ATTEST

Ref. Ares(2022)8624442 - 12/12/2022

WP6

ICT Platform to enhance TSO/DSO coordination

TSO/DSO data access and storage component

D6.1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 864298.

DOCUMENT CONTROL PAGE

DOCUMENT	D6.1 TSO/DSO data access and storage component
TYPE	Other
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	30 / 10 / 2022
DATE OF DELIVERY	30 / 11 / 2022
VERSION	V0.8
DELIVERABLE RESPONSIBLE	SOFTLAB
AUTHOR(S)	Clara Bagnasco, Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato
OFFICIAL REVIEWER(S)	Filipe Soares

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CHANGES
0.1	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	11/10/2022	First version of D6.1: index
0.2	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	23/10/2022	Introduction
0.3	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	15/11/2022	First version of the data model and services
0.4	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	22/11/2022	Updated version of the data model and services, plus the data tables, Importers/exporters
0.5	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	25/11/2022	Importers/exporters
0.6	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	29/11/2022	Added a detailed services list in the appendixes
0.7	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	30/11/2022	Added an Installation guide in the appendices.
0.8	Christian Biasuzzi, Davide Longo, Massimo Ferraro, Lara Pinato	30/11/2022	Fixes. Version to be reviewed.
1.0			Final version

Table of Contents

List of Tables

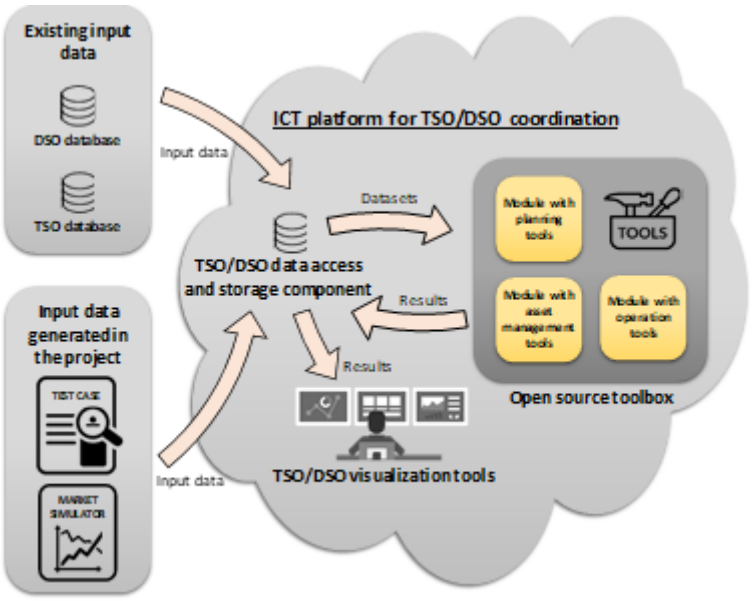
Abbreviations and Acronyms

API	Application Programming Interface
DB	Database
CIM	Common Information Model
CRUD	Create-Read-Update-Delete
CSV	Comma-Separated Values
ER	Entity-Relationship
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
JDK	Java Development Kit
JPA	Java Persistence API
JSON	JavaScript Object Notation
JWT	JSON Web Token
MVC	Model-View-Controller
ODS	Open Document Sheet
RDBMS	Relational Database Management System
REST	Representational State Transfer
SLD	Single Line Diagram
SVG	Scalable Vector Graphics

1. Introduction

The ATTEST project developed a modular open-source toolbox: a suite of tools to support TSOs/DSOs in operating, maintaining and planning the energy systems.

The open-source toolbox has been embedded into an ICT platform that provides data access connectors and converters, tools orchestration functionalities and visualization interfaces, as shown in the picture below.



Inside the ICT platform, the common TSO/DSO data access and storage component allows storing and accessing all the networks information from the TSO/DSO data sources and data from the electricity markets. This component's services prepare the tools' inputs and store their outputs into its internal storage. The component also provides data conversion functionalities to handle the different formats on which the tools rely.

This document accompanies the release of the software implementing the TSO/DSO data access and storage component, providing information about it:

- Section 2 outlines the data model used in the database for storing network, market and tools data.
- Section 3 describes importers and exporters implemented for loading into the database networks and other support data, and for preparing the input of the tools.
- Section 4 reports some information about the services that allow to access and manage the data stored in the database.
- Section 5 speaks about the identity manager used for handling user and user's services.
- Section 6 provides some information about the technologies used for the implementation of the component.
- The appendices contain the installation guide for the whole ICT platform (including the data storage component), a detailed list of the data model tables, and a detailed list of REST services exposed by the platform.

2. Data model

The ICT platform's data model has been designed to represent data relevant to ATTEST: e.g., network, forecasts, profiles, tools inputs and outputs. The model covers also data used by the ICT components: e.g., tasks and simulations.

2.1. Technical description

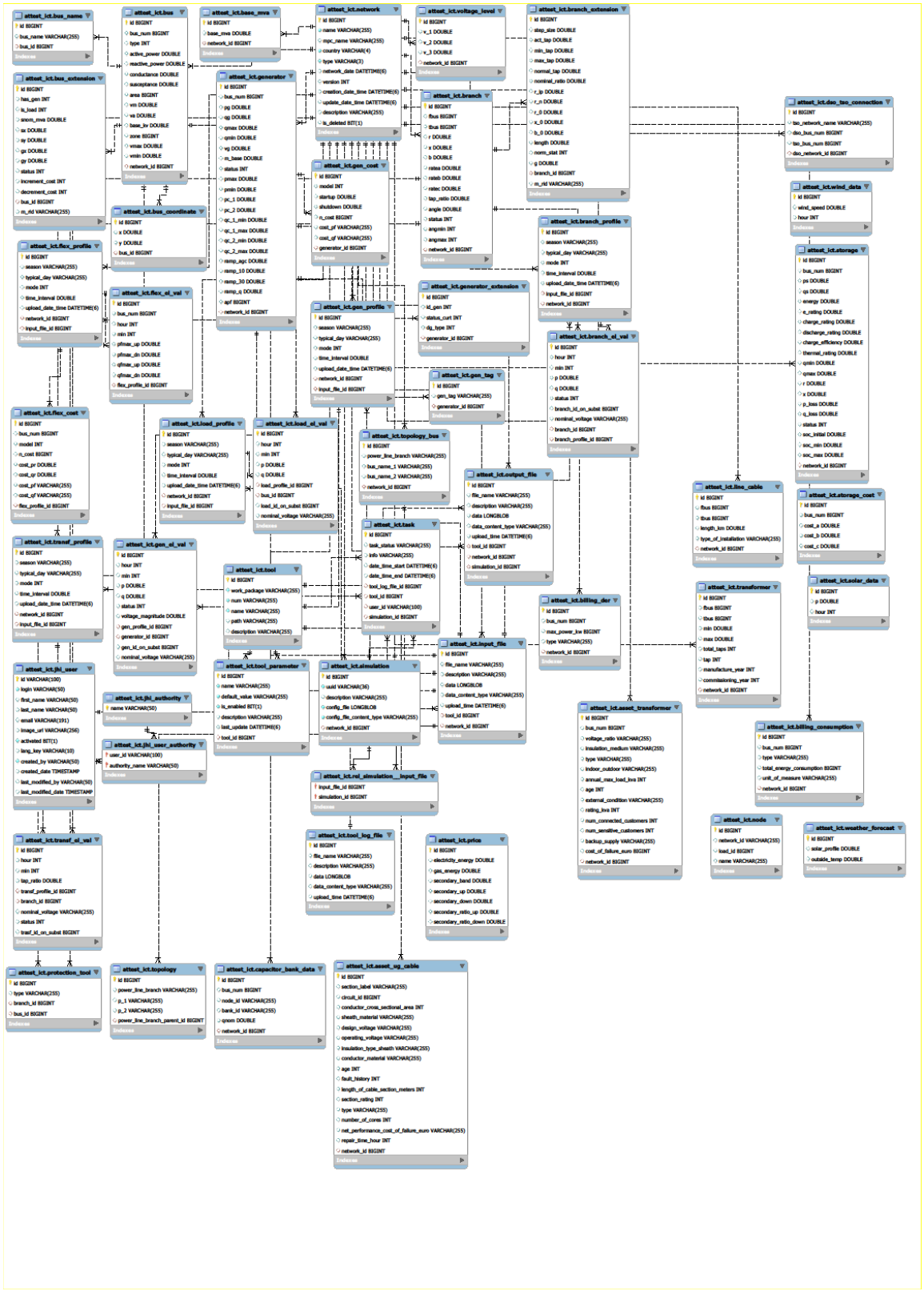
As the tools in the open-source toolbox require the bus-branch model of a network, in a MATPOWER[1] compatible format, the ICT model's network section follows in large part the MATPOWER specification[2] and includes tables representing buses, branches, generators, etc. Profiles, market, and flexibility data are part of the model's auxiliary data section. Tool related data (e.g., inputs and outputs), tasks, users and other support data are also part of the ICT model.

