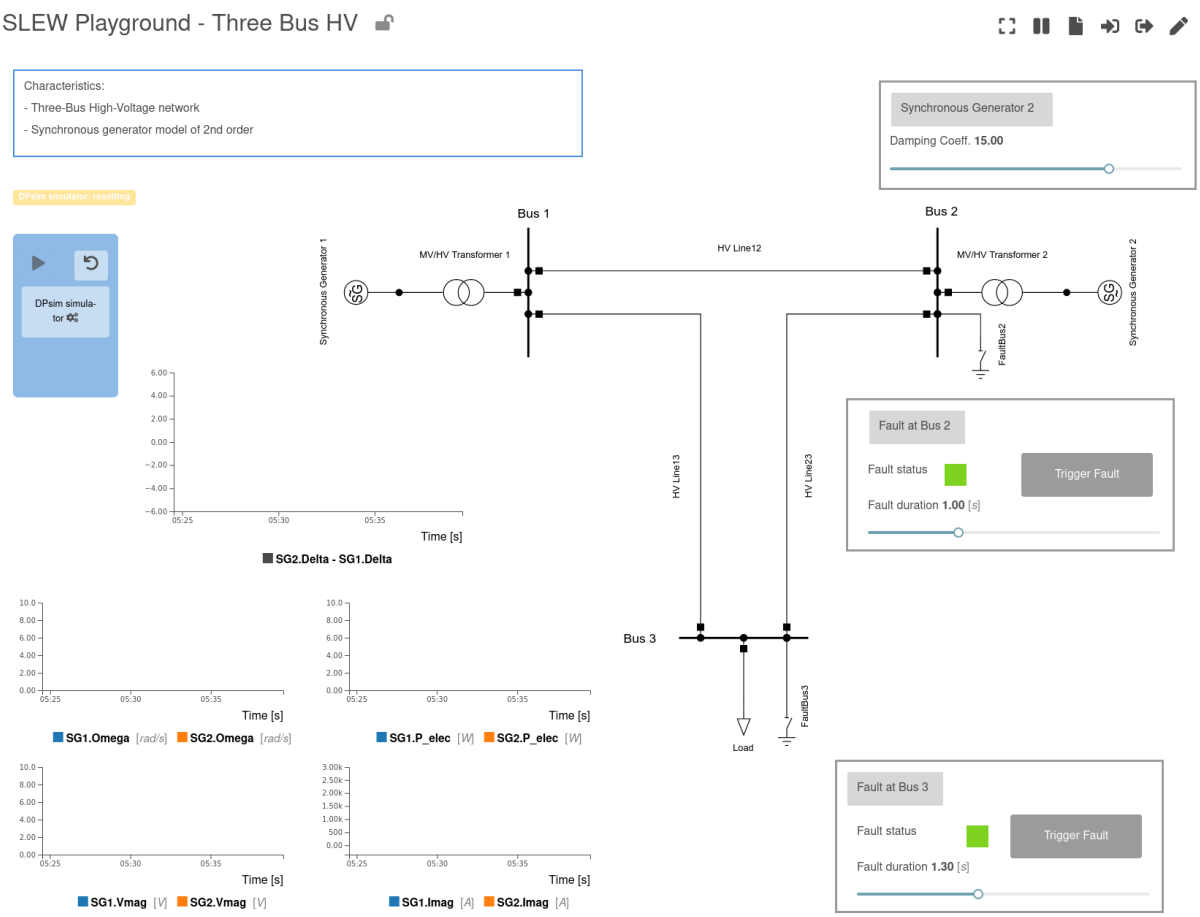

Figure 12: Example Dashboard. Each visualization and control element is a Widget.

Dashboards and Widgets

- Visualize ongoing experiments in real-time (see Figure 12)
- Interact with ongoing experiments in real-time
- Use widgets to design the dashboard according to the needs

Files

- Files can be added to scenarios optionally
- Can be images, model files, CIM, XML files
- Can be used in widgets or component configurations

Results

- Results are the simulation data (csv) and optionally a log file.
- After each execution of an experiment the corresponding results can be downloaded

4.3.3 Control Component

The Control Component (VILLAScontroller) controls and monitors Infrastructure Components (ICs), referred to as Component in Figure 13. These components can be either Managers, Gateways or

