European Research Infrastructure supporting Smart Grid Systems Technology Development, Validation and Roll Out

Work Package 05

NA5 - Holistic System Integration and Testing Procedure

Deliverable D5.2

D-NA5.2 Partner profiles

Grant Agreement No:	**654113**
Funding Instrument:	**Research and Innovation Actions (RIA) – Integrating Activity (IA)**
Funded under:	**INFRAIA-1-2014/2015: Integrating and opening existing national and regional research infrastructures of European interest**
Starting date of project:	**01.11.2015**
Project Duration:	**54 months**

Contractual delivery date:	**30/04/2017**
Actual delivery date:	**04/05/2018**
Name of lead beneficiary for this deliverable:	**18 VTT Technical Research Centre of Finland**
Deliverable Type:	**Other (O)**
Security Class:	**Public (PU)**
Revision / Status:	**released**

Project co-funded by the European Commission within the H2020 Programme (2014-2020)

Document Information

Document Version: 7

Revision / Status: released

All Authors/Partners

Anna Kulmala, Kari Mäki, Erkka Rinne / VTT
Oliver Gehrke, Kai Heussen, Esteban Bondy / DTU
Maurizio Verga, Carlo Sandroni, Daniele Pala / RSE
Van Hoa Nguyen, Yvon Besanger / GINP
Marita Blank, Martin Buescher / OFF
Mislav Findrik, Paul Smith / AIT
Alexandros Rigas / ICCS
Ata Khavari, Mihai Calin, Maria Sosnina / DERlab
Evangelos Rikos / CRES
Rishabh Bhandia / TUD
Ibrahim Abdulhadi / UST
Quoc Tuan Tran / CEA

Distribution List ERIGrid consortium members

Document History

Revision	Content / Changes	Resp. Partner	Date
1	First version of the data definition document	VTT	30.09.16
2	Second version of the data definition document	VTT, DTU, ICCS, GINP, OFF, UST, OFF, DERlab	02.01.17
3	Third version of the data definition document	VTT, DTU, RSE, GINP, OFF, AIT, ICCS, DERlab, CRES, TUD, UST, CEA	21.02.17
4	First deliverable draft, online working document	VTT	20.30.17
5	Internal review version	VTT, DTU, RSE, GINP, OFF, AIT, ICCS, DERlab	13.04.17
6	Complete version	VTT	05.05.17
7	Revised version based on feedback and recommendations from the mid-term project review	VTT, DERlab	06.04.18

Document Approval

Final Approval	Name	Resp. Partner	Date
Review Task Level	Anna Kulmala Oliver Gehrke, Kai Heussen	VTT DTU	14.04.17
Review WP Level	Erik de Jong Ron Brandl	DNVGL IWES	23.04.17
Review Steering Com. Level	Thomas Strasser	AIT	19.05.17
Review Steering Com. Level	Thomas Strasser	AIT	04.05.18

Disclaimer

Copyright Notice

Table of contents

Abbreviations

CHIL	Controller Hardware-In-the-Loop
CIM	Common Information Model
CPU	Central Processing Unit
DB	Database
DER	Distributed Energy Resource
DoA	Description of Action
EMS	Energy Management System
E-SC	Experiment Specific System Configuration
GSC	Generic System Configuration
GSM	Global System for Mobile Communications
HDD	Hard Disk Drive
HIL	Hardware-In-the-Loop
HMI	Human Machine Interface
ICT	Information and Communications Technology
IDS	Intrusion Detection System
IED	Intelligent Electronic Device
IP	Internet Protocol
IT	Information Technology
LTE	Long Term Evolution
MPP	Maximum Power Point
OPC	Open Platform Communications
OuI	Object under Investigation
PC	Personal Computer
PHIL	Power Hardware-In-the-Loop
P&ID	Piping and Instrumentation Diagram
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
PV	Photovoltaic
RAM	Random Access Memory
RDF	Resource Description Framework
RI	Research Infrastructure
RI-GSC	Research Infrastructure Generic System Configuration
RI-SC	Research Infrastructure System Configuration
RMS	Root Mean Square
RTU	Remote Terminal Unit
SAN	Storage Area Network
SC	System Configuration
SCADA	Supervisory Control And Data Acquisition

SGAM	Smart Grid Architecture Model
SQL	Structured Query Language
SSL	Secure Sockets Layer
SuT	System under Test
TA	Trans-national Access
TC	Test Case
TC-GSC	Test Case Generic System Configuration
TLS	Transport Layer Security
TS-SC	Test Specific System Configuration
UC	Use Case
UC-GSC	Use Case Generic System Configuration
UG	User Group

Executive Summary

Testing advanced smart grid solutions requires a combination of different Research Infrastructure (RI) capabilities depending on the functionality to be tested as well as the objectives of the experiment. Harmonized information on different RIs is needed to be able to utilize the available infrastructures as efficiently as possible. Currently, however, consistent data on RIs is not available but each RI uses its own description. The level of detail in the associated documentation can vary significantly and often some important information is missing from the written documentation and is only available in the memory of the internal users of the specific RI. Harmonized RI data is mandatory to facilitate the holistic testing procedure that is currently being developed within the ERIGrid project.

This report outlines the ERIGrid approach for RI profiling. The RI profiles are stored in the ERIGrid RI profiles database that has been designed based on identified cases for using the database and is aligned with conventions and information models commonly used in ERIGrid. The data structure builds on previous work in ERIGrid where a method for the description of holistic testing scenarios was determined [1]. The RI profiles database defined in this deliverable implements the two system configuration types related to RIs (Research Infrastructure Generic System Configuration – RI-GSC and Research Infrastructure Specific System Configuration RI-SC) which were not discussed in detail in [1]. These system configurations are a mandatory part of the ERIGrid holistic testing procedure and will serve as an input for future ERIGrid work enabling development of mapping procedures to select the most suitable RIs for specific test cases. The already existing DERlab database has been used as a basis for the work.

Three cases for using the ERIGrid RI profiles database have been identified. These cases are:

- The application of the holistic testing procedure developed by ERIGrid
- The preparation and execution of experiments conducted by external users as part of the TA programme
- The provision of better and more detailed laboratory information to interested parties, through the ERIGrid website

Also new, yet unknown, use cases may emerge when the database is published to a wide audience.

This report introduces a general data model for RI profiling (RI-GSC) and discusses the database implementation of ERIGrid (RI-SC). The general data model consists of the data structure definition and detailed data attributes for each database object. A database instance is implemented based on these definitions and data on ERIGrid partner RIs is entered into this database according to the modelling principles introduced in this report. In ERIGrid, it is not required from all RIs to use the most detailed modelling possibilities of the database and minimum requirements for the data to be entered/provided are given. Component information extracted from the database instance filled with ERIGrid partner data is publicly available at the ERIGrid website. The whole database instance can be made available to TA users upon request.

1 Introduction

Testing different smart grid use cases and scenarios requires a combination of different Research Infrastructure (RI) capabilities. It is therefore judicious to improve collaboration among research and industrial institutions, to efficiently exploit the existing platforms and to complement the missing infrastructure with available assets from other partners. Consequently, it is necessary to have interoperability among the various platforms and harmonized information on the RIs to be able to find the most suitable RIs for specific tests. Nowadays, however, each RI uses its own documentation and consistent data is not yet available.

This deliverable aims at defining an RI description method that can be used as a part of the ERIGrid holistic testing procedure introduced in [1]. Furthermore, the RI profiles can be used to simplify the RI access process for new internal users or external users through for example Transnational Access (TA). In addition to defining the description method, the deliverable also provides profiles of all ERIGrid RIs modelled using the defined description method. The RI profiles are stored in a database (DB)and component information extracted from the DB is publicly available at the ERIGrid website [2] .

1.1 Purpose of the Document

The purpose of this document is to present the ERIGrid approach for RI profiling. A database format was selected to enable possible automated steps in the holistic testing procedure [1]. Furthermore, it enables querying the relevant information for each particular test case and avoiding the need to read long documents in the initial RI selection phase. If, for instance, a test requires a photovoltaic (PV) panel, the RIs that do not have one can be automatically excluded before more comprehensive assessment of RI capabilities is conducted.

This document describes the structure of the database and attributes for each database object. This definition is generic and can be used also for different uses of the database than the ones identified in this deliverable. The deliverable gives also general instructions on how to model RIs in the database and describes the database instance implemented in ERIGrid.

1.2 Scope of the Document

The RI profiles of this deliverable represent static data of the RIs. This means information that does not change when the test setup is changed but needs to be updated only rarely when new equipment is purchased or the static RI configuration is changed. The level of details in the database has been defined as a trade-off between completeness and search-ability of information, and the complexity and effort required to enter the data into the DB. The ERIGrid project includes also other activities that consider the documentation of non-static RI information [3] and real-time data transfer between RIs [4]. This type of data is out of the scope for this deliverable but the naming conventions etc. will be aligned in these activities to have a consistent way of describing the ERIGrid RIs and enable reusing static data of this deliverable in pre- and post-experiment data processing and real-time data transfer between RIs.

The ERIGrid RI profiles database is used through a simple web user interface that does not have advanced functionalities. In many cases, the database users need to use Structured Query Language (SQL) to enter or query data and, therefore, should have basic knowledge of SQL coding. The simplest operations (e.g., adding a component) can be performed also through the simple graphical user interface without a need to compose SQL queries. A more advanced graphical user interface with a multitude of predefined queries can be implemented at a later stage but is not included in the ERIGrid project scope. Basic knowledge of the Common Information Model (CIM) may help in understanding the DB structure but is not a mandatory requirement for the database users.

Since using the database through SQL queries might be arduous for some users, it was decided

that the whole database as such is not published but the data is prepared in a way which is more easily understandable for domain experts. Component information extracted from the DB is presented in a table format at ERIGrid website. Also, the whole database can be made available to TA users upon request.
The modelling instructions given in this deliverable define general requirements on how different parts of the RIs should be modelled in the DB. This document does not include a step-by-step tutorial on how to use the web user interface utilized in ERIGrid and should not be seen as a manual or troubleshooting reference, but rather as requirements definition document.

1.3 Structure of the Document

The cases for using the database are discussed in Section 2. These use cases set the requirements for the information that has to be available from the database. The general database structure is defined in Section 3. In addition to the data structure definition, this section introduces also the conventions and information models relevant for the database definition. Section 4 introduces the database instance implemented as a part of this deliverable and discusses the minimum requirements for the data to be entered into the database. Section 5 consists of examples that explain the modelling principles in more detail. In addition, some example queries to extract data from the database are given. Section 6 includes the public part of the components and attributes information extracted from the database for each laboratory. This information is presented in ERIGrid website. Detailed data attributes of each database object are presented in the Annex.

In addition to this document, further material related to the database will be available on the ERIGrid project website [5]. This material will include a SchemaSpy [6] description of the DB.

2 Cases for Using the Database

The ERIGrid workplan (i.e., Description of Action – DoA) provides a high-level view on the partner profiling activities. It mentions the goal of RIs becoming "comparable and specifically identifiable" as well as the proto-use case of "finding the most appropriate partner for a specific task". It does not, however, specify concrete use cases. In order to collect the right type and amount of information and to structure the information in a convenient and efficient manner, it is necessary to define the intended use of the partner profiles and the database they are stored in. Specifically, it needs to be known *who* will be using the database, and *in which way*.

A collection of detailed information about a large number of European RIs is a valuable asset, and as is often the case with data (especially machine-readable data), yet unknown use cases may emerge just because the data is available. It is therefore impossible to compile an exhaustive list of potential users and use cases at this point. However, some cases are known, understood and described (or at least mentioned) in the ERIGrid DoA or have emerged during the work described in this deliverable. These cases are:

- The application of the holistic testing procedure developed by ERIGrid,
- The preparation and execution of experiments conducted by external users as part of the TA programme, and
- The provision of better and easy understanding detailed laboratory information (important component and their attributes) to interested parties who could be the TA potential applicants, through ERIGrid website.

These cases will be described in more detail in the following subsections.

2.1 Holistic Testing Procedure

ERIGrid's approach on holistic testing may be considered as a vision of a standardized process and methodology realizing the testing of systems to address both system complexity and multi-domain testing challenges. The extensions of conventional testing employs formalizations of test objectives and process, as well as enhanced means of testing, allowing increased benefits from modern system testing technologies such as co-simulation, power system and Information and Communication Technology (ICT), real-time simulation and Power Hardware-in-the-Loop (PHIL). Consequently, this vision extends to the joint use of resources from multiple research infrastructures to allow further integration and effective use of testing resources by interoperation and mutual access to a shared experimental platform: to conduct *a)* parallel, consecutive, cascaded experiments, as well as *b)* online integrated tests using consortium-wide specification of interoperable RI profiles.