The standard CIM model, normally used by the operators to represent their networks, is not included in the ICT model, because it is not directly used by the tools. However, to provide a bridge between ATTEST and the external world, CIM models are stored in an external CIM repository component. A specialized converter component takes care of the conversions from the CIM repository to the ICT model. More details on this component can be found in D4.4 "Tool for State Estimation of Distribution Networks" and the connection with the ICT model will be included in D6.2 "Integration of the open-source toolbox".

The ICT platform's data model is relational and has been instantiated in a relational database. The following section includes the ER model diagram. A detailed description of the tables can be found in the appendix "ICT Model database tables".

[1] https://matpower.org/doc/
[2] https://matpower.org/doc/manuals/

2.2. ER model diagram



3. Data importers and exporters

The test cases prepared for the ATTEST project have the networks in .m text files in MATPOWER format and the auxiliary data in CSV and XLS files.

The tools in the open-source toolbox want their inputs in specific formats, e.g., network as .m text files in MATPOWER format, auxiliary data as .xlsx Excel files or .ods OpenDocument Spreadsheet files.

A set of dedicated importers and exporters has been therefore developed in the ICT platform to convert data from the original test cases to the ICT database and from the ICT database to the formats expected by the tools.

3.1. Importers

- from .m(MATPOWER) to ICT database: used to import network data to the ICT database. To implement this importer, parsing and reading data from the text files, it was used the uniVocity[3] library.
- from .ods (OpenDocument Sheet) to ICT database: used to import network data, enriched for specific tools (e.g., T4.1), to the ICT database. To implement this importer, accessing to the OpenDocument sheets and data, it was used a customized version of the open-source SODS library[4].
- from .xlsx to ICT database, used e.g., to import profiles and flexibilities data. The implementation of the importer uses the Apache POI library[5] for accessing the Excel sheets and data.
- from .csv to ICT database, used e.g., to import profiles and flexibilities data. The implementation of the importer uses the uniVocity library[6] for parsing and reading data from the CSV files.

3.2. Exporters

- From ICT database to .m (MATPOWER) used to export a network stored into the DB to a .m file to be used by the tools. The exporter implementation uses the uniVocity library.
- From ICT database to .ods (OpenDocument Sheet): The exporter implementation uses the SODS library.
- From ICT database to SVG, used to render the SLD diagram of a network. the SLD diagram is then displayed by the SLD visualization component in the interface layer (please see deliverable D6.3 for details). The ICT model of the network, read from the database, is first converted to an intermediate IIDM model[7], subsequently used to feed the SVG converter. The SVG converter has been implemented with the PowSyBl Diagram library[8].

The converters are exposed as REST Web Services among the ICT services (see next section); the services are described in the Appendix: Services.

[3] https://github.com/uniVocity/univocity-parsers
[4] https://github.com/miachm/SODS
[5] https://poi.apache.org/
[6] https://github.com/uniVocity/univocity-parsers
[7] https://www.powsybl.org/pages/documentation/grid/model/
[8] https://github.com/powsybl/powsybl-diagram

4. Services

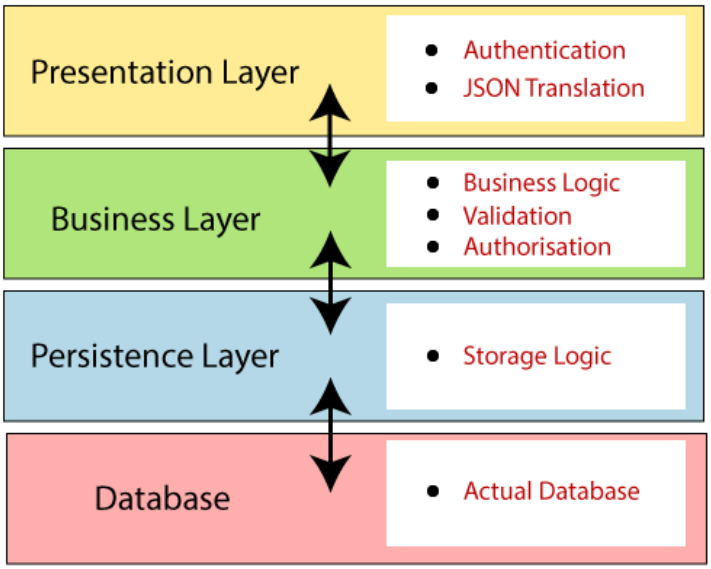
The access and storage component provides services that implement a layer on top of the database, allowing to access and manage the data stored.

The services are implemented as REST Web Services, enabling the execution of different operations on the data: GET methods for retrieving information, and POST, PUT, PATCH and DELETE methods for modifying it.

JSON is the open standard used as data interchange format.

The services are protected through authentication and secure protocols, with an access token and using JWT[9] standard.

The access and storage component, for providing the services, uses a 4 layered architecture, depicted below, where each layer communicates with the layer directly below or above it (hierarchical structure).



The Presentation Layer handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer.