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

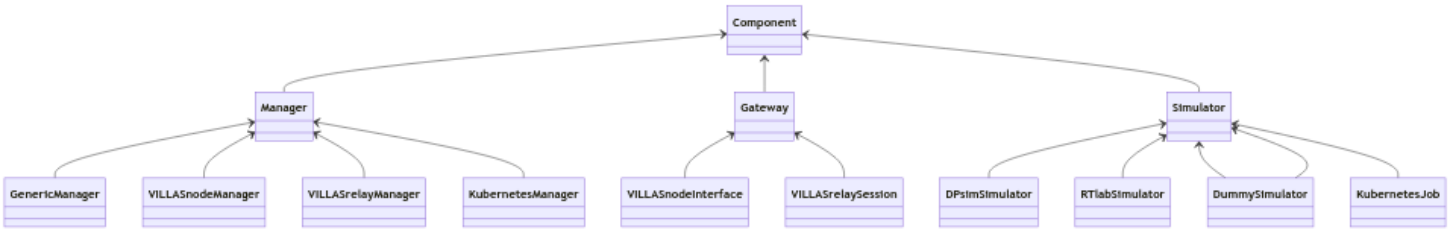

Figure 13: Control Component Data Model.

Simulators. Managers manage e.g., the creation of simulation pods in the case of KubernetesManager. The Gateway classes provide the interface to communicate with the Data Gateway Component. The subclasses of Simulator allow controlling a simulator, e.g., starting, stopping, pausing, resuming and shutting down the simulation. Furthermore, they allow to monitor the state of the simulation. The allowed states and transitions are expressed as a finite state machine.

4.3.4 Requirements

This is the list of requirements the Marketplace **must** implement to support the Utilities Sandboxes module as provided by RWTH. Optional requirements are mentioned in the user stories.

Requirement #	Description
ACC_01	The Marketplace module shall manage the users and roles. Three categories of users exist: Guest, User and Admin. These are their capabilities: • Guests have only read access and cannot modify anything. • Users have access to scenarios, can see available infrastructure components and modify their accounts except for their role. • Admin users have full access, including changing the role of existing users and creating new users. Only admins can add or modify infrastructure components.
ACC_02	The Marketplace should support user groups (see also user stories).
SCE	The Marketplace shall support granting and restricting access to certain scenarios to certain users (currently not implemented in VILLASweb frontend). This option should be at least available upon creation of the scenario but preferably also thereafter. It would be nice if this was also available for dashboards.
AUTH	Provide OAuth2 for authentication.
CLUSTER	A Kubernetes Cluster must be made available to run the depicted pods.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4.3.5 User Stories

These user stories describe how a user typically uses the Utilities Sandboxes module as provided by RWTH through the BD4NRG Marketplace. These user stories reflect only the usage of the module through the BD4NRG Marketplace frontend not all capabilities of the framework. The implementation status in parentheses in the MoSCoW column indicates whether a corresponding feature has been implemented in the VILLASweb frontend. As can be clearly observed most of the features already exists. Therefore, it is recommended to reuse the VILLASweb frontend in the BD4NRG Marketplace. It is available under the GNU General Public License (GPL) v3 license.

Table 8: User Stories of the Web Component.

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [24]
ACC_01	1	Guest (Web Component User)	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Consumer User.	I have received the confirmation of a verified account as part of the BD4NRG system and the API key.	Must (implemented)
ACC_02	1	User (Web Component User)	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Consumer User and register applications.	I have received the confirmation of a verified account as part of the BD4NRG system and the API key.	Must (implemented)
ACC_03	1	Administrator (Web Component User)	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Consumer User and Service Provider User and register applications.	I have received the confirmation of a verified account as part of the BD4NRG system and the API key.	Must (implemented)

[24] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

ACC_04	1	User, Admin (Web Component User)	See and edit my account information	I can change my password or other account information	My account information has changed.	Must (implemented)
ACC_05	1	Web Component User	Make or delete a user group and add one or more users to them.	I can conveniently use user groups for access rights.	The user group exists/is deleted and the user is (not) part of the group anymore.	Should (not implemented)
SCE_01	2	User (Web Component User)	See the available scenarios	I can run the simulation	I have received the simulation result.	Must (implemented)
SCE_02	2	User, Admin (Web Component User)	Change access rights to the scenario.	Only the desired users have access.	Only the desired users have access.	Should (not implemented)
SCE_03	2	User, Admin (Web Component User)	Add/delete new scenario.	Users (no longer) have access to them.	Scenario (does not) show(s) up in the web frontend.	Must (implemented)
RES_01	3	Web Component User	Download the experiment results	I can analyse them.	CSV (and optionally log) are downloaded	Must (implemented)
SIG_01	3	User, Admin (Web Component User)	Create output signal from input signal.	Data based on raw input can be visualized in a widget on a dashboard.	Desired output signal displayed in the dashboard.	Must (implemented)
DASH_01	4	Web Component User	View dashboards	Experiment can be understood/parameters be changed.	Dashboard is accessible.	Must (implemented)
DASH_02	4	User, Admin (Web Component User)	Add widget to dashboard	Experiment visualization can be extended.	Widget is added to the Dashboard.	Should (not implemented)