The main steps of the envisioned holistic testing procedure are outlined in Figure 2.1. Implementing a harmonized and holistic testing procedure applicable for all ERIGrid partners and RIs is a long-term task requiring a strong interoperability. This implies however multiple harmonization efforts and eventually changes of actually employed standards and protocols in the participating laboratories.

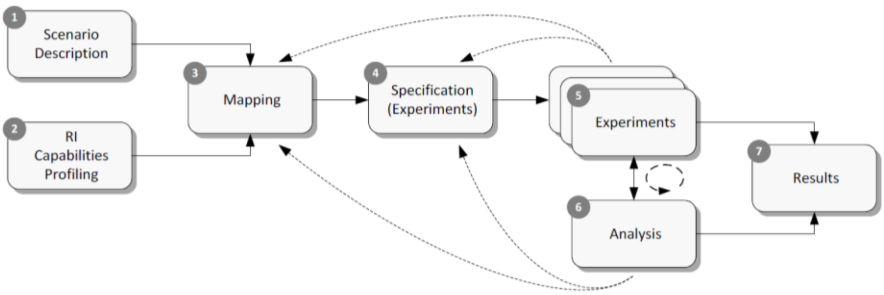


Figure 2.1: Outline of holistic validation procedure from ERIGrid DoA

Several concrete challenges can be pointed out:

- Difference in actual infrastructures (communication protocols, standards, platform, etc.), and resulting issues with transferability of tests
- Incomplete RI profiling due to lacking information about functional aspects of the RI, as typically only physical resources are listed
- Difficulty in adaptation of tests to a partner infrastructure due to the gap of the RI to the defined test requirements
- Ineffective exchange of test results and analysis because of lack of test annotations or agreed exchange formats
- The potential for conflicting interpretation of the terminologies in test specification and evaluation due to 'shoptalk': the local RI way of describing things drifts with differences in e.g., scales of systems, the lab component types and primary domains of inquiry (e.g., electrical, ICT, heat), etc.
- Software, model or licensing issue causes difficulty of implementing co-simulation or concurrent tests even if 'on paper' compatibility is available
- Rules and access procedures across research infrastructures can follow different principles

The partner profiling and corresponding database aims to help resolving many of the aforementioned issues in several ways:

i. By developing a common format for representing all of the partner RIs it will be possible to create a common information model which will facilitate information exchange between RIs, for example when exchanging experimental results and analysis
ii. The RI description database will facilitate testing resource management and coordination among partner infrastructures
iii. Enable the application of systematic mapping of test systems to available RI infrastructure as outlined in the holistic testing procedure [1]
iv. Improve interoperability between infrastructures and facilitate post-experiment data processing

The main description elements required for a holistic testing procedure have been outlined in the ERIGrid deliverable [1] as summarized in Figure 2.2. The RI profiles database here holds the information about what system configurations are possible across the partners.

Several types of system configurations have been defined in [1], as summarized in Table 2.1. The generic RI information model (RI-GSC) is developed in this report, and the RI database is populated with RI profile information (RI-SC) in Step ②.

Following the sequence of the holistic test procedure, the information contained in generic system configurations is used in a preliminary feasibility assessment of test cases in Step ①; the RI database can at this point be queried for the type of equipment available across the partners' RIs and match results against test objectives and generic system configuration elements (starting from use case generic system configuration UC-GSC, leading to a more test-oriented test case generic system configuration TC-GSC). Once a holistic test case is specified, it can be divided into a number of sub-tests (Step ③), each with specific test criteria and different test system configurations (test specific system configuration TS-SC: a specific grid configuration, related loads and generation, control system topology, etc.); the RI database is used here to assess feasible test system configurations (e.g., how large a test system can be emulated in the available real-time simulators? Is a communication simulator available in the same infrastructure as a real-time power system simulator?). Eventually, in Step ④, a specific test system configuration is mapped to a specific research infrastructure, resulting in an experiment setup (experiment specific system configuration E-SC); a first draft of this setup can be based directly on the RI profile information from the RI database, and could thus ideally be performed by partners external to the respective RI. The remaining gap between first draft and final experiment setup can then be closed in dialogue with the hosting RIs.

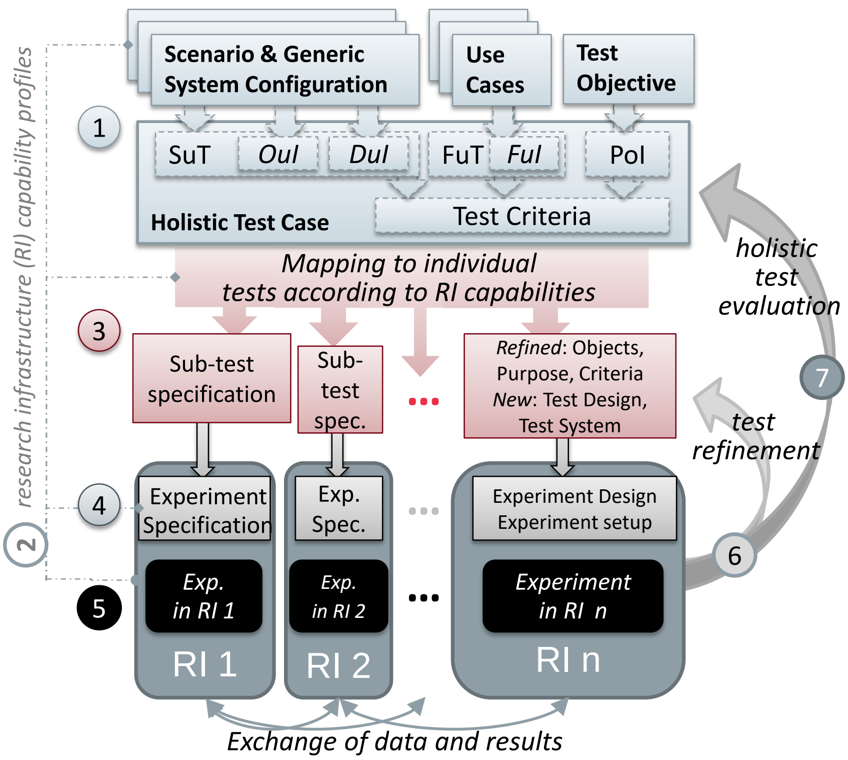


Figure 2.2: Main steps and specifications of the ERIGrid methodology [1]. The RI profiles represented in the RI database (step ②) are relevant in several steps (①, ③, ④ and ⑤), as the dashed grey line indicates. The core application is Step ④, where the RI profiles data is mapped to a specific experiment configuration.

Table 2.1: Classification of System Configuration Types (SCTypes) [1]. GSC refers to a generic system configuration, and (S)SC to a specific system configuration. While the GSCs define a contextual information model, a (S)SC defines a specific instance of a system configuration.

Name/ Purpose	Context / Document	GSC/ (S)SC	SCType	Explanation
Function-System Alignment	Use Case	GSC	UC-GSC	As SGAM domains & zones: reference designation for functions, independent of test case. Corresponds to generic system configurations first defined in [7].
Test Case context model	Test Case	GSC	TC-GSC	Establishes type conventions for test case: relevant SC component types, domains, etc., and categorically identifies the SuT (and optional OuIs); specifies multiplicities; “class model”.
Test System	Test Specification	(S)SC	TS-SC	A concrete instance of TC-GSC to address a specific OuI and test criteria; labelled terminals and specific connections; OuI and SuT identified as overlay annotation.
Experiment Setup	Experiment Specification	(S)SC	E-SC	The configuration and interconnection of RI components, representing the SuT, and including OuI; also “Test Setup”
RI Description	RI database entry	(S)SC	RI-SC	Lab configuration with components, including potential multiplicity and potential connectivity of lab components, but may have undefined connectivity.
RI information model	RI profiling	GSC	RI-GSC	Specification of Lab profiling data structures, including component types and domain types.

Eventually, this process may support further automation of elements of this mapping as well as the interoperability of RIs for integrated experiments.

2.2 Information for External Users

In addition to the holistic testing procedure introduced in the previous section, the DB can also be utilized to provide more detailed information on the laboratories to the external user groups. In the following, two examples are presented.

ERIGrid offers external users transnational access to the research infrastructures of its consortium partners along with logistical, technological and scientific support. Industrial and academic researchers active in the field of smart grids are targeted as prospective participants for the TA. This access is funded by the H2020 RI Initiative and as such is offered *free of charge* to researchers that want to carry out research projects characterized by a high level of excellence and innovation that are in line with the ERIGrid and EU objectives. The goal of the TA programme is to support and accelerate the innovation and development of new smart grid solutions and products, and support the development of an integrated European RI.

The TA User Groups (UG) preparing the proposal should indicate the preferred infrastructures (up to three options) where the experimental research will be carried out [8]. In addition to the description of ERIGrid RIs which are provided in [8] and at the ERIGrid website [2], the component information extracted from the ERIGrid RI profiles database will also support the UGs to find the most suitable RI(s) for carrying out their proposed research project and provide more consistent proposals in terms of feasibility during the preparation phase. Accepted UGs can also obtain access to the whole ERIGrid RI profiles database which they can utilize in the experiment execution phase considering the provided technical details on the capabilities and features of the RIs.

The public version of the ERIGrid RI profiles database is integrated into the ERIGrid website. Therefore, beside to the descriptions of each ERIGrid laboratory for the TA use, which have been presented on the website, an overview of the RIs and their facilities including the main components and their attributes will be accessible for the public. The resulting database can be used by researchers who are interested to apply for transnational access provided by ERIGrid RIs as well as researchers who seek for a suitable RI in order to carry out specific experiments. It would be possible to integrate the public version of the ERIGrid RI profiles database also into the DERlab database but is out of scope for ERIGrid.

2.3 Requirements Summary

The two previous subsections have defined user groups, the motivations these user groups may have to access the database, and the processes that create these motivations. Beyond this level of detail, it is difficult or impossible to compile an exhaustive list of potential database queries, because it is likewise difficult or impossible to compile an exhaustive list of potential laboratory experiments.

The following list provides a small selection of concrete usage examples for each of the cases introduced in Sections 2.1 and 2.2 based on either experienced needs in previous projects (e.g., EU FP7 DERri) or future testing to be performed during the ERIGrid project (e.g., as defined in the single-RI and multi-RI test cases compiled in ERIGrid deliverable [9]) and beyond.

Case 1: Holistic Testing

- Retrieve a list of available components of a specific type (e.g., PV panels) and their association with laboratory locations to assess the availability of relevant infrastructure for a given testing need
- Retrieve a list of several types of components (also characterized by component parameters, such a maximum power), grouped with respect to RI to identify suitable RIs for a given testing need
- Extract the available and configurable electrical topology to assess whether a specific test system grid configuration can be emulated by the lab grid structure

- Explore the control hierarchy of a particular RI, in order to determine how to represent e.g., a distributed voltage controller during the mapping stage
- Retrieve a list of potentially relevant cybersecurity equipment to be considered when designing a multi-RI experiment
- Generate load flow simulation models of individual RIs from electrical grid data, in order to determine and compare expected voltage rises for power injection at the longest possible feeder

Case 2: Trans-national Access

- Learn about the topology and possible configurations of the electrical network at a particular RI
- Retrieve nameplate data for all DER units involved in a planned experiment
- Learn which options exist for controlling a particular DER unit at a particular RI, and how to interface to this DER
- Identify the RIs which have physical storage units and PV inverters installed
- Learn about the capabilities and possible operating modes of a particular unit at a partner RI

3 General Database Characteristics

The database has been designed based on the usage requirements defined in the previous section and taking into account conventions and information models commonly used in ERIGrid. It should be noted that the data definition consisting of the data structure (Section 3.2) and detailed data attributes (Annex 9.3) can be subject to change in the future due to either currently unforeseen cases for database use or changes in technology that require adding new information to the database. Hence, extensions to the database definition can occur during later stages of the ERIGrid project.

This section will first introduce the system configuration concept defined in [1] and the common representation of control infrastructure defined in [3]. Thereafter, the database structure is introduced. The attributes of database objects are defined in Annex 9.3.

3.1 Conventions and Information Models

3.1.1 System Configuration Description

The system configuration description method offers a standard way of representing systems. System configurations provide information on domains, components, connectivity, constraints and attributes in a system. RI profiling information consists exactly on these types of information and, therefore, the system configuration description method is used as a basis for defining the database characteristics. In [1], the system configuration information can be provided in a table form and/or using a standardized graphical SC representation. The DB, naturally, utilizes the table form of a SC but graphical representations can be stored as well.