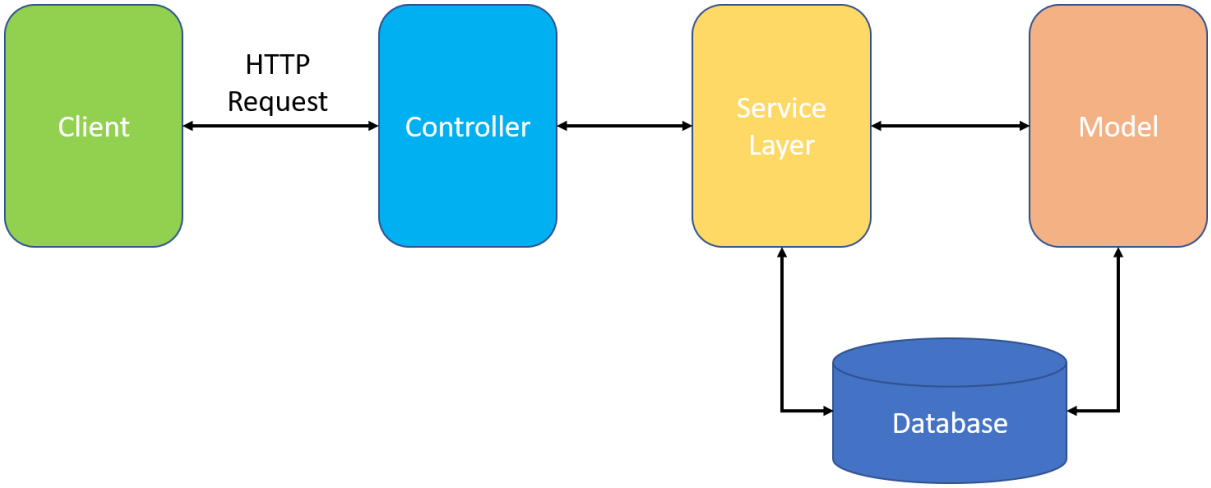
The Business Layer handles all the business logic: it consists of service classes and uses services provided by data access layers. It also performs authorization and validation.

The Persistence Layer contains all the storage logic and translates business objects from and to database rows.

The Database Layer performs CRUD (create, retrieve, update, delete) operations on the database data.

The flow architecture of the services is represented in the picture below.

[9] https://jwt.io/



An HTTP request, coming from a client, is handled by a Controller, that implements the REST Web Services and maps that request. The Controller, as necessary, calls the Service Layer, in charge of managing all the business logic. A Model, used by the Service Layer, maps the data in the database.

The developed services are grouped into these main areas:

- Data store management: networks, auxiliary data: these services include importers, exporters, converters and other generic read and write services operating on the DB entities, and are used to feed the ICT database and to retrieve data from it. Some of them provide custom views on the model specifically designed to support the visualization components described in deliverable D6.3.
- Tools and simulations management: these services target the execution of the tools from the opensource toolbox. To each tool corresponds a dedicated service that works as a wrapper for the tool: it prepares the input data for the tool, it executes the tool, retrieve its results, and finally stores the results in the ICT database.

A detailed list of the most significant services can be found in the Appendix ICT services.

5. Identity management

The ICT data access layer integrates the popular open-source Keycloak[10] as the user management component. Keycloak, handles users and roles and provides the authorization and authentication features used, for example, by the ICT frontend to secure access to the platform.

6. Technical implementation details

The ICT platform (data access layer, services layer) has been developed using Java[11] and related technologies, in particular:

- Spring Boot[12] for application configuration
- Maven[13] configuration for building, and running the application

[10] https://www.keycloak.org/
[11] https://openjdk.org/projects/jdk/11/
[12] https://projects.spring.io/spring-boot/
[13] https://maven.apache.org/

- Spring Security[14]
- Spring MVC REST[15] and Jackson[16]
- Spring Data JPA[17]
- Database updates with Liquibase[18]

The RDBMS used to store the ICT model is MySQL[19].

[14] https://docs.spring.io/spring-security/site/index.html
[15] https://spring.io/guides/gs/rest-service/
[16] https://github.com/FasterXML/jackson
[17] https://projects.spring.io/spring-data-jpa/
[18] http://www.liquibase.org/
[19] https://dev.mysql.com/

7. Appendices

7.1. Installation guide

The ICT platform includes different software components: the database, the ICT server and frontend application, the Identity manager. Besides, to be able to run the tools, the installation of the open-source toolbox is also required.

This guide describes in detail the installation of the ICT platform from the platform source code.

The installation of the open-source toolbox and its requirements will be detailed in deliverable D6.2: Integration.

7.1.1. Requirements

Please note that the Installation of some of these components requires an Administrator's privileges.

7.1.1.1. Database

MySQL is the RDBMS implementation used for the ICT platform. MySQL can be downloaded here https://dev.mysql.com/downloads/installer/.

Throughout the development it was used MySQL v8.0.27; However, more recent versions (or even older versions) should work, as well.

7.1.1.2. Java

the ICT server and frontend component was implemented using java-related technologies, therefore a JDK installation is needed.

The version of the JDK used throughout the project is 11.0.2 and can be downloaded here: https://jdk.java.net/java-se-ri/11

7.1.1.3. KeyCloak

The identity manager used in the project is KeyCloak v15.0.2. It can be downloaded here: https://www.keycloak.org/archive/downloads-19.0.2.html

7.1.1.4. ICT source code

The ICT server and frontend source code can be downloaded here https://github.com/ATTEST-project/attest-ict

(Menu Code/Download ZIP).

The customized SODS library can be downloaded here https://github.com/ATTEST-project/sods

(Menu Code/Download ZIP).

Caveat: all the components in the requirement list are multiplatform. However, this installation guide targets a MS WINDOWS server, since most of the tools in the opensource toolbox were developed in WINDOWS OS. Making all the tools work in a different environment is surely possible but it could require some effort and time.

7.1.2. Installation

Please refer to the installation guides for MySQL, JDK and KeyCloak on their home sites. The installation of the JDK is required to proceed with the next step.

To install ICT from its sources, unzip the attest-ict and sods .zip archives in c:\ATTEST\sources

Note: in this guide the default target installation root directory, is c:\ATTEST. The settings defined in the default configuration files were chosen to minimize the number of changes required to run the ICT platform. It is, of course, possible to install the software to other folders; in that case, however, more changes to the configurations would be required.

1. Open a CMD terminal in the c:\ATTEST\sources\sods, and enter the command
   `mvnw clean install`
2. Open a CMD terminal in the c:\ATTEST\sources\attest-ict, and enter the command
   `mvnw clean package -DskipTests -Dmodernizer.skip`
3. Once the latter building process has finished,
   - copy the file target/attest-ict-x.y.z.jar to the directory c:\ATTEST\ICT-Platform, and rename it to AttestICT.jar
   - copy the files src\main\resources\config*.yml to the directory c:\ATTEST\ICT-Platform\configs

7.1.3. Configuration

7.1.3.1. MySQL

Open a connection to MySQL (an admin user with enough permissions to create users and new DBs is required)

7.1.3.2. Create a new ATTEST MySQL user

```
CREATE USER 'attest_ict_user'@'%' IDENTIFIED BY 'attest_ict_user';

SET password for 'attest_ict_user'@'%' ='MYSQL_ADMIN_PWD';
```

7.1.3.3. Create a new ATTEST DB

```
DROP DATABASE IF EXISTS attest_ict;

CREATE DATABASE IF NOT EXISTS attest_ict CHARACTER SET utf8 COLLATE
utf8_unicode_ci;

GRANT ALL ON attest_ict.* TO 'attest_ict_user'@'%';
```

7.1.4. KeyCloak

7.1.4.1. Edit the configuration file:

<ATTEST_KEYCLOAK_INSTALLATION>\standalone\configuration\standalone.xml

(where <ATTEST_KWYCLOAK_INSTALLATION> is KeyCloak's installation directory, on WINDOWS it could be usually found in C:\Program Files (x86)\Keycloak\keycloak-15.0.2)

and change this setting:

```
<socket-binding name="http" port="${jboss.http.port:8080}"/>
```

To:

```
<socket-binding name="http" port="${jboss.http.port:8180}"/>
```

7.1.4.2. Start the KeyCloak server

open a CMD terminal in the <ATTEST_KEYCLOAK_INSTALLATION> directory and enter

```
bin\standalone.bat
```