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

DASH_03	4	User, Admin (Web Component User)	Disable control widgets on dashboard for certain users/user groups (whitelist and/or blacklist).	Experiment control can be controlled granularly.	Control widget input value cannot be changed by user.	Could (not implemented)
INFRA_01	5	Admin (Web Component User)	Add simulation infrastructure.	Simulations can be run on them.	Simulation has been successfully run on them and results are available.	Must (implemented)
INFRA_02	5	Admin (Web Component User)	Remove simulation infrastructure added by me	Unavailable resources cannot be used.	The resource is disposed and/or not accessible anymore.	Must (implemented)

4.3.6 Interfaces

Table 9: Interfaces that the SIMaaS module exposes.

Interface #	Description
Websocket	Websocket (for livestreaming simulation data)
AMQP	AMQP Message Broker Protocol
REST	Web Component REST API (and REST APIs of the other components)
Node CLI	Data Gateway Component Command Line Interface as described in the functional specification

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

5 MLaaS

5.1 Architecture

This module is aimed at providing pre-trained ML models that can be deployed on demand. The main goal is to enable quick adaptation and reuse of machine learning models along different contexts. Its main interactions are depicted in Figure 14.

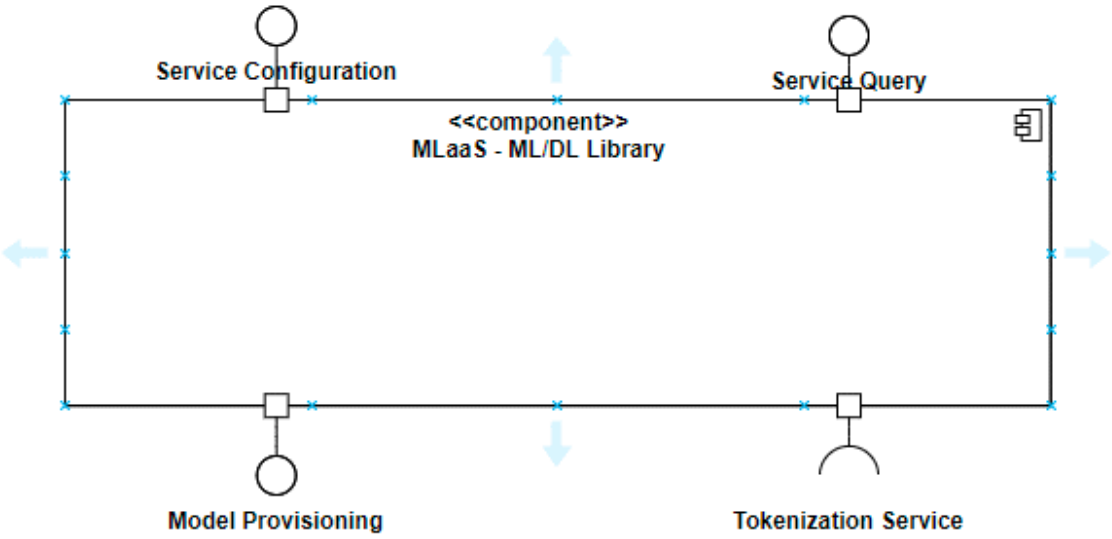

Figure 14: MLaaS – ML/DL Library

5.1.1 Anomaly Detection Service

Some scenarios don't lend themselves for direct reutilization of pre-trained ML models. In certain scenarios ML models might be to specialized on certain idiosyncratic elements of the underlying training data. In essence models might be overfitted during training and yield poor out of sample performance. In these scenarios retraining or finetuning of the algorithm must be considered. This issue is further compounded by the fact that not all ML methods can be retrained or support incremental learning methods. Predominantly out-of-core ML methods must be used for this scenario.

Another scenario deals with unsupervised ML methods, which do not require labeled data. These methods can be used for anomaly detection on both batch and streamed data. In both scenarios the creation of unsupervised ML methods from direct use-case data is preferred. To this end, we propose the introduction of several unsupervised anomaly detection methods which can be directly trained from MLaaS component.