The class diagram presented in Figure 3.1 identifies the fundamental modelling concepts (upper ontology) used in the system configuration description method. The method is applicable in all domains (electrical, control, heat etc.). A CIM based connectivity concept has been adopted where each component has *terminals* (0 or more) and the topology is modelled through connecting the terminals by *connectivity nodes* [1]. This principle enables more than pairwise connectivity, and also allows unconnected terminals to be defined. The CIM connectivity concept is however extended to allow connections and terminals in any *domain.* A domain can thus be for example ICT networks, which can be modelled with the same approach as the electricity networks. All terminals connected to a particular connectivity node have to be associated with the same domain and a terminal can be connected to only one connectivity node. Domains can be organized hierarchically to represent different levels of precision (e.g., the 33kV domain is a *subdomain* of the electrical power domain). This data structure has been used as a starting point for the database definition of this deliverable.

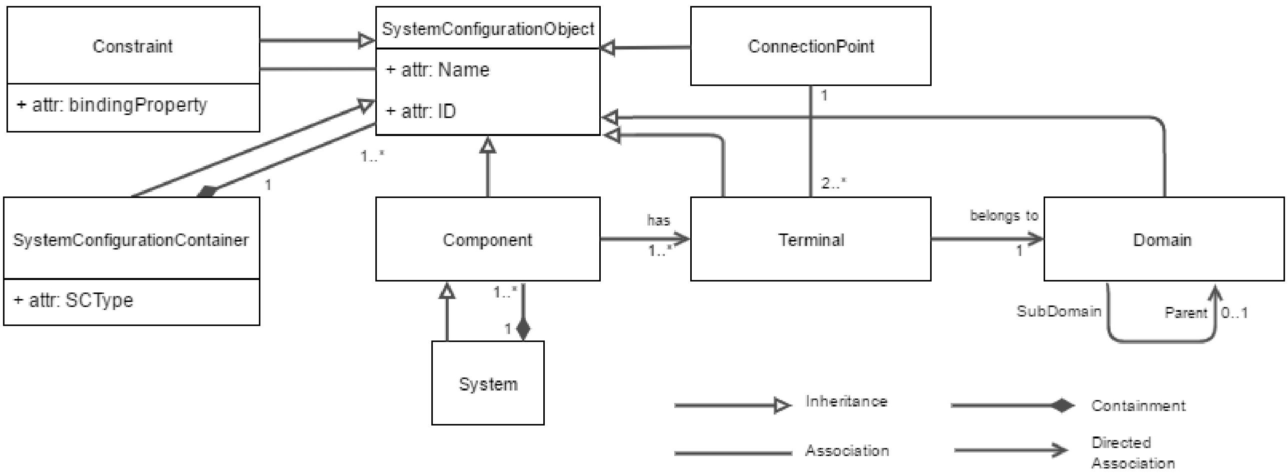


Figure 3.1: Data structure class diagram from [1]

System configurations can be applied in several contexts and can range from high-level generic system configurations to very detailed experiment specific descriptions. The different types of system configurations defined in [1] have been introduced in Table 2.1. The first four (UC-GSC, TC-GSC, TS-SC, E-SC) are related to different stages of the holistic testing scenario description consisting as outlined above. This deliverable concentrate on the last two SC types (RI-SC, RI-GSC) which are used to model the RI capabilities. RI-GSC defines the information model and implied data structure that is used to model the RIs, and is the same for all database RI entries. RI-GSC does not include any information on particular laboratories but just provides the generic data structure for modelling any RI. ERIGrid RI-GSC is defined in this deliverable in Sections 3.2 and Annex 9.3. RI-SC is an instance of the RI-GSC where data on modelled RIs is entered. Section 4 discusses ERIGrid specific instructions for recording RI data and the filled RI profiles database is the ERIGrid RI-SC.

3.1.2 Common Representation of Control Infrastructure

Descriptions of laboratory capabilities often focus on the physical characteristics of the setup and its components, such as the design, physical placement and operating limits of equipment, or the topology of e.g., the electrical power grid or a communication network. A detailed description of the control capabilities of the infrastructure is often omitted, as well as the possibilities for deploying custom control solutions at the infrastructure. However, in the context of smart grid research, a description of these control capabilities is essential for understanding which types of experiments can be conducted at a particular research infrastructure.

The participating RIs are quite diverse in their purpose, design, capability and degree of automation. In order for a laboratory database to help understand and compare laboratory capabilities, a common representation for control and automation capabilities of individual RIs has to be found. Such a representation has been developed as the "generic model for control infrastructure" in ERIGrid and will be fully documented in [3]. Those parts of the model which are relevant to the database are summarized below.

The central concept of the control model is the definition of five distinct "control levels" and an enumeration of the communication interfaces by which controllers on different levels interact. Each level is further subdivided into "internal" and "external" control, where internal controllers are defined as entities which are considered part of the laboratory installation, whereas external controllers may be 3rd party equipment brought to the laboratory (e.g., by a test client) or a controller in a different location which interacts with the laboratory through a remote interface. In all cases, the term "controller" covers control software as well as dedicated software-hardware combinations.

The availability of communication interfaces between the different components and parts of a laboratory determines whether the laboratory presents itself to the user as a collection of hardware components or as an integrated system. It also determines the degree of flexibility in the installation in terms of which types of external components can be part of the system under test.

Figure 3.2 provides an overview of the generic model. The five control levels are marked C1...C5. Internal controllers have been assigned the levels D1...D5, corresponding to their associated control levels, while external controllers are denoted X1...X5. Communication between the local controllers, i.e., communication links which are considered to be part of the laboratory infrastructure, are enumerated L1...L4, while those communication links which enable the interaction between the laboratory and external controllers, have been assigned the identifiers E1...E12. These are being discussed in detail in [3], including selected examples and use cases.

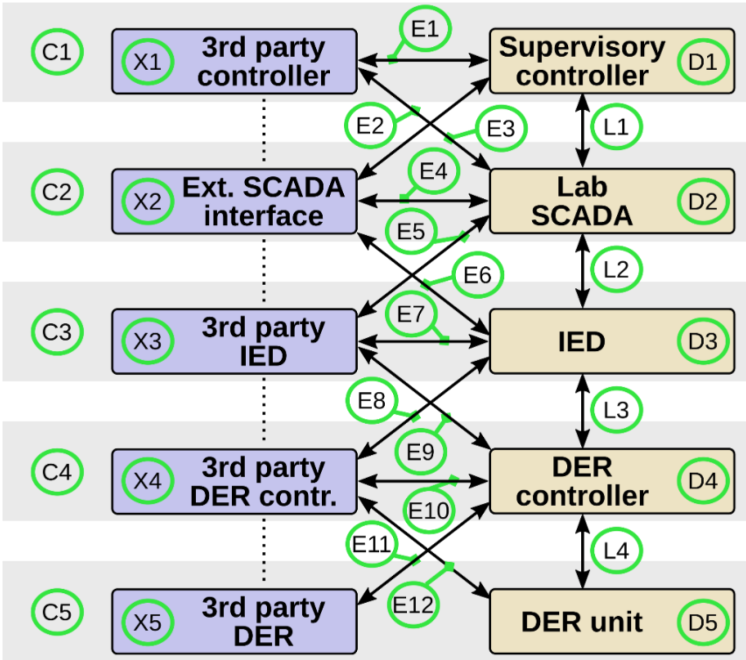


Figure 3.2: Control levels defined in [3]

The five control levels are defined as follows:

- **Level D5: DER units** and other electrical power hardware, including generation, consumption and storage units such as PV inverters, batteries or loads as well as other types of remote accessible power equipment such as e.g., motorized circuit breakers, variable transformers or measurement equipment. DER components are typically fitted with sensors, contacts and actuators as their primary interface for monitoring and influencing the operation of the device. DER units may contain simple control and protection elements behind the sensor/actuator interface, such as e.g., the speed governor of a genset, the thermostat of an electrical space heater or the mechanical emergency brakes of a wind turbine.
- **Level D4:** The majority of DER units, when commercially produced, are being delivered with a **DER controller** as an integral part of the device. DER controllers are typically microcontrollers, embedded computers or PLCs (programmable logic controllers) with embedded firmware. The DER controller contains the operational logic, which coordinates the interaction between different subsystems in the unit. This logic may contain control loops such as e.g., the maximum power point (MPP) tracker of a PV inverter, as well as stateful logic, such as e.g., start and stop sequences for a flow battery system. One of its most important tasks is the enforcement of operational limits in order to ensure device integrity and user safety, and to limit wear.
- **Level D3:** In electrical power systems, the term **Intelligent Electronic Device (IED)** is used to describe a microprocessor-based device, which is deployed in the field, typically in close proximity to an energy resource controlled by the IED. Besides providing control functionality such as e.g., protective tripping, breaker interlocking or voltage regulation, IEDs may serve as a gateway to a SCADA (Supervisory Control and Data Acquisition) system, translating between a grid-wide communication standard used by the SCADA system and specific protocols understood by energy resources.

  The requirement for more and more complex field-deployed systems, driven by the desire for more advanced smart grid functions, requires a broad-ranging definition of what an IED is. The generic model uses three criteria: A local controller (or a local, microprocessor-based unit onto which control software can be deployed) which is not considered (by the manufacturer or the user) to be part of a DER installation, and/or which may serve as a gateway for communication with one or multiple DER units.

- **Level D2:** A **Lab SCADA** system integrates a collection of laboratory components into a system, which can be accessed through a unified interface, for the purpose of remote control and monitoring. Typical functions associated with a lab SCADA system are data transfer, data validation, data logging and the access to historical data, user authentication and access control, visualization and the enforcement of set point limits.

  Supervisory controllers (e.g., Energy Management System (EMS) applications) and data acquisition equipment (e.g., Phasor Measurement Units (PMUs) or IEDs) are not considered to be part of the SCADA system; rather, the purpose of the SCADA system is to allow data exchange between these subsystems.

- **Level D1: Supervisory controllers** perform control at the system level, where the “system” in question may refer to the entire laboratory, or an ad-hoc collection of resources that belong to an experiment. Examples for supervisory controllers in a laboratory context could be:
  - A piece of software, custom built for the purpose of conducting an aggregation experiment, which implements the functionality of an aggregator and is capable of coordinating the behaviour of several DER units through the lab SCADA system
  - A piece of software, custom built for the purpose of performing a long-term repeat experiment in an unattended fashion, by e.g., remote-controlling DER units, switching circuit breakers, or starting and stopping measurement devices and data loggers
  - A software function, considered to be part of the permanent laboratory infrastructure, which restores the entire laboratory to a predefined default state.

Supervisory controllers in a laboratory context may provide set points to an IED or directly to a DER controller. Other arrangements are possible; for example, a supervisory controller may be implemented as a distributed system, which executes on several capable IEDs.

3.2 Data Structure

This section discusses the general concepts behind the data structure adopted for the database.

The primary thought governing the choices made has been the desire to find a good compromise between effort and usefulness, specifically: Providing a high enough level of detail to make a real difference to data users (e.g., compared to the existing laboratory descriptions) while not creating excessive work for the individual RIs which have to render their laboratories into the chosen format. It has been a matter of concern that, if the bar for participation is set too high, the database would end up containing incomplete or inaccurate information. At the same time, it should be possible for individual RIs to provide detailed - and possibly (partly) machine-readable – models of their infrastructure.

In order to solve this issue, the data structure has been designed to contain mandatory and optional elements. The mandatory data is generally of lower complexity than the optional data. This principle has been applied to two main areas: Infrastructure representation and object detailing.

The representation of infrastructure, i.e., the composition of the electrical, thermal and communication networks as well as the hierarchy of control, borrows from the high-level class model (components, terminals, connectivity nodes) in order to be aligned with the approach taken for the description of system configurations outlined in [1] (as summarized above).