Once the KeyCloak server has started, connect to the KeyCloak admin console. Point the browser to http://localhost:8180/auth/admin

Push the Add realm button from the top-left realm menu (the default is Master). Import file <ATTEST_SOURCES>\src\main\docker\realm-config\jhipster-attest-realm.json

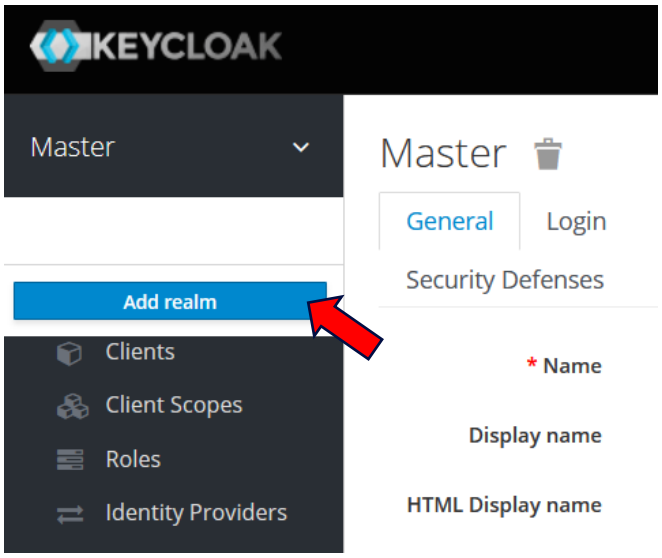


FIGURE 1- KEYCLOAK: CREATE A NEW REALM

Once the realm configuration is complete, select the new realm jhipster-attest from the top-left realm menu, then select the Manage/import menu, point to the file

<ATTEST_SOURCES>\src\main\docker\realm-config\jhipster-users-0.jsonn

Make sure that the attribute “Direct Access Grant Enable” is set to On, in the Clients/web_app section

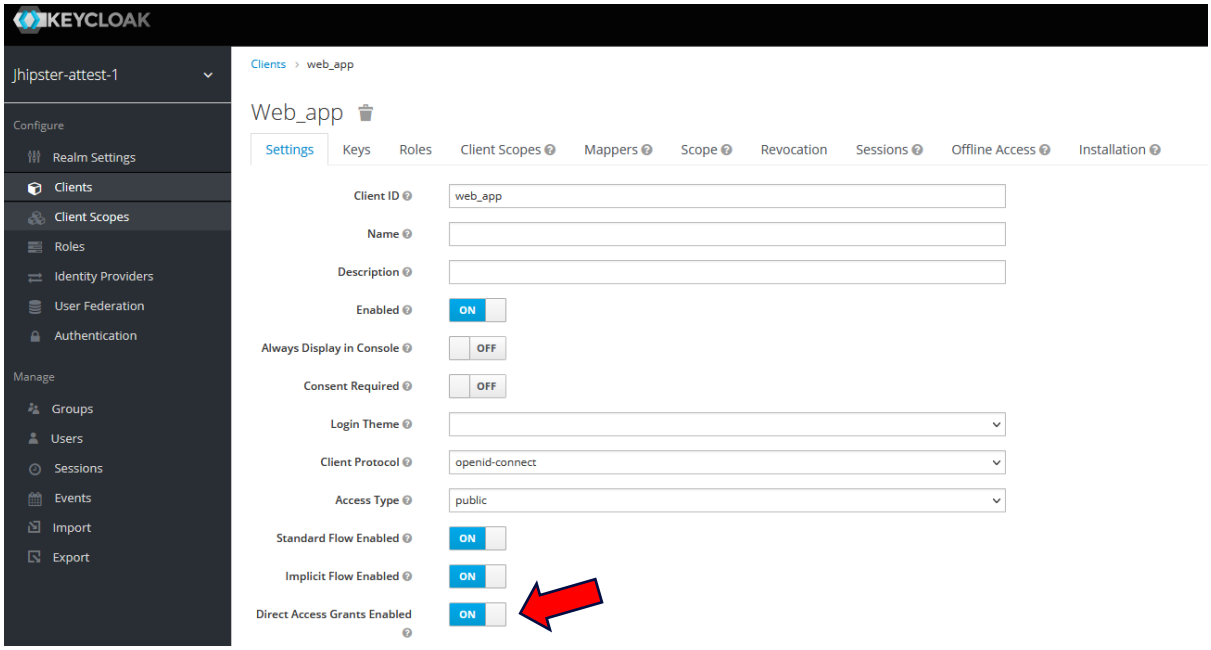


FIGURE 2 - KEYCLOAK CLIENT SETTINGS

Note: KeyCloak must be re-started, for example after a reboot. A restart does not need the initial realm-import configuration: just open a CMD terminal in the <ATTEST_KEYCLOAK_INSTALLATION> directory and enter the command

```
bin\standalone.bat
```

7.1.4.3. Configuring KeyCloak as a WINDOWS service

To avoid a manual start, KeyCloak could be configured as a WINDOWS service (Administrato privileges are required): copy the directory

<ATTEST_KWYCLOAK_INSTALLATION>\docs\contrib\scripts\service

to

<ATTEST_KWYCLOAK_INSTALLATION>\bin,

then open a CMD terminal in the <ATTEST_KEYCLOAK_INSTALLATION>\bin\service directory and enter the command

```
service.bat /name keycloak /display Keycloak /desc "Keycloak Http Server"
```

7.1.4.4. Create new ATTEST users

Point the browser to the admin console: http://localhost:8180/auth/admin

Select the ATTEST realm, then select the Manage/Users menu and push the “Add user” button