The architecture, see Figure 14, will be based around the concept of a simple processing pipeline. Users will be capable of defining a data source, simple data formatting and pre-processing, method selection and configuration. It is important to mention that one of the most important pre-processing methods relates to data normalization which has a big impact in ML method performance and utility.

When dealing with unsupervised methods in general and anomaly detection methods in particular it is difficult to determine why a particular event or series of events have been marked as anomalous.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

This coupled with the propensity of these types of methods to yield both false positives and negatives make it difficult to make meaningful, actionable decisions. We will also include some basic explainable AI methods which will analyze why a particular event has been designated as anomalous. This form of root cause analysis usually takes the form of eighter a global or event-based feature importance showing how each feature for the dataset contributed to a prediction. Any analysis results will be appendable to the anomaly reporting mechanism.

5.2 Implementation and Deployment Technologies

This section describes the technologies to implement and deploy this specific component.

5.2.1 Anomaly Detection Service

In the case of the unsupervised anomaly detection component, it will be based around several key Python ML libraries. Scikit-learn[25] is one of the most popular Python ML libraries and has several state-of-the-art unsupervised ML methods, including specialized anomaly detection method such as IsolationForest. Although these methods are suitable for most scenarios we aim to provide as robust a solution as possible for the BD4NRG Marketplace. Integration of the PyOD[26] library will increase the number of state-of-the-art anomaly detection methods substantially. The main difference between PyOD and the Scikit-Learn methods is that PyOD methods are largely designed with multivariate data in mind. Also, the latter provides Deep learning-based models such as Auto-Encoders, Variational Auto-Encoders as well as Single and Multi-Objective Generative Adversarial Active Learning. These Deep learning methods require the integration of TensorFlow/Keras.

Furthermore, with the integration of the PySAD[27] library streamed (incremental learning) anomaly detection methods will also be included. PySAD also allows the inclusion of batch anomaly detection methods from PyOD for stream processing by enabling the definition of micro-batches (similar to how Spark Streaming works).

[25] https://scikit-learn.org/stable/
[26] https://pyod.readthedocs.io/en/latest/
[27] https://pysad.readthedocs.io/en/latest/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

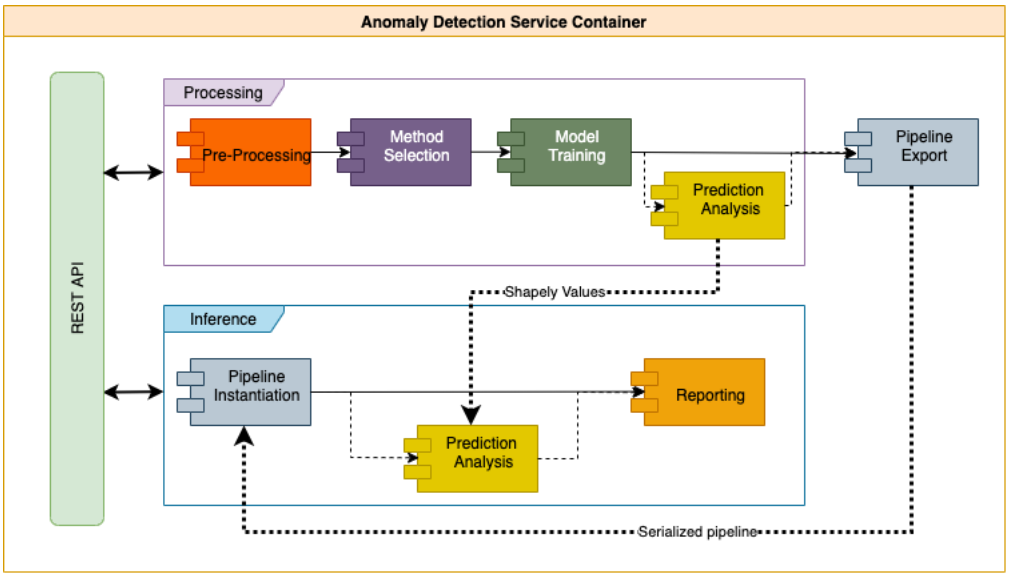

Figure 15: Anomaly Detection Service Component

In the case of the explainable AI methods used we propose the SHAP[28] library which utilizes Shapely values to explain anomalous event selection. These Shapely values can be then used to calculate global or event-based feature importance. Feature importance can then be added to the reporting mechanism. Calculating Shapely values is a computationally expensive task thus approximations are usually used, which is the case for the SHAP library. For the first iteration of this, we will focus on simple Shapely Value calculation, further development and optimization will be necessary for full integration.