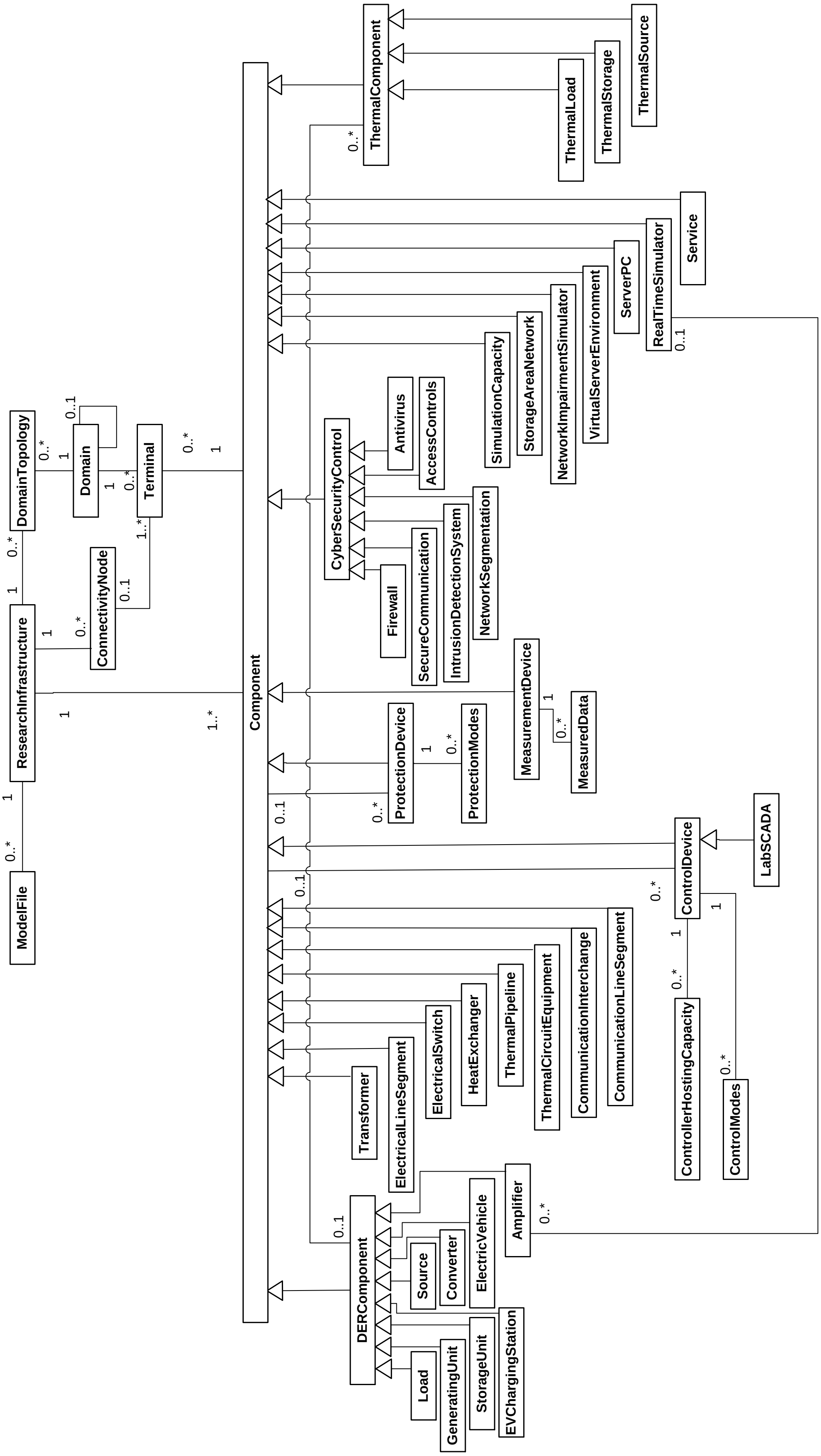


Figure 3.3: The logical table structure of the RI DB

The logical table structure used in the RI profiles DB is presented in Figure 3.3 defining the RI-GSC. The structure is a modified version of the class diagram of Figure 3.1 from [1]. Compared to the full adoption and application of the generic system configuration model, some simplifications and additions for the database structure have been made:

- The SystemConfigurationObject, *Constraints* and *System* classes have been removed.
  - *ControlModes, ControllerHostingCapacity, ProtectionModes* and *MeasuredData* tables have been added; these are used to model capabilities of control devices, protection devices and measurement devices, and can be viewed as specialized “constraints” with respect to the high-level class model.
  - Direct associations between related components have been introduced for selected component types (e.g., an amplifier can have an association to a real-time simulator).
- Numerous subclasses for different types of components have been introduced: this is the core specialization of the RI-GSC.
- A *ResearchInfrastructure* table has been added to be able to link objects to a particular RI.
- A *DomainTopology* table has been added to enable storing graphical representations of topologies in different domains.
- A *ModelFile* table has been added to enable storing binary files (e.g., CIM files) related to RIs in the database.

Several of the changes are motivated by a pragmatic approach to the modelling of system topologies: rendering all networks of a large RI into the terminals / domains / connectivity nodes structure constitutes a large effort. Further, maintaining a coherent application of the domains and terminals structures requires additional training and coordination.

Two special cases have therefore been considered: The *DomainTopology* class has been introduced to store an illustrated representation of network topology, i.e., a network drawing including the location of instrumentation such as e.g., measurement instruments. While the provision of the network illustration is mandatory, the full object rendering of the network (the representation of an electrical grid as individual database objects representing lines, busbars, breakers etc.) is defined as optional. This ensures that a human-readable representation of the network exists as a minimum, whereas a machine-readable representation can be created by RIs able to expend the resources to do so.

The second special case is the addition of two optional ways of directly expressing links between model elements. A special field is introduced for some component types to model associations between related components (e.g., *Amplifier, ThermalComponent*). In these associations the connection mode and domain is implicit. Similarly, an optional simplified description of communication links in free text has been introduced for *ControlDevice* components.

Likewise, mandatory and optional attributes (table fields) have been defined for all database objects. This ensures a minimum level of information while allowing individual RIs to create more detailed descriptions.

Despite the nature of the data structure as a CIM-inspired, a relational database has been selected for the implementation, rather than a triple store or other type of resource description framework (RDF) database, which is the preferred choice for CIM data. A number of reasons motivate this choice:

- The main strength of RDF databases / triple stores is the possibility to efficiently process queries relating to topologies and paths. Calculating the active path from one node to another through the network, for example to enumerate all circuit breakers and busbars which need to be reserved/locked in order to connect two nodes, would be difficult, impossible or inefficient to express in common relational database languages such as SQL due to the requirement for SELF JOINs. However, this type of request is not considered likely to occur in the application cases outlined earlier.

- RDF databases / triple stores require external enforcement of integrity constraints, which typically requires additional programming effort in wrapping database access. Relational databases offer built-in enforcement through type checks and normalization. In the given environment where almost 20 RIs with different views on their infrastructure and different skill levels are required to enter data manually and independently of each other, a properly designed relational database provides the opportunity to implement data integrity by design without the need to develop an access layer.
- Finally, but importantly, it has been considered that expertise on using relational databases (specifically SQL-based databases) is much more widespread than expertise on triple stores or similar databases. This has been estimated to apply to the participating RIs (which need to fill in the database) as well as to external users such as candidates for transnational access (which may need to query the database).

In total 36 different components were identified to be relevant for modelling the RIs in adequate detail. These components are further divided into six categories. In the DB logical structure, some of the categories are represented as superclasses. The modelled components and their classifications are the following:

- DER components: *GeneratingUnit, Load, StorageUnit, Converter, EVChargingStation, ElectricVehicle, Source, Amplifier*
- Thermal components: *ThermalSource, ThermalLoad, ThermalStorage*
- Connecting components: *ElectricalLineSegment, Transformer, ElectricalSwitch, CommunicationLineSegment, CommunicationInterchange, HeatExchanger, ThermalPipeline, ThermalCircuitEquipment*
- Control and monitoring components: *ControlDevice, ProtectionDevice, MeasurementDevice, LabSCADA*
- Cybersecurity control components: *SecureCommunication, Firewall, IntrusionDetectionSystem, Antivirus, NetworkSegmentation, AccessControls*
- ICT components: *SimulationCapacity, NetworkImpairmentSimulator, StorageAreaNetwork, VirtualServerEnvironment, ServerPC, RealTimeSimulator, Service*

Detailed data attributes have been defined for all of the objects and are represented in the Annex 9.3.

4 ERIGrid Database Instance

The ERIGrid RI profiles database includes detailed information on a large number of European RIs. Currently the database includes information on RIs of ERIGrid partners and is also published/opened (and later can be extended) for other users (e.g., TA potential users) through ERIGrid website.

Two versions of the database have been constructed. The first one is a public database, which is published as part of this deliverable and is integrated into ERGrid website. The second database contains all information from the first one but will additionally include data deemed sensitive or confidential by the RIs, e.g., cybersecurity information that cannot be published but is still important for ERIGrid project work. This database is only accessible to the ERIGrid consortium members. Confidentiality practices differ between ERIGrid partner organizations but adequate information on all RIs is needed for the development of the holistic testing procedure. To avoid a need to fill in the RI data twice (for the public and for the confidential database), data attributes which can be used to define whether a piece of information is confidential or public are introduced in the database definition.

This document describes the database design and gives instructions on how to fill in the relevant data. Besides, it includes the public part of the components and attributes information extracted from the database for each laboratory. All ERIGrid partners have added their RI information to the database and the completed database has been made. The database entries will also be updated during the whole project as necessary i.e., when new equipment is purchased, the RI configuration permanently changed or a completely new RI is constructed or added.

4.1 Technical Implementation

The ERIGrid RI profiles database is implemented in the PostgreSQL database management system, which includes an open source database engine [10]. The implementation is based on the logical data structure introduced in Section 3.2 and the data attributes defined in Annex 9.3. Implementation related details have not been defined as part of the data structure definition (e.g., how inheritance is to be implemented) and have been defined during the implementation process. Inheritance from *Component* class has been implemented by adding to the subclasses a foreign key referring to the *Component* table. The *Component* table is needed both in the terminal – connectivity node based connectivity modelling and when defining primary components for control and protection devices. Inheritance from *DERComponent, ThermalComponent* and *CyberSecurityControl* has been implemented by including the superclass attributes in the subclass tables and the superclasses do not have dedicated tables in the DB. *DERComponent, ThermalComponent* and *CyberSecurityControl* are abstract classes and cannot be instantiated by themselves. In addition, they do not have any such associations to other components that separate tables would be needed. The class and data attribute names have been slightly modified compared to the definitions in Section 3.2 and Annex 9.3. All capital letters have been removed and in the middle of names have been replaced with _ (e.g., ResearchInfrastructure → research_infrastructure). This has been done because the utilized database management system does not handle capital letters well.

The final database implementation is presented in the additional material in the ERIGrid website [5]. The material includes aSchemaSpy [6] description of the DB. This material includes information on the table structure, data formats, primary keys etc. and can be used to view the details of the DB. The graphical representation is too large to be added as a readable figure in this document but the basic idea of it is shown in Figure 4.1. The figure details can be viewed from the SchemaSpy description.

As mentioned, two versions of the database need to be constructed (public and confidential). This is handled by adding a “confidential” attribute to tables *ResearchInfrastructure, Component, DomainTopology* and *ModelFile* (see Annex 9.3). These attributes are used to remove all data marked as confidential from the public version of the database. Also, terminal data related to confidential components and connectivity nodes that do not have a link to any public terminals will be removed.

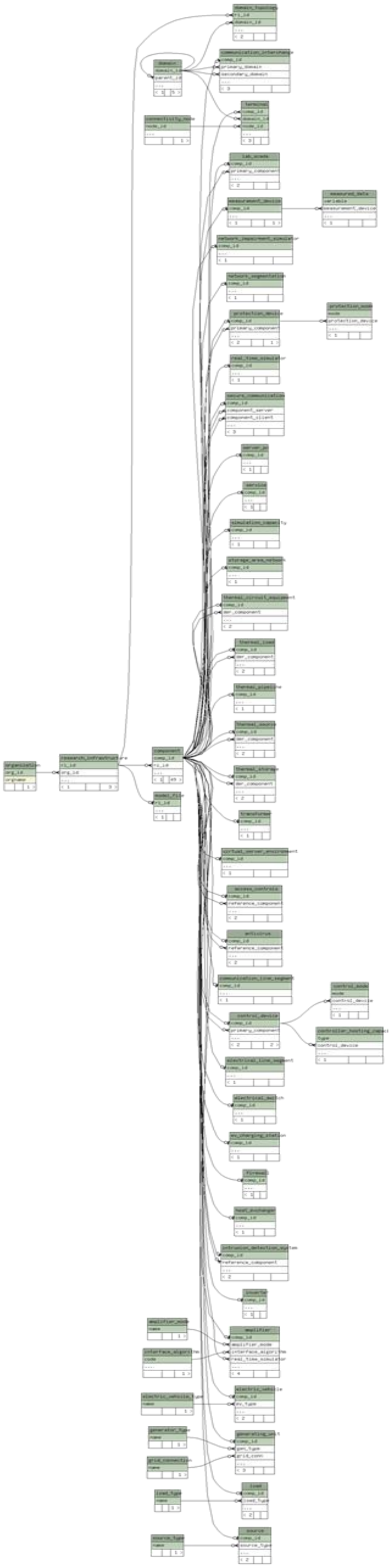


Figure 4.1: Schema of the implemented database

All ERIGrid partners have remote access to the database. This remote connection is used to enter the RI information and to query information on the RIs of other partners. The database is used through a simple web user interface.