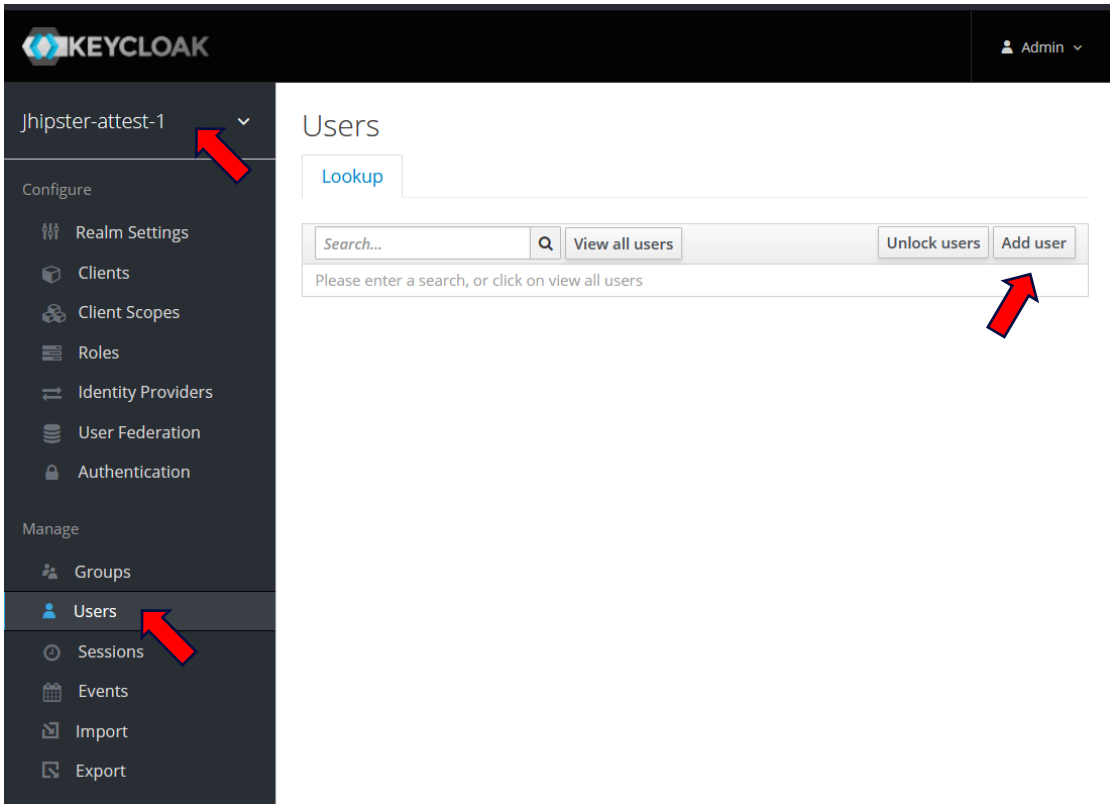


FIGURE 3-KEYCLOAK: CREATE NEW USERS

Then, fill the form with the new user data and push the "Save" button. Please remember to set the attribute "Email Verified" to "On".

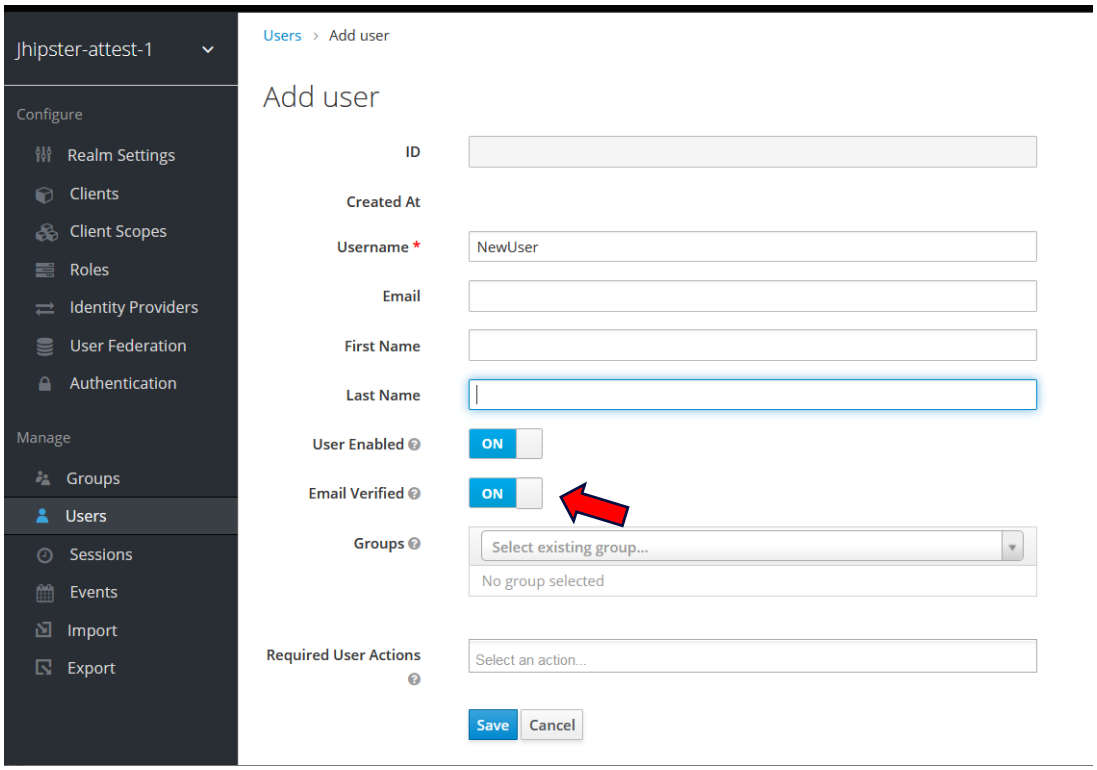


FIGURE 4 – KEYCLOAK: INSERT NEW USER DATA

Roles are assigned to a user in the "Role mapping" tab. A ROLE_USER role is needed to access the ATTEST frontend.

7.1.5.ICT-platform

Edit the file

c:\ATTEST\ICT-Platform\configs\application-dev.yml

and change the datasource section to suit the local installation:

```
datasource:
    type: com.zaxxer.hikari.HikariDataSource
    url:
jdbc:mysql://localhost:MYSQL_PORT/attest_ict?useUnicode=true&c
haracterEncoding=utf8&useSSL=false&useLegacyDatetimeCode=false
&serverTimezone=UTC&createDatabaseIfNotExist=true
    username: attest_ict_user
    password: MYSQL_ATTEST_USER_PASSWORD
```

MYSQL_PORT and MYSQL_ATTEST_USER_PASSWORD must reflect the MySQL installation and the ATTEST user defined in the "Create a new ATTEST MySQL use" section.

Note that in a standard MySQL installation, MYSQL_PORT is usually 3306. In this example a local MySQL installation is assumed; Of course, it is also possible to use a remote MySQL using a server name instead of "localhost", in the url attribute.

7.1.6.Starting the ICT-platform

Open a CMD terminal in the c:\ATTEST\ICT-Platform directory and enter the command:

```
java -Xmx512m -jar AttestICT.jar --spring.config.location=./configs/
```

Point a browser to the address below, to access the ICT Platform frontend application:

http://localhost:9090

Note that the default port 9090 can be changed in the ICT configuration file, described above (server.port attribute).

Please refer to the D6.3 for a description of the ICT Platform frontend and its components.

7.2. ICT Model database tables

In this chapter are listed all the tables related to the main ATTEST objects in the model

7.2.1. Networks

This table contains networks, identified by unique network ids and names. Other optional attributes can be used to further characterize a network: a date, a version number, a type, etc.. All buses, branches, generators and other data tables refer to an entry in the *network* table.

Table Name: **network**

NAME	TYPE	DESCRIPTION
Id	Number	Table's primary key
Name	String	Name of the network
mpc_name	String	MATPOWER case name
Country	String	Country: • PT: Portugal • ES: Spain • HR: Croatia • UK: United Kingdom
Type	String	Network type: • DX: distribution network • TX: transmission network
network_date	DateTime	Date of network
Version	Number	Version of the network
creation_datetime	DateTime	Date of creation
update_datetime	DateTime	Date of update
description	String	Additional information
isDeleted	Boolean	Default value is false

TABLE 1 NETWORKS

7.2.2. MATPOWER model

This section lists the tables that model the MATPOWER format for the D2.3's .m files, used in ATTEST. Some tables (e.g., Base MVA, Bus, Branch and Generator) refer to the *network* table, others are specific attributes for Bus, Branch or Generator. These tables correspond to the same-name structures of a MATPOWER case file format (see Appendix B – Data File Format of MATPOWER Manual[20]l): base_mva, bus, branch,generator, bus_name, gen_cost.

7.2.2.1. Base MVA

The value of Base MVA of a network.

[20] https://matpower.org/doc/

Table Name: **base_mva**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
base_mva	Number	Value of Base MVA of the network
network_id	Number	Refers to Network table

TABLE 2 BASE MVA

7.2.2.2. Bus

The buses that exist in the network are saved in this table.