All functionality of this component will be contained in a dockerized deployment containing a REST API for configuration and querying. Using this API users will be able to both setup the creation of a new anomaly detection methods and the exporting of trained predictive models. The exported model will not only consist of the pretrained model but a partial or even full serialization of the component processing pipeline. The most likely candidate for serialization is the cloudpickle[29] library. This is necessary because most models will be firmly dependent on the operations applied to the raw data (ex. Normalization). We should note that preprocessing can be done also externally, by calling services from other components from BD4NRG solution. The final split of which pre-processing capabilities will be fully integrated in this component and which will be used from external sources is still to be decided. An initial architecture can be seen in Figure 15.

At a later stage, transprecise optimization methods can also be included for anomaly detection method creation. By transprecise adaptation includes some optimization methods which can give a workable tradeoff between computational requirements and model inference performance (ex. Inference time and precision). This will allow for more flexibility in the deployment of this component on Edge/Fog type scenarios. Moving the predictive model on hardware with low computational power while still being able to provide workable insight into the available data.

[28] https://github.com/slundberg/shap
[29] https://github.com/cloudpipe/cloudpickle

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

5.3 Functional Specification

The present section recaps relevant objectives and functionalities of this particular module, as well as provides a hint into the actors that may be involved.

In addition, it offers an initial collection of requirements and user stories.

5.3.1 Anomaly Detection Service

There are several subcomponents in the Anomaly Detection Service component. The first proposed specification will be detailed in the following paragraphs. All subcomponents can be configured using a REST API. These subcomponents are not to be considered as a final, they are subject to modification/extension until the end of the project.

- **Data pre-processing** component is responsible for all raw data pre-processing steps. These can take the form of simple filtering, pivoting, windowing and data normalization. In the future this will probably leverage other tools from BD4NRG solution, so that no pre-processing operations have to be included in this component.
- **Method Selection** components is responsible for the anomaly detection method to be created. All selection will take the form of JSON formatted calls to the provided REST API. We should mention that in the case of stream anomaly detection methods, no separate Inference pipeline is required, only the definition of a polling window of the data in pre-processing.
- **Model Configuration and training** component is responsible with hyper-parameter configuration for the selected methods. For some anomaly detection methods such as Deep Learning based ones, different method topological structures can be defined. Later version can also include transprecise optimization methods.
- **Prediction Analysis** component is responsible with the calculation of Shapely values which can be used to explain why a particular event has been marked as anomalous. As mentioned before, it can give an estimation of global or event-based feature importance. Seeing as this method is difficult to compute it is not a mandatory part of the Processing pipeline, if it is defined via the REST API it will be appended and serialized together with the predictive model and pre-processing steps.
- **Pipeline Export** component is responsible with serializing the processing pipeline which can then be instantiated by the **Inference Pipeline.** Thus, it is possible to create several anomaly detection pipelines and then instantiate them as needed via the REST API.

5.3.2 Requirements

Requirement #	Description
UM_1	The Marketplace module shall manage the users and roles.
ADS_1	The Marketplace module shall manage the users and roles. Three categories of users exist: Owner and Guest. These are their capabilities:

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

	• Guests have only read access and cannot modify anything • Owner/s have access to pipeline configuration, can see available pipelines and the type of data they are trained for.
ADS_2	Component must have access to raw data source, both streaming and historical data.
ADS_3	Must have mechanism for saving and referring to serialized processing pipeline.
ADS_4	Must provide specialized hardware access, General-Purpose Graphics Processing Unit (GPGPU), for the fast/optimized execution of both processing and inference pipeline instantiation.
ADS_5	Must have access to anomaly reporting mechanism.
ADS_6	Raw data should be representable in tabular and time-series format.

5.3.3 User Stories

These user stories describe how a user typically uses the Anomaly Detection Service component provided by TS through the BD4NRG Marketplace. The user stories reflect only the initial usage scenario, it should not be considered as a final version.