4.2 Minimum Data Requirements

Minimum requirements for the data to be entered into the database are defined in this section. This data is deemed vital for the identified cases for using the database and has to be made available by all ERIGrid partners. In addition to this data, the partners may choose to enter more detailed information as well. The database structure is flexible and enables descriptions with different level of details.

The minimum dataset requested from the ERIGrid partners is primarily defined by the scope and the intended use of the database. In particular, the database content must allow to:

- Unequivocally define the laboratory characteristics, in terms of electrical network, DER and other electrical components, thermal network and components (if present), control topology (including control devices, control modes implemented and control hierarchy), measurement devices, ICT and cybersecurity components.
- Identify how the laboratory can be best accessed and fully exploited by other users, including other ERIGrid partners, e.g., for RI integration experiments, and by external users, e.g., as part of the transnational access programme.

The minimum data requirements defined in this deliverable consist of general modelling principles and detailed mandatory/optional definitions for all database object attributes. This section discusses the general principles on minimum data requirements. The data attribute information is included in Annex 9.3.

Data being entered into the DB should always be with the user perspective in mind. In particular, the following questions should always be answered when entering data:

- Does the input data contain adequate detail for an external user to fully understand the testing capability of my RI?
- Starting from a use case to be tested, can an external user find the minimum set of information that allows identifying the most suitable RI?
- Once the most suitable RI has been identified, does the DB contain sufficient information to allow the creation of test and experiment specifications?

Answers to the first two questions form the minimum data requirements; the last one can be fulfilled with more detailed information. The more detailed information can be entered into the DB but is not required from all RIs.

In order to correctly understand the *RI testing capability*, the following information must be available:

- Electrical network topology
- DER component characteristics
- Electrical component description
- Thermal components (if the RI allows to test also in thermal domain)
- Available measurements
- SCADA systems
- Real-time simulators and advanced testing components (as HIL systems)
- ICT characteristics and dedicated components
- Cybersecurity infrastructure (if existing)

Regarding the possible *use cases* that can be successfully tested in the RI, in addition to previous information, the DB must contain a minimum set of data with respect to:

- Available control devices and their relation to RI components
- Available control modes and their association with control devices
- The possibility to deploy third-party software on existing control devices (this may include different or new control modes)

The two previous sets of information allow to clearly identify the range of the ways in which a particular RI can be used and can be considered as minimum requirements for the data to be entered into the database.

The last set of information that allows to define *test and experiment specifications* are:

- Detailed connection of components as defined by terminals and connectivity nodes
- Detailed control device protocol information
- Detailed information on ICT network and cybersecurity components

This last set of information can be generally considered as optional but, in specific cases (e.g. if the RI is specifically designed also for cybersecurity tests), can be considered mandatory. Only the RI manager himself can perform an evaluation if any piece of data is particularly significant for the RI description or not.

4.2.1 Minimum Requirements for Modelling Network Topologies

In order to better clarify the RI characteristics, the following information describing available networks (electrical, thermal, control, communication) has to be available. A more detailed description of different approaches to describe network topologies is included in Section 5.1.

4.2.1.1 Electrical Network

Electrical network information has to be inserted following a common structure. The database provides two alternative ways of modelling the electrical network topology. Detailed modelling is possible using terminals and connectivity nodes as indicated in the logical table structure of the RI profiles database in Figure 3.3. This approach requires more modelling effort when entering the data into the database but, on the other hand, enables very accurate modelling and utilization of the data also e.g., as an input to algorithms that require network topology data.

In many cases, a picture of the electrical network structure should be adequate and it is not mandatory to use the terminals and connectivity nodes to model the topology. The network picture should preferably be a single-line diagram in which network parts that cannot be directly interconnected are highlighted e.g., by using different colours (e.g., transition from a 3ph to a 1ph sub-grid, transition from an AC to a DC sub-grid, different voltage levels). In the network diagram, the components must be labelled and component names must be consistent with the data entered into the database. In addition, the most significant measurement locations should be indicated in the diagram. At least all measurement devices listed in the DB have to be marked in the diagram and, preferably, all measurement locations should be indicated. The topology picture is stored in the *DomainTopology* table in the database and is mandatory for all RIs (even those that utilize also the modelling approach using terminals and connectivity nodes).

4.2.1.2 Thermal network

In ERIGrid, the concepts of terminals and connectivity nodes are extended also to other domains than electrical. Detailed modelling of thermal networks can, therefore, be conducted using the terminals and connectivity nodes.

Also for thermal networks, a simplified picture of the installed components and their connection should be added. In the network picture, the main components must be clearly indicated and component names must be consistent with the data entered into the database. A thermal circuit piping and instrumentation diagram (P&ID) is the preferred solution. Also, the thermal network topology figure is stored in the *DomainTopology* table.

4.2.1.3 Communication Network and Control Hierarchies

Also, communication networks can be modelled using the terminals and connectivity nodes and/or the graphical representation. Communication networks differ from the electrical and thermal networks such that connections have directions and the direction information is modeled using the *type-D* attribute in the *Terminal* table. Also, the endpoint type (e.g., master/slave) can be modelled using the *type-E* attribute in the *Terminal* table.

In many cases, a graphical representation is adequate. Communication networks can be of very high complexity and a very detailed description is only needed if an RI is designed and used also for specific test on communication. Usually a more simplified description (textual or graphical) that includes the main communication lines and devices (as hubs, routers, gateways), the connection to main components and the adopted protocol is adequate.

The simplest way of modelling control hierarchies is through using the textual descriptions in the *ControlDevice* table (*upperlevelControlDevice, lowerlevelControlDevice, horizontallevelControlDevice*). This is the minimum required level for control hierarchy modelling. A more detailed way of modelling control hierarchies is to combine the communication network modelled using terminals and connectivity nodes and the control level information of control devices stored in *ControlModes* and *ControllerHostingCapacity* tables.

Besides, a control network picture is very useful in order to understand control topology and available control devices and control modes in the RI. The picture should be a graphical representation of control devices (with implemented control modes). Control relations must be clearly defined regarding hierarchy direction (downstream/upstream/horizontal) and associated communication protocols. Component names must be consistent with the data entered into the database.

4.2.2 Minimum Requirements for Component Attributes

The mandatory data attributes are indicated in the tables of Annex 9.3 and will not be discussed further in this section. In addition to the information available in the tables, the following considerations must be taken into account for some specific components:

- **Communication connecting components:** If an RI foresees specific testing on communication, modelling communication connecting components (*CommunicationLineSegment* and *CommunicationInterchange*) is mandatory. In this case, all data attributes should be inserted.
- **Protection devices:** Protection devices used as testing components should be modelled in the database. For these devices, all attributes in *ProtectionDevice* and *ProtectionMode* tables are mandatory. It is not mandatory to model protection devices whose sole purpose is to protect some lab component. If these components are entered into the database, the RI can decide which data attributes are entered. Attributes *modeltype* and *primaryComponent* are the only mandatory ones in this case.
- **Cybersecurity control components:** Cybersecurity control components data attributes are mandatory if an RI foresees specific testing on cybersecurity. In this case, the minimum required data according to data attribute tables in Annex 9.3 should be inserted.

5 Tutorial on Filling and Querying the ERIGrid Database Instance

This section provides instructions on how to use the ERIGrid RI profiles database capabilities to model different types of RIs and how to query information from the database. The first objective of the section is to further explain how the data structure defined in Section 3.2 should be used to model different characteristics of the RIs so that the modelling decisions of different database users will be consistent. The second aim is to identify some basic questions that are foreseen to be relevant for many database users and to provide example SQL queries for these cases.

In the database, a research infrastructure means a set of resources that can be used together. If a partner has separate RIs that cannot be connected to each other, they should be separated as individual RIs even if they are located at the same place.

5.1 Modelling Examples

Modelling examples related to electrical networks, control hierarchies and real-time simulators are represented in this section.

5.1.1 Electrical Network

As previously described, a diagram of the electrical network structure is the minimum requirement for electrical network topology modelling. In the picture, the main components must be clearly indicated and component names must be consistent with the data entered into the database.

Figure 5.1 is an example of the minimum requirements of a laboratory electrical network description that includes DER components, line segments, transformers, switches and measurements.

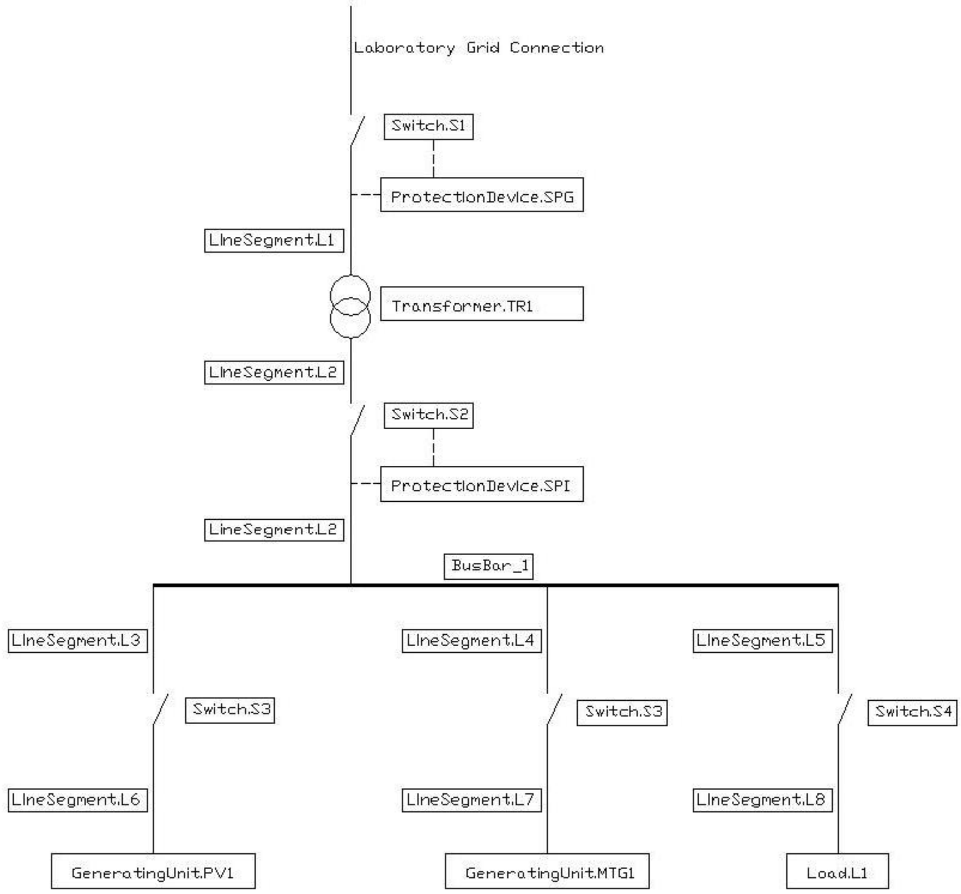


Figure 5.1: Example of minimum requirements of electrical network description

A more detailed electric network description should include in the graphical representation also terminals, as depicted in Figure 5.2, as well as summary tables of connectivity nodes as in Table 5.1.