Table Name: **bus**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_num	Number	Bus number
type	Number	Bus type: • 1 = PQ • 2 = PV • 3 = ref • 4 = isolated
area	Number	Area number
vm	Number	Voltage magnitude (p.u.)
va	Number	Voltage angle
base_kv	Number	Base voltage (kV)
zone	Number	Loss zone
vmax	Number	Maximum voltage magnitude (p.u.)
vmin	Number	Minimum voltage magnitude (p.u.)
active_power	Number	Real power demand (MW)
reactive_power	Number	Reactive power demand (MWAr)
conductance	Number	Shunt conductance
susceptance	Number	Shunt susceptance
network_id	Number	Refers to Network table

TABLE 3 BUS

7.2.2.3. Bus Extension

Table Name: **bus_extension**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key

has_gen	Number	If bus has generator
is_load	Number	If bus is a load
sx	Number	
sy	Number	
gx	Number	
gy	Number	
status	Number	Status of a bus
increment_cost	Number	Increment Cost
decrement_cost	Number	Decrement Cost
bus_id	Number	Refers to Bus
m_rid	String	External id
Snom_mva		

TABLE 4 BUS EXTENSION

7.2.2.4. Branch

The branches that exist in the network are saved in this table.

Table Name: **branch**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
fbus	Number	"from" bus number
tbus	Number	"to" bus number
r	Number	Resistance (p.u.)
x	Number	Reactance (p.u.)
b	Number	Total line charging susceptance
ratea	Number	MVA rating A (0 = unlimited)
rateb	Number	MVA rating B (0 = unlimited)
ratec	Number	MVA rating C (0 = unlimited)
tap_ratio	Number	Transformer off nominal turns ratio
angle	Number	Transformer phase shift angle
status	Number	Initial branch status: • 1 = in-service • 0 = out-of-service
angmin	Number	Minimum angle difference
angmax	Number	Maximum angle difference
network_id	Number	Refers to Network table

TABLE 5 BRANCH

7.2.2.5. Branch Extension

Table Name: **branch_extension**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
step_size	Number	
act_tap	Number	
min_tap	*Number*	
max_tap	*Number*	
normal_tap	*Number*	
nominal_ratio	*Number*	
r_ip	*Number*	
r_n	*Number*	
r_0	*Number*	
x_0	*Number*	
b_0	*Number*	
length	*Number*	*Length of a branch*
norm_stat	*Number*	
g	*Number*	
branch_id	*Number*	*Refers to Branch*
m_rid	*String*	External id

TABLE 6 BRANCH EXTENSION

7.2.2.6. Generator

The generators belonging to the network are saved in this table.

Table Name: **generator**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_num	Number	Bus number where generator is connected to
pg	Number	Real power output (MW)
qg	Number	Reactive power output (MWAr)
qmax	Number	Maximum reactive power output (MWAr)
qmin	Number	Minimum reactive power output (MWAr)
vg	Number	Voltage magnitude setpoint (p.u.)
m_base	Number	Total MVA base of machine
status	Number	Machine status: • > 0 = machine in-service • ≤ 0 = machine out-of-service

pmax	Number	Maximum real power output (MW)
pmin	Number	Minimum real power output (MW)
pc_1	Number	Lower real power output of PQ capability curve (MW)
pc_2	Number	Upper real power output of PQ capability curve (MW)
qc_1_min	Number	Minimum reactive power output at PC1 (MVAr)
qc_1_max	Number	Maximum reactive power output at PC1 (MVAr)
qc_2_min	Number	Minimum reactive power output at PC2 (MVAr)
qc_2_max	Number	Maximum reactive power output at PC2 (MVAr)
ramp_agc	Number	Ramp rate for load following/AGC (MW/min)
ramp_10	Number	Ramp rate for 10 minute reserves (MW)
ramp_30	Number	Ramp rate for 30 minute reserves (MW)
ramp_q	Number	Ramp rate for reactive power (MVAr/min)
apf	Number	Area participation factor
network_id	Number	Refers to Network table

TABLE 7 GENERATOR

7.2.2.7. Generator Extension

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
id_gen	Number	ID of generator
status_curt	Number	
dg_type	Number	
generator_id	Number	Refers to Generator

TABLE 8 GENERATOR EXTENSION

7.2.2.8. Bus Name

This table is used to map the names of the buses to the ids of the buses.

Table Name: **bus_name**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_name	String	Bus name
bus_id	Number	Refers to Bus table

TABLE 9 BUS NAME

7.2.2.9. Bus Coordinate

Table that represents coordinates of each bus of the network.

Table Name: **bus_coordinate**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
x	Number	X coordinate
y	Number	Y coordinate
bus_id	Number	Refers to Bus table

TABLE 10 BUS COORDINATE

7.2.2.10. Generator Tag

Identifiers that indicate generation technology type of generators.

Table Name: **gen_tag**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
gen_tag	String	Generator tag name
generator_id	Number	Refers to Generator table

TABLE 11 GENERATOR TAG

7.2.2.11. Generator Cost

This table saves the Generator Cost Data.

Table Name: **gen_cost**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
model	Number	Cost model: • 1 = piecewise linear • 2 = polynomial
startup	Number	Startup cost
shutdown	Number	Shutdown cost
ncost	Number	Number N = n + 1 of data points

cost_pf	String	
cost_qf	String	
generator_id	Number	Refers to Generator table

TABLE 12 GENERATOR COST

7.2.2.12. Transformer

The transformers that exist in the network are saved in this table.

Table Name: **transformer**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
fbus	Number	"from" bus number
tbus	Number	"to" bus number
min	Number	
max	Number	
total_taps	Number	
tap	Number	
network_id	String	Refers to Network table
manufacture_year	Number	
commissioning_year	Number	

TABLE 13 TRANSFORMER

7.2.2.13. Voltage Level

The voltage levels (kV) of the network.

Table Name: **voltage_level**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
v1	Number	Voltage Level value 1
v2	Number	Voltage Level value 2
v3	Number	Voltage Level value 3
network_id	Number	Refers to Network table

TABLE 14 VOLTAGE LEVEL

7.2.2.14. Capacitor Bank Data

The capacitor bank data of the network are saved in this table.

Table Name: **capacitor_bank_data**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_num	Number	Bus number
node_id	String	Node identifier
bank_id	String	Bank identifier
qnom	Number	
network_id	Number	Refers to Network table

TABLE 15 CAPACITOR BANK DATA

7.2.3. Auxiliary data

These tables represent auxiliary data of test cases, please refer to D2.3.