Ref. No.	EPIC	As a (stakeholder)	I want to	So that	And is considered 'done' when	MoSCoW [30]
UM-1	1	Data Provider	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Provider User and Register an Application.	I have received the confirmation of a verified account as part of the BD4NRG system and the API key.	Must
AR-2	1	Data Consumer	Register an account on BD4NRG Marketplace.	I may access the BD4NRG platform as a Data Consumer	I have received the confirmation of a verified account as part of the BD4NRG	Must

[30] Must, Should, Could, or Won't

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

				User Register an Application.	system and the API key.	
ADS_1	1	Owner	Select raw data source.	I may use BD4NRG platform data sources.	I have received confirmation of granted access and API data source clearance (key).	Must
ADS_2	1	Owner	Setup an anomaly detection pipeline.	I may use BD4NRG platform for creating use-case specific anomaly prediction models (both batch and stream-based models).	I have received confirmation that model training has been completed.	Must
ADS_3	1	Owner	Start prediction analysis using Shapely values.	I may get Shapely value-based root cause analysis report.	I have received confirmation that Shapely value computation is complete.	Should
ADS_4	1	Owner	Start pipeline export.	I may reuse trained pipeline on new incoming data, or in the case if streamed anomaly detection anomalous instances.	I have received confirmation and have access/reference of the serialized detection pipeline, including shapely values if calculated.	Must

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

ADS_5	2	Guest	Select pretrained detection pipeline.	I may get access to pretrained anomaly detection pipeline.	I have received confirmation that anomaly detection pipeline has been instantiated and started checking for anomalous events.	Must

5.3.4 Interfaces

Interfaces that the component exposes:

Interface #	Description
ADS_1	REST API for definition of raw data source
ADS_2	REST API for definition of processing pipeline; pre-processing, method selection, model training, prediction analysis and export
ADS_3	REST API for definition of inference pipeline.
AIDS_4	Reporting mechanism via REST API and other BD4NRG data sharing mechanisms.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

6 Conclusion

This document describes the first technology release of the main components of the BD4NRG Marketplace developed in the context of WP5.

The document focuses on the description of the component architecture, the specification of the implementation and the deployment, and describe the first functional specification of the components.

The main component of the Marketplace includes the specification of the web applications and the specification of the backend components that integrates the functionality of the BD4NRG tools and services developed in WP3 and WP4.

The Utilities Sandboxes component is also described in this document specifying the model-based simulated data on Hardware-In-the-Loop facilities in the context of WP5. The component is integrated in the Marketplace by means of the API and providing a unified web interface.

The complete specification of the technology releases of the BD4NRG Marketplace will be completed, extended, and refined in the future in document D5.3 (Second Technology Release) in M21 and D5.4 (Final Technology Release) in M28 in an iterative and incremental mode according to the progress in the definition and specification of the different tools and services developed for the BD4NRG framework taking into account the requirements and specifications of the pilots in the role of the different stakeholders of the BD4NRG project.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

References

[1] ATOS, «GitLab Marketplace Backend,» [En línea]. Available: https://scm.atosresearch.eu/ari/bd4nrg-backend.

[2] ATOS, «GitLab Marketplace Frontend,» [En línea]. Available: https://scm.atosresearch.eu/ari/bd4nrg-frontend.

[3] ATOS, «GitLab Marketplace Environment,» [En línea]. Available: https://scm.atosresearch.eu/ari/bd4nrg-environment.

[4] «Platone,» [En línea]. Available: https://www.platone-h2020.eu/.

[5] «Ethereum,» [En línea]. Available: https://en.wikipedia.org/wiki/Ethereum.

[6] «ERC Specification,» [En línea]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md.

[7] «ERC 721 Specifications,» [En línea]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721.md.

[8] «Oauth 2.0,» [En línea]. Available: https://oauth.net/2/.

[9] RWTH, "FEIN Website," [Online]. Available: https://www.fein-aachen.org/projects/villas-framework/.

[10] RWTH, "Lab exercises," [Online]. Available: https://villas.fein-aachen.org/doc/node-guide.html.

[11] RWTH, "Git doc source," [Online]. Available: https://git.rwth-aachen.de/acs/public/villas/documentation.

[12] P. A. M. RWTH, "VILLAS framework concept - Slide share slides," [Online]. Available: https://www.slideshare.net/SteffenVogel1/villasframework-concept.

[13] RWTH, "VILLAS Data Gateway Component," [Online]. Available: https://villas.fein-aachen.org/doc/node.html.

[14] RWTH, "VILLAS Web Interface Component," [Online]. Available: https://villas.fein-aachen.org/doc/web.html.

[15] RWTH, "VILLAS Control Component," [Online]. Available: https://villas.fein-aachen.org/doc/controller.html.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.