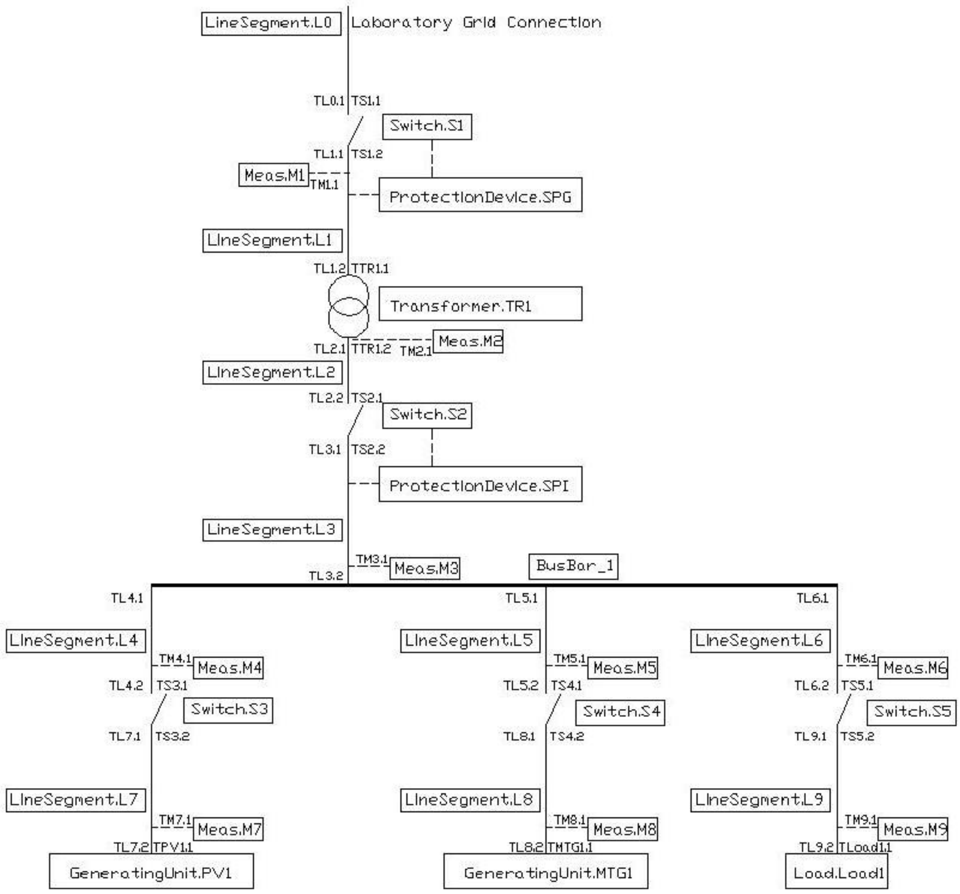


Figure 5.2: Example of electrical network description including component terminals

Table 5.1: Example of electrical network connectivity node table

Connectivity node	**Terminals**				
0	TL0.1	TS1.1			
1	TL1.1	TS1.2	TM1.1		
2	TL1.2	TTR1.1			
3	TL2.1	TTR1.2	TM2.1		
4	TL2.2	TS2.1			
5	TL3.1	TS2.2			
6	TL3.2	TL4.1	TL5.1	TL6.1	TM3.1
7	TL4.2	TS3.1	TM4.1		
8	TL7.1	TS3.2			

Connectivity node	Terminals				
9	TL7.2	TPV1.1	TM7.1		
10	TL5.2	TS4.1	TM5.1		
11	TL8.1	TS4.2			
12	TL8.2	TMTG1.1	TM8.1		
13	TL6.2	TS5.1	TM6.1		
14	TL9.1	TS5.2			
15	TL9.2	TLoad1.1	TM9.1		

The information of Table 5.1 can be represented in the database using the *Terminal* and *ConnectivityNode* tables.

5.1.2 Control Hierarchies

This section discusses the rendering of control hierarchies, i.e., descriptions of control devices and the relationships between them. A “control device” in this context is to be understood in the broadest sense as a hardware or software entity which takes part in the control and/or automation of the physical laboratory, and which is available to laboratory users on a permanent or semi-permanent basis such that it is considered to be part of the laboratory installation.

It may seem counter-intuitive, or inviting inconsistency, not to distinguish between software and hardware control devices. However, to the users of a laboratory, two main properties of a control device are of greater practical interest: Which built-in control capabilities does the control device offer, and does the control device permit the deployment of custom control strategies (if yes, in which way). Whether a two-level control hierarchy consists of two separate physical entities, or a single entity with two layers of software, is of much less consequence to the control capabilities of the lab.

The possibility of communication between the different components and parts of a laboratory determines whether the laboratory presents itself to the user as a collection of hardware components or as an integrated system. It also determines the degree of flexibility in the installation in terms of which types of external components can be part of the system under test.

The model adopted in the ERIGrid RI profiles database for modelling control hierarchies consequently focuses on these three properties of control devices: Built-in control capabilities (referred to as “control modes”), the ability to host custom control software (“controller hosting capacity”) and the communication between controllers within or across the control levels as defined in Section 3.1.2.

Communication between controllers is expressed in terms of upstream (from device towards supervisory controller), downstream (from supervisory controller towards device) and horizontal (between peer controllers) control relations. It must be noted that these control relations are different in nature from communication links in that they do not express the mere ability to communicate, but indicate the hierarchy of control, i.e., which of the entities is considered to be an “outer loop” or a “supervisory controller” relative to another one. These relations are tightly coupled to the features of the communication protocol in use – to implement a desired control relation, both sides have to (a) understand the same protocol, and (b) be able to express the relation through this protocol. For example, a peer-to-peer (horizontal) control relation can only be implemented very awkwardly by using the Modbus protocol, since Modbus relies on a clear and virtually unchangeable assignment of master and slave roles. This is true even if the underlying physical communication link supports bidirectional communication, and allows each partner to initiate the sending of information, such as is the case for e.g., Modbus/TCP over Ethernet: Each partner can send Ethernet packets to the other, but the Modbus protocol only allows the partner in the master role to initiate communication.

For this reason, and in line with the principles introduced in Section 3.2, two modelling styles for control relations are being offered: A mandatory simple model, which qualifies control relations in terms of hierarchy direction (downstream/upstream/horizontal) and an associated communication protocol. For RIs wishing to provide a more detailed description of their control capabilities, an optional CIM-compatible description using terminals and connectivity nodes can be additionally provided. In this description, each combination of communication medium and protocol is being assigned to a separate domain, such that two control devices with terminals linking to the same domain are assumed to be able to form a control relation. For example, a control device A with a terminal A1 implementing a Modbus/TCP slave and a control device B with a terminal B1 implementing a Modbus/RTU master cannot form a control relation without protocol translation. A1 and B1 therefore do not share a domain.

In the above example, a third control device C with a terminal C1 implementing a Modbus/RTU master would not be able to form a control relation with B1 despite both occupying the same domain, due to the lack of multi-master support in the Modbus protocol. More detail is therefore required to accurately describe the situation. The *type-D* (direction type) and *type-E* (endpoint type) fields in the *Terminal* table can be used to provide this information.

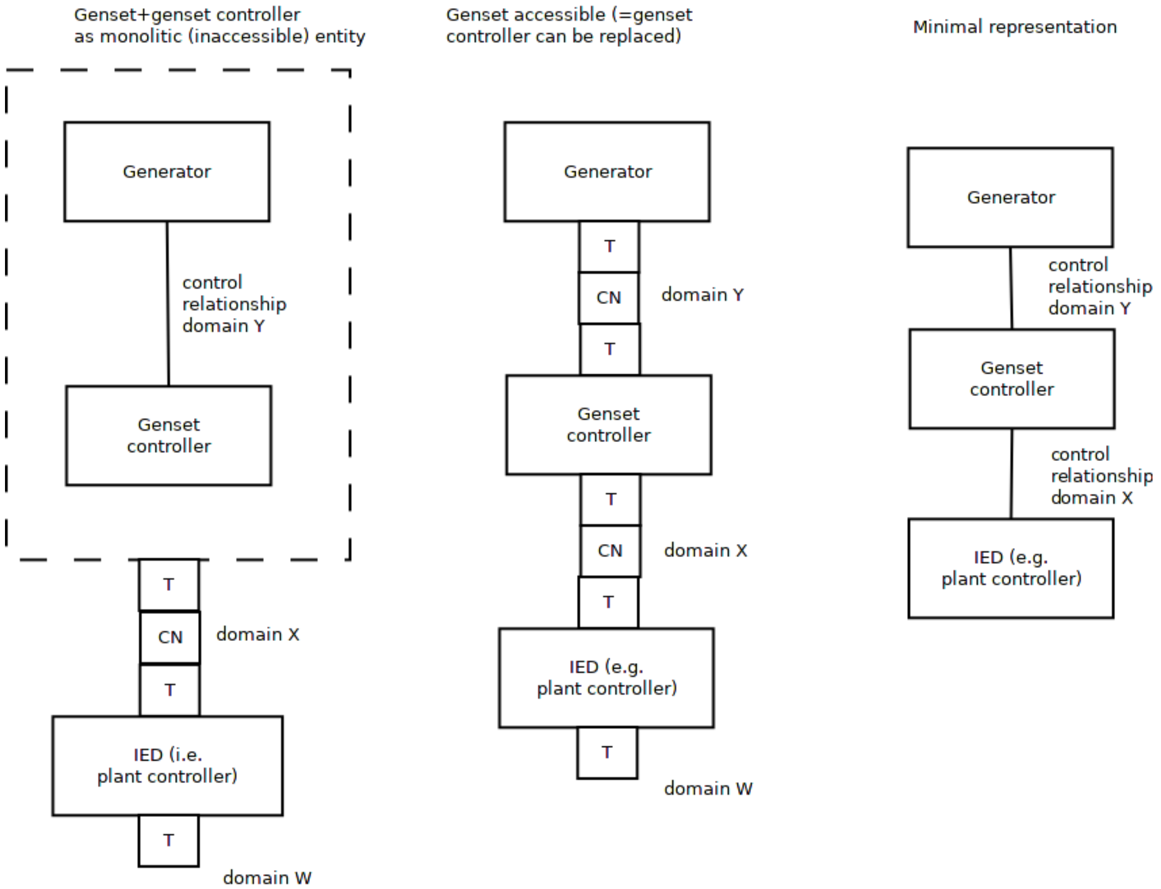


Figure 5.3: Minimal, detailed and hybrid modelling of control relations

Figure 5.3 provides an overview of the simple/mandatory and advanced/optional description options, as well as hybrid forms. The three panels represent three renderings of the same control hierarchy: A diesel generator set which is controlled by a DER unit controller, which in turn is controlled by an IED, for example a plant controller. The rightmost panel uses only mandatory information, i.e., the classification of control relations as upstream/downstream/horizontal and an associated protocol. In this case, control relations are defined in the *ControlDevice* table text fields *upperlevelControlDevice, lowerlevelControlDevice* and *horizontallevelControlDevice* using the following convention:{ControlDevice 1, protocol 1; ControlDevice 2, protocol 2; …; ControlDevice n, protocol n} (see Annex 9.3). The control devices specified in these fields represent the default lab configuration and possibilities to communicate with external controllers are not explicitly modelled. Theoretically, any of the default connections can be separated and an external device inserted instead of the original one. In practice, some of the connections cannot be utilized this way due to for instance safety, warranty and insurance related matters. This is a matter of lab policy rather than lab capability and, therefore, will be covered in [3].

The center panel adds the CIM-compatible information and expresses the control relations using terminals and connectivity nodes.

The leftmost panel represents a hybrid of both styles. It could e.g., be used if the detailed model of a control relation is considered to be irrelevant for the purposes of the database. If the genset controller is manufacturer fitted, embedded into the genset and the communication between genset and controller cannot be accessed, this relation could be modelled using the simple style. The upstream/outside fieldbus interface of the genset controller, on the other hand, may be accessible to laboratory users and is being modelled in detail.

The term “control device” spans a broad range of functionality and complexity. Some control devices have a single, hardcoded function, others are programmable multi-function devices. The same applies to the hosting capacity of control devices: Many control devices do not permit the deployment of any custom software, whereas others may be based on off-the-shelf embedded computers and support almost any kind of executable code in many languages. For this reason, control modes and hosting capacities have been modelled as separate table objects, each of which can be associated with a control device, which is then understood to support the respective control mode or hosting capability. For this reason, control devices do not have a defined control level D1...D5 according to Section 3.1.2. Rather, a level is associated with each control mode or hosting capability. This allows the representation of general purpose computers as controllers, which may have the ability to host an IED function and/or a supervisory control function controlling the IED, at the same time.

5.1.2.1 Modelling Examples

The control and automation infrastructures available at the participating RIs in the ERIGrid project are highly diverse. The following two modelling examples demonstrate these differences. In both cases, the figures are divided into an upper half expressing the physical relation between devices and the communication between these. The lower half represents the mapping to the controller model used for the ERIGrid database.

The first example (Figure 5.4) illustrates how to model a PV inverter controlled through a traditional SCADA system. The simple/mandatory rendering option has been chosen in order to keep the drawing readable. In the example, the PV inverter is equipped with an embedded, factory-fitted DER controller. A remote terminal unit (RTU) serves as the local endpoint of a central SCADA system and is used to translate the protocol supported by the inverter (Sunspec) to the protocol used internally by the SCADA system (IEC61850). A “pass-through” control mode has been added to the RTU to illustrate the mere gateway function of the RTU.