7.2.3.1. Asset Transformer

Table Name: **asset transformer**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
bus_num	Number	Bus number
voltage_ratio	String	
insulation_medium	String	
type	String	
indoor_outdoor	String	
annual_max_load_kva	Number	
age	Number	
external_condition	String	
rating_kva	Number	
num_connected_customers	Number	
num_sensitive_customers	Number	
backup_supply	String	
cost_of_failure_euro	Number	
network_id	Number	Refers to Network table

TABLE 16 ASSET TRANSFORMER

7.2.3.2. Asset UG Cables

Table Name: **asset_ug_cable**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
section_label	String	Section
circuit_id	Number	
conductor_cross_sectional_area	Number	
sheath_material	String	
design_voltage	String	
operating_voltage	String	
insulation_type_sheath	String	
conductor_material	String	
age	Number	
fault_history	Number	
length_of_cable_section_meters	Number	

section_rating	Number	
type	String	
number_of_cores	Number	
net_performance_cost_of_failure_euro	String	
repair_time_hour	Number	
network_id	Number	Refers to a Network

TABLE 17 ASSET UG CABLES

7.2.3.3. Billing Consumption

Table Name: **billing_consumption**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_num	Number	Bus Number
type	String	Type of consumption
total_energy_consumption	Number	Total energy of consumption
unit_of_measure	String	Unit of measure
network_id	Number	Refers to a Network

TABLE 18 BILLING CONSUMPTION

7.2.3.4. Billing DER

Table Name: **billing_der**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
bus_num	Number	Bus Number
max_power_kw	Number	Max power
type	String	Type
network_id	Number	Refers to a Network

TABLE 19 BILLING DER

7.2.3.5. Generator Profile

Table Name: **gen_profile**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
season	String	A: Autumn S: Spring Su: Summer W: Winter
typical_day	String	Bd: business day Sa: Saturday Su: Sunday

mode	Number	Profile Mode: 1 = full time-series for a year 2 = a representative business day for a season 3 = a representative business day for a month 4 = a representative weekend for a season 5 = a representative weekend for a month
time_interval	Number	Time interval in minutes
upload_datetime	DateTime	Upload date time
network_id	Number	Refers to the network where the generator is connected to
input_file_id	Number	Refers to Input File

TABLE 20 GENERATOR PROFILES

7.2.3.6. Generator Electrical Values

Table Name: **gen_el_val**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
bus_num	Number	Refers to the network bus where the generator is connected to.
hour	Number	Hour
min	Number	Minutes
p	Number	Active power value
q	Number	Reactive power value
status	Number	Status of the generator
voltage_magnitude	Number	Voltage Magnitude value
gen_profile_id	Number	Refers to the generator profile
generator_id	Number	Refers to Generator
gen_id_on_subst	Number	Generator identification number on substation
nominal_voltage	String	Nominal voltage of substation

TABLE 21 GENERATOR ELECTRICAL VALUES

7.2.3.7. Load Profile

Table Name: **load_profile**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
season	String	A: Autumn

		S: Spring Su: Summer W: Winter
typical_day	String	Bd: business day Sa: Saturday Su: Sunday
mode	Number	Profile Mode: 1 = full time-series for a year 2 = a representative day for a year/season/month 3 = a representative business day 4 = a representative weekend
time_interval	Number	Time interval in minutes
upload_datetime	DateTime	Upload date
network_id	Number	Refers to the network where the load is connected to
input_file_id	Number	Refers to Input File

TABLE 22 LOAD PROFILE

7.2.3.8. Load Electrical Values

Table Name: **load_el_val**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
hour	Number	Hour
min	Number	Minutes
p	Number	Active power value
q	Number	Reactive power value
load_id_on_subst	Number	Id number of load on substation
nominal_voltage	String	Nominal voltage on substation
bus_id	Number	Refers to the network bus where the load is connected to.
load_profile_id	Number	Refers to load_profile

TABLE 23 LOAD ELECTRICAL VALUES

7.2.3.9. Transformer Profile

Table Name: **transf_profile**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
season	String	A: Autumn

		S: Spring Su: Summer W: Winter
typical_day	String	Bd: business day Sa: Saturday Su: Sunday
mode	Number	Profile Mode: 1 = full time-series for a year 2 = a representative business day for a season 3 = a representative business day for a month 4 = a representative weekend for a season 5 = a representative weekend for a month
time_interval	Number	Time interval in minutes
upload_date_time	Date	Upload date time
network_id	Number	Refers to a Network
input_file_id	Number	Refers to Input File

TABLE 24 TRANSFORMER PROFILE

7.2.3.10. Transformer Electrical Values

Table Name: **transf_el_val**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
hour	Number	Hour
min	Number	Minutes
tap_ratio	Number	Transformer Tap Ratio value
nominal_voltage		
status		Transformer status
Trasf_id_on_subst		
transf_profile_id	Number	Refers to a Transformer Profile
branch_id	Number	Refers to Branch

TABLE 25 TRANSFORMER ELECTRICAL VALUES

7.2.3.11. Branch Profile

Table Name: **branch_profile**

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
season	String	A: Autumn S: Spring Su: Summer W: Winter
typical_day	String	Bd: business day Sa: Saturday Su: Sunday
mode	Number	Profile Mode: 1 = full time-series for a year 2 = a representative business day for a season 3 = a representative business day for a month 4 = a representative weekend for a season 5 = a representative weekend for a month
time_interval	Number	Time interval in minutes
upload_date_time	Date	Upload date time
network_id	Number	Refers to a Network
input_file_id	Number	Refers to Input File

TABLE 26 BRANCH PROFILE

7.2.3.12. Branch Electrical values

Table Name: branch_el_val

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
hour	Number	Hour
min	Number	Minutes
p	Number	
q		
status		
branch_id_on_subst		
branch_profile_id	Number	Refers to a Transformer Profile
branch_id	Number	Refers to Branch

TABLE 27 BRANCH ELECTRICAL VALUES

7.2.3.13. Flexibility Profile

Table Name: **flex_profile**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
season	String	A: Autumn S: Spring Su: Summer W: Winter Empty
typical_day	String	Bd: business day Sa: Saturday Su: Sunday Empty
mode	Number	Profile Mode: 1 = full time-series for a year 2 = a representative business day for a season 3 = a representative business day for a month 4 = a representative weekend for a season 5 = a representative weekend for a month
time_interval	Number	Time interval in minutes
upload_datetime	DateTime	Upload date time
network_id	Number	Refers to the network where load is connected to

TABLE 28 FLEXIBILITY PROFILE

7.2.3.14. Flexibility Electrical Values

Table Name: **flex_el_val**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
bus_num	Number	Refers to the network's bus where the load is connected to.
hour	Number	Hour
min	Number	Minutes
pfmax_up	Number	maximum active power for upwards flexibility
pfmax_dn	Number	maximum active power for downwards flexibility
qfmax_up	Number	maximum reactive power for upwards flexibility

qfmax_dn	Number	maximum reactive power for downwards flexibility
flex_profile_id	Number	Refers to flex_profile

TABLE 29 FLEXIBILITY ELECTRICAL VALUES

7.2.3.15. Flexibility Cost

Table Name: **flex_cost**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
bus_num	Number	Bus Number
model	Number	Cost model: • 1=piecewise linear • 2=polynomial • 3= dynamic • 4= static
n_cost	Number	Number N=n+1 of data points defining an n-segment piecewise linear cost function
cost_pr	Number	Reserve active power cost
cost_qr	Number	Reserve reactive power cost
cost_pf	String	Parameters defining total cost function f(p), based on model
cost_qf	String	Parameters defining total cost function f(q), based on model
flex_profile_id	Number	Refers to Flex Profile

TABLE 30 FLEXIBILITY COST

7.2.3.16. Line Cable

Table Name: **line_cable**

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
fbus	Number	From Bus Number
tbus	Number	To Bus Number
length_km	Number	Length of the line (km)
type_of_installation	String	Installation type
network_id	Number	Refers to a Network

TABLE 31 LINE CABLE

7.2.3.17. DSO TSO Connection

This table persist nodes connecting TSO and DSO networks. Values are taken from auxiliary files 'Network_id_DSO_TSO connection.txt', included in the test cases:

'HR_DX_01_2020', 'HR_DX_02_2020', 'HR_DX_03_2020', 'HR_DX_05_2020'

Table Name: dso_tso_connection

NAME	*TYPE*	*DESCRIPTION*
id	Number	Table's primary key
tso_network_name	String	Reference to Transmission Network Name
dso_bus_num	Number	Reference to Distribution Bus Number
tso_bus_num	Number	Reference to Transmission Bus Number
dso_network_id	Number	Reference to Distribution Network ID

TABLE 32 DSO-TSO CONNECTION

7.2.3.18. Task

This table persists information related to the execution of the tools, such as running status, the user who launched the tool, etc.