The central SCADA system provides real-time and long-term databases and other functions. Two workstations are connected to the SCADA server, a human machine interface (HMI) workstation that provides control room functionality and built-in system-wide supervisory controllers, and a developer workstation that allows the deployment of custom supervisory controllers. In this example, the HMI workstation contains a microgrid management application, as well as dashboard/control panel applications, which allow the manual control of the system.

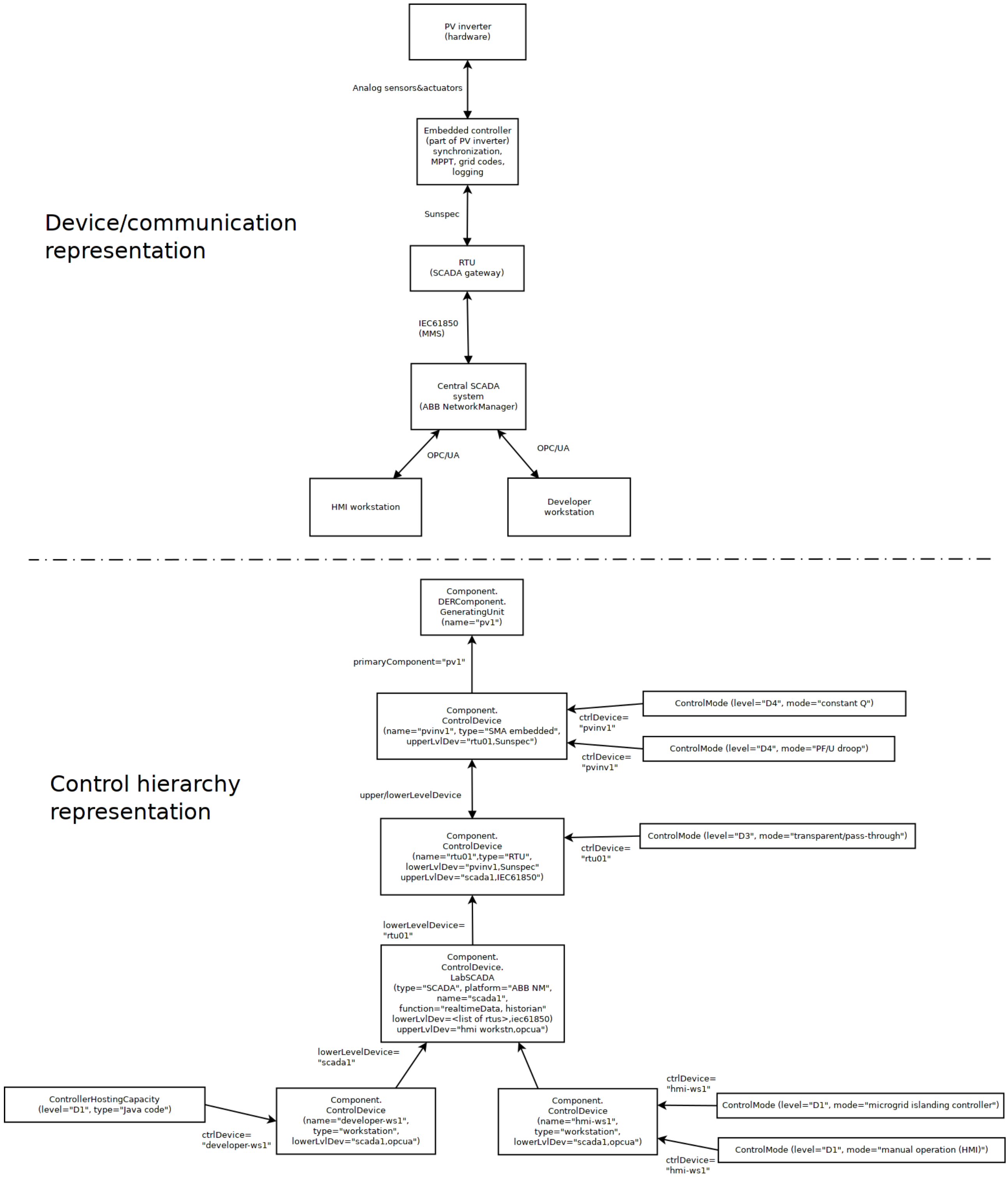


Figure 5.4: Control hierarchy modelling of a central lab SCADA system

Only the HMI workstation is listed as an upper level control device of the SCADA system. This may be used to express the idea that the developer workstation (or a visitor laptop) are not considered to be part of the permanent lab installation. Since the SCADA system offers an upstream OPC (open platform communications) connection which can potentially be used by many clients at the same time, the list of potential clients may be too long to explicitly provide, for example if the SCADA system is theoretically open for access from any staff laptop. In this case, only the control room HMI is listed as a permanently connected client.

The second example (Figure 5.5) illustrates how to model a less traditional control infrastructure, in this case represented by the “SYSLAB” RI at DTU. SYSLAB does not employ a central SCADA system; instead, every DER and other controllable physical device are equipped with a “SYSLAB node”, a multi-purpose control computer. Control hierarchies are entirely defined by the software run on the nodes; therefore, a node may occupy roles at different control levels at the same time. Control relations between nodes, including the definition of upstream or downstream, are specific to each experiment. While this setup offers great flexibility, this flexibility is difficult to describe in a static laboratory database because part of the control relations can be dynamically defined. In this case, the flexibility is modeled by defining horizontal control relations between all nodes in the entire laboratory, and by associating a controller hosting capacity at multiple control levels with every node, such that any node may be performing control functions at the device level (D3) or at the system level (D1).

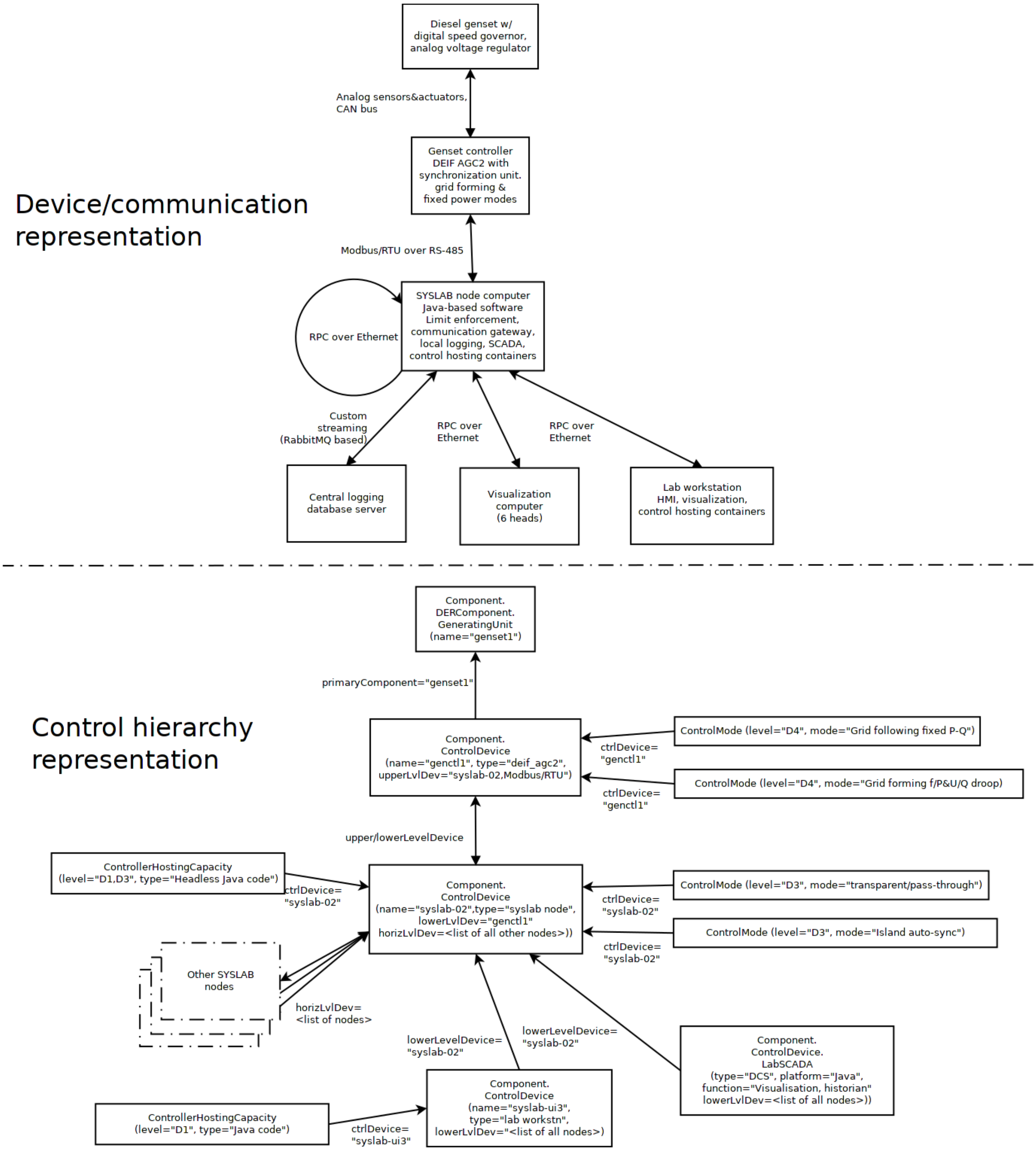


Figure 5.5: Control hierarchy modelling of a modular distributed control system

Similar to the first example, no upper level controllers are defined for the SYSLAB nodes. Because any computer within the lab network is able to access data from, and send control commands to any node, and because these control relations would be established ad-hoc as dictated by the need of an experiment, they are not considered permanent. A random lab workstation is modelled as an example for a machine which could be used to deploy a supervisory controller for a specific experiment. A “control room” visualization system and a central data store which replicates the data that is logged locally on each node have been added as another example.

5.1.3 Real-Time Simulator

In this section, a tutorial of possible connections of a virtual entity simulated by a real-time simulator with other laboratory devices is shown. The real-time simulator tool is used worldwide for the closed-loop testing of protection and control equipment (Controller Hardware-In-the-Loop, CHIL) as well as hardware equipment (PHIL). A vast library of models provided by the simulator allows the user to design power systems with various components and operate them in real-time. The connection of the real-time simulator with other components changes depending on the test case. The test case dependent connections should not be modelled in the RI profiles database using terminals and connectivity nodes. The DB includes a possibility to associate amplifiers with a real-time simulator using a dedicated attribute in the *Amplifier* table but otherwise the non-permanent connections are not modelled in the table structure. However, a graphical representation on the capabilities of the real-time simulator can be useful. This graphical representation can be stored in the *DomainTopology* table.

An example case on real-time simulator modelling is provided below. The real-time simulator used for the example is from RTDS®. The connection of the RTDS with the rest of the laboratory devices of the ICCS/NTUA Power Systems Laboratory is depicted in Figure 5.6. The figure describes the connection of the RTDS with the power amplifier device needed for the PHIL test case as well as the connection with a server PC, which aggregates and stores test data while also offers the ability of monitoring the testing procedure involving laboratory devices including the RTDS. However, taking into account the aforementioned ability of the RTDS to participate in multiple test cases which result in different connection possibilities than the ones shown (with the exception of the connection with the server PC which is permanent and does not depend on test case variations), the figure cannot really encompass the full extent of the RTDS connections.

The RI description method established in this document offers the ability to model RTDS connections as well as RI infrastructure in general in a uniformed and comprehensive way. Although the alternative connections of the RTDS should not be modelled using the *Terminal* and *ConnectivityNode* tables, the modelling approach can be utilized when composing the graphical representation of the RTDS capabilities. Such a modelling of RTDS connections based on the database format is shown in Figure 5.7. In this figure, every possible connection of the RTDS within the laboratory infrastructure that is test case dependent is illustrated.