Table Name: Task

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
task_status	String	Tool's running status. Possible values are: • PASSED • ONGOING • FAILED
info	String	Additional Information
date_time_start	DateTime	
date_time_end	DateTime	
tool_log_file_id	Number	Refers to Tool_Log_File
tool_id	Number	Refers to Tool
user_id	Number	Refers to jhi_user
simulation_id	Number	Refers to simulation

TABLE 33 TASK

7.2.3.19. Simulation

This table persists information about simulation scenario, the references to the network used and configuration file used for launching the tool.

Table Name: Simulation

NAME	TYPE	DESCRIPTION
id	Number	Table's primary key
uuid	UUID	Universally Unique Identifier
description	String	
Config_file	BLOB	Configuration file, containing all parameters needed for running a specific tool.
Config_file_content_type	String	Configuration file's content type. Usually: 'application/json'
network_id	Number	Refers to Network

TABLE 34 SIMULATION

7.3. ICT services

This section details the main services exposed by the ICT platform

7.3.1. Network data related services

7.3.1.1. Create Network:

POST /api/networks

Creates a new entry in the Networks table and returns the just-created network data

Input (Payload) Example (json format):

```
{
  "name" : "HR_Dx_04_2020",
  "mpcName" : "",
  "country" : "HR",
  "type" : "DX",
  "description" : "test ",
  "isDeleted" : false,
  "networkDate" : "2022-11-28T23:00:00Z",
  "version": 1,
}
```

7.3.1.2. Update Network

PUT /api/networks/:id

Updates an existing network, given its id

Path parameter:

Id: id of the network to be updated.

Returns the updated network data, in json format.

7.3.1.3. Networks' names list:

GET api/networks-names

Returns the list of all the networks' names

7.3.1.4. Networks' data

GET /api/networks/:id

Returns the network data, given the network's Id.

7.3.1.5. Buses list:

GET api/buses?networkId.equals=id
Returns the list of all the network's buses, given the network's id.

7.3.1.6. Branch list

GET api/branches?networkId.equals=id

Returns the list of all the network's branches, given the network's id.

7.3.1.7. Generator List

GET api/generators?networkId.equals=id

Returns the list of all the network's generators, given the network's id.

7.3.2. Exporters

7.3.2.1. ODS Network Export

GET api/ods/export/:id

`Id: id of the network to be exported`

Returns a file in '.ods' format, containing the network's representation according to the format used by the tool T41 (sheets: buses, lines, loads and gens)

7.3.2.2. MATPOWER Network Export

GET api/export-data/:networkName

`networkName: name of the network to be exported`

Returns a '.m' file, containing the network's representation in MATPOWER format

7.3.3. Importers

7.3.3.1. Import Network data from file:

POST /api/upload-network

Imports into the ICT database the network's data from a file.

`Request parameters:`

- `file: File to upload (allowed formats:'.m' and '.ods')`
- `networkName: Name of the network`

7.3.3.2. Import Profile form '.csv' file format for Distrubution networks

POST /api/csv/load-profile

`Request parameters:`

- `file: File to upload in '.csv' format`
- `networkId: network's id`
- `season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W' for Winter`

- typicalDay: possible values are: 'Bd' for a business day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

POST /api/csv/gen-profile

Request parameters:

- file: File to upload in '.csv' format
- networkId: network's id
- season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W' for Winter
- typicalDay: possible values are: 'Bd' for a business day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

7.3.3.3. Import Profiles from file excel for Distribution networks

POST/api/excel/load-profile

Request parameters:

- file: File to upload in '.xcsv' format
- networkId: network's id
- season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W' for Winter
- typicalDay: possible values are: 'Bd' for a business day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

POST/api/excel/gen-profile

Request parameters:

- file: File to upload in '.xcsv' format
- networkId: network's id

- season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W for Winter
- typicalDay: possible values are: 'Bd' for a business Day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

POST/api/excel/flex-profile

Request parameters:

- file: File to upload in '.xcsv' format
- networkId: network's id
- season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W' for Winter
- typicalDay: possible values are: 'Bd' for a business day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

Each file (.xlsx or .csv) used for import profile data, must respect the following convention:

- The first column refers to the node where the load/generator is connected
- The second column indicates if the values correspond to active (P) or reactive power (Q)
- The next columns include the power values for each period of 15 minutes or 1 hour of that specific day.

7.3.3.4. Import all profile Transmission network

POST/api/excel/all-profile

Request parameters:

- file: File to upload in '.xlsx' format
- networkId: network's id
- season: possible values are: 'A' for Autumn; 'S' for Spring; 'Su' for Summer; 'W' for Winter
- typicalDay: possible values are: 'Bd' for a business day; 'Sa' for Saturday; 'Su' for Sunday
- mode possible values are:
    - 1 = full time-series for a year
    - 2 = a representative business day for a season
    - 3 = a representative business day for a month
    - 4 = a representative weekend for a season
    - 5 = a representative weekend for a month

The sheets loaded from the file are: 'Load P (MW)', 'Load Q (Mvar)', 'Gen status', 'Gen P (MW)', 'Gen Vg (p.u)', 'Transformer tap ratio', 'Transformer status', 'Branch status'.

7.3.4. Visualization

7.3.4.1. Single Line Diagram network view

GET /api/v1/sld-network/id/:id

`Id: network's id`

Returns a json with the SVG and metadata used to display the single-line diagram for the whole network.

7.3.4.2. Single line Diagram substation view

GET /api/sld-substations/id/:id

`Id: network's id`

Returns a json with the SVG and metadata for each substation, used to display their single-line diagrams.

7.3.4.3. Chart representation of Load/Generator profile

GET /api/chart/load-seasons/:networkId/:busNum

`Path parameters:`

- `networkId: network's id`
- `busNum: bus num where load or generator is connected`

Returns data, in json format, containing all the information needed to plot a chart with power values of a specific day, grouped by season.

7.3.5. Tools

A set of services has been developed for running, displaying, and downloading the results of every tool integrated in the ICT platform. Below are listed examples of services related to a generic tool, to describe the approach; the detailed services for the tools in the open-source toolbox will be listed in D6.2: Integration.

7.3.5.1. Run tools:

POST /api/tools/wp<WorkPackageNum>/run

The parameters to run the service depend on the tool

7.3.5.2. Display Results:

GET /api/tools/wp<WorkPackageNum>/show-charts

Returns data in a json file, used to show a chart representation of the results generated by a tool.

The request parameters used by each service, depend on the needs of the specific tool.

7.3.5.3. Download tools execution log:

GET api/tool-log-files/taskId/:id

Parameter:

Id: unique task identifier

Returns the 'log.txt' file generated by the tool, for a given task id.

7.3.5.4. Download tools results:

GET /api/tasks/tool-results/:id

Parameter:

Id: unique task identifier

Returns a .zip archive containing all files generated by the tool, for a given task id

7.3.5.5. Tasks list:

GET /api/tasks

Returns a list of tasks. Each task is generated every time a tool is launched and returns information about the status of the tool.