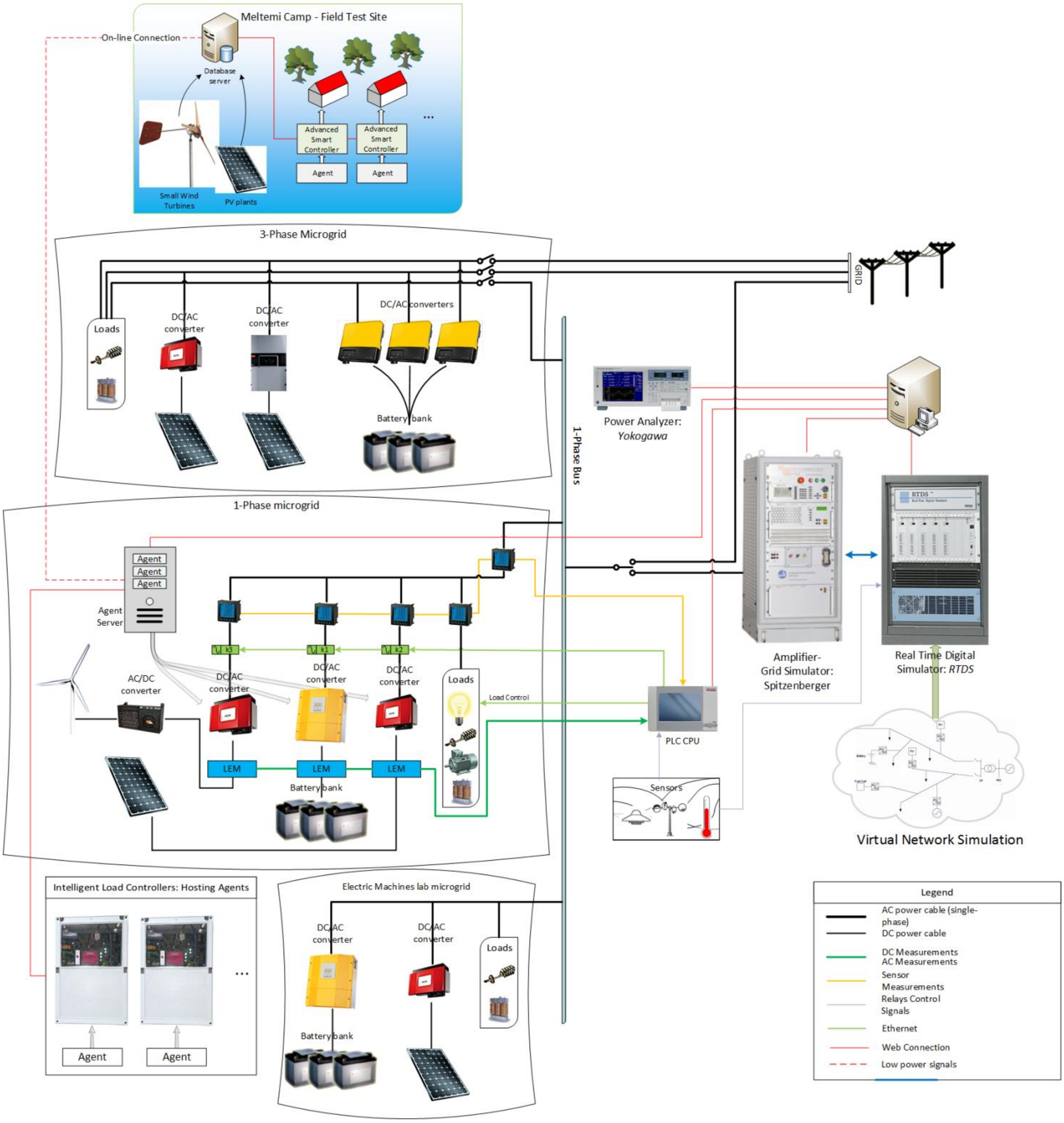


Figure 5.6: General layout of the ICCS/NTUA Power Systems Laboratory

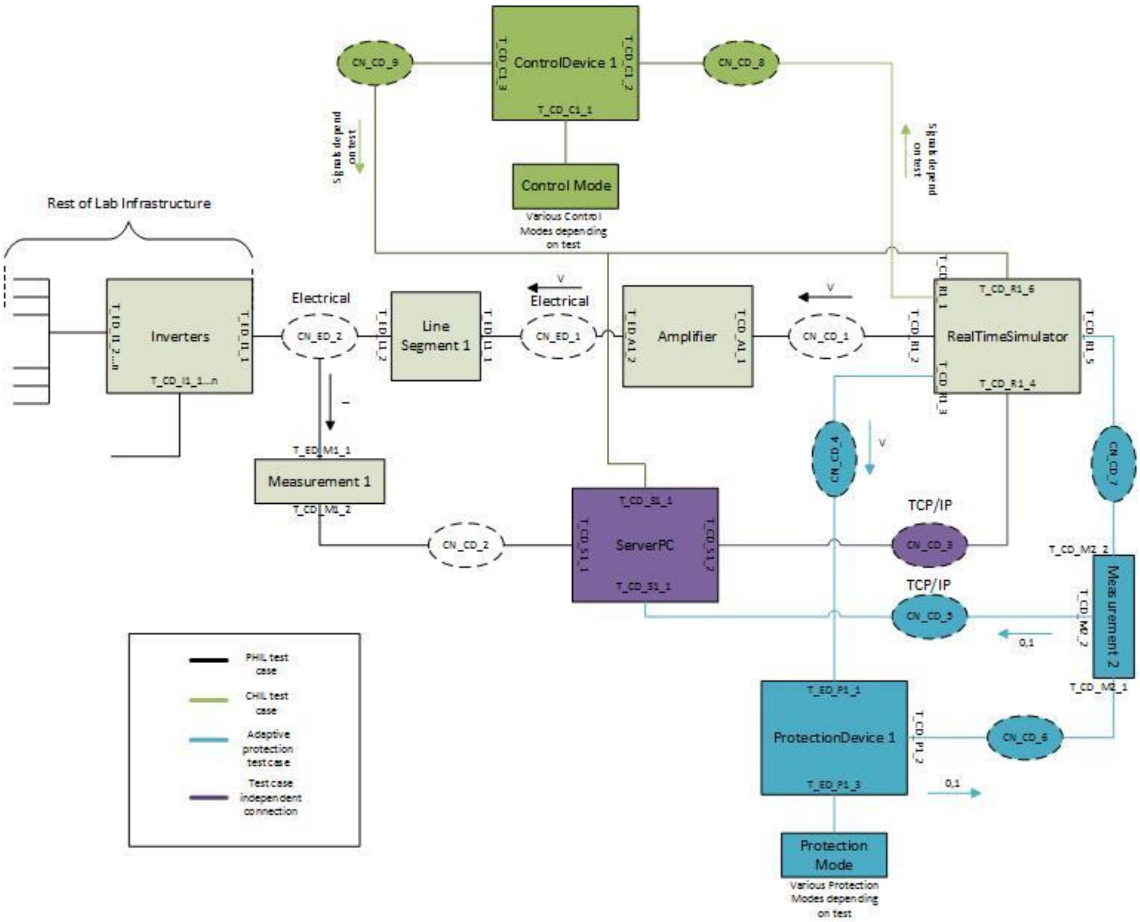


Figure 5.7: Possible connections of the RTDS within the laboratory infrastructure

The power amplifier is associated with the RTDS through attribute *realTimeSimulator* in the *Amplifier* table. An association between the RTDS and the control and protection devices in Figure 5.7 is not made in the static tables.

5.2 Query Examples

The ERIGrid RI profiles database is used through a simple web user interface that does not have advanced functionalities. A graphical user interface with a multitude of predefined queries can be implemented at a later stage but is not included in the ERIGrid project scope with the exception of the extracted component data available at the ERIGrid website. For now, the users of the whole database need to formulate their questions using the SQL language. This section gives some examples on queries that are foreseen to be relevant for database users but does not provide an exhaustive list of possible queries.

The database was designed such that the person filling in the RI data can decide the level of details to be entered into the DB as long as the minimum requirements discussed in Section 4.2 are fulfilled. Since the minimum required representation is not the best adapted for queries, the queries presented in this section will not have the same outcome for all RIs. Along the same lines, data in the database is in several cases more suitable for human users than for automated queries. Examples are the text fields containing reference and protocol information in free text, as *in upperlevelControlDevice, lowerlevelControlDevice* and *horizontallevelControlDevice* in the *ControlDevice* table.

Two types of cases for database use are considered in this section: Exploration of suitable laboratories and Mapping of test system to RI.

5.2.1 Use Case 1: Exploration of suitable laboratories

This section provides some example questions that can be relevant to explore the capabilities of different RIs included in the database.

1. What communication protocols are supported in a given RI?

The database offers two alternative ways to model the protocols supported in an RI: the full SC-oriented model using domains/terminals for associating devices with protocols, or the minimal approach, using text fields. For RIs modelled in the latter approach, the information will not be harmonized or might not be available at all.

If the domain-based modelling approach is used, the protocols are associated with component terminals and the protocol information can be obtained from the *Domain* table as communication subdomains. These can be queried by building a list of all the domains that are subdomains of the domain “communication” using the following SQL statement:

```
WITH RECURSIVE domains (domain_id, name, parent_id) AS (
    SELECT domain_id, name, parent_id FROM domain WHERE name = 'communication'
  UNION ALL
    SELECT d2.domain_id, d2.name, d2.parent_id
    FROM domains d1, domain d2
    WHERE d1.domain_id = d2.parent_id
  )
SELECT name FROM domains
```

If communication protocols are not modelled as their own domains but terminals are used in modelling, the protocols can be obtained by finding all terminals associated with communication domains and by cross-referencing the terminal list with the communication domain topology illustration. All terminals associated with domains obtained from the previous query in the given RI (RI_name) can be queried using the following SQL statement:

```
WITH RECURSIVE domains (domain_id, name, parent_id) AS (
    SELECT domain_id, name, parent_id FROM domain WHERE name = 'communication'
  UNION ALL
    SELECT d2.domain_id, d2.name, d2.parent_id
    FROM domains d1, domain d2
    WHERE d1.domain_id = d2.parent_id
  )

SELECT t.name, t.node_id, c.comp_name, d.name, ri.name
FROM terminal t
JOIN domain d ON t.domain_id = d.domain_id
JOIN component c ON t.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE (d.name IN (SELECT name FROM domains) AND ri.name = 'RI_name')
```

The communication domain topology illustration can be queried using the following SQL statement:

```
SELECT dt.picture
FROM domain_topology dt
JOIN domain d ON dt.domain_id = d.domain_id
JOIN research_infrastructure ri ON dt.ri_id = ri.ri_id
WHERE (ri.name = 'RI_name' AND d.name = 'communication')
```

If the RI utilizes only minimal descriptions of communication by the text fields in *ControlDevice* (*up-*

perlevelControlDevice, *lowerlevelControlDevice, horizontallevelControlDevice*), *ProtectionDevice* (*intProtocol*) and *MeasurementDevice* (*upstreamProtocol*), the protocol information can be obtained using the following approach:

a) List for all protection devices belonging to the RI of interest and the contents of the *intProtocol* field. These can be queried using the following SQL statement:

```
SELECT c.comp_name, pd.int_protocol
FROM protection_device pd
JOIN component c ON pd.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE ri.name = 'RI_name'
```

b) List all measurement devices belonging to the RI of interest and the contents of the *upstreamProtocol* field. These can be queried using the following SQL statement:

```
SELECT c.comp_name, md.upstream_protocol
FROM measurement_device md
JOIN component c ON md.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE ri.name = 'RI_name'
```

c) List all control devices and the contents of the *upstreamControlDevice/downstreamControlDevice/horizontalControlDevice* text fields. Protocol related information might be contained here. *LabSCADA* is a subclass of *ControlDevice* and, therefore, its values should also be included. These can be queried using the following SQL statement:

```
SELECT c.comp_name, cd.upperlevel_control_device, cd.lowerlevel_control_device,
cd.horizontallevel_control_device
FROM control_device cd
JOIN component c ON cd.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE ri.name = 'RI_name'
UNION
SELECT c.comp_name, ls.upperlevel_control_device, ls.lowerlevel_control_device,
ls.horizontallevel_control_device
FROM lab_scada ls
JOIN component c ON ls.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE ri.name = 'RI_name'
```

In short, if communication protocols are modeled by domains, the protocol information is harmonized independent of the RI, and directly available from the first query.

2. *Which RIs have a real-time simulator with PHIL capability?*

To be PHIL capable, an RI has to be able to connect a real-time simulator and an amplifier that is suitable for PHIL testing. Then the amplifiers need to be compared in their features. The *Amplifier* table needs to be queried for amplifiers that are associated with a real-time simulator using the *realTimeSimulator* data attribute. Relevant information on these amplifiers can be queried using the following SQL statement:

```
SELECT ri.name, c.comp_name, a.rated_u, a.rated_s, a.phases
FROM amplifier a
JOIN component c ON a.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE (ri.name = 'RI_name' AND a.real_time_simulator IS NOT NULL)
```

If more amplifier fields are needed for the evaluation, these should be added at the SELECT statement in the first row.

3. *Which voltage levels are supported by the available RIs?*

The RI voltage level (MV, LV or none) is an attribute in the ResearchInfrastructure table. The voltage levels of all the RIs modelled in the DB can be queried using the following SQL statement:

```
SELECT name, voltage_level
FROM research_infrastructure
```

4. *What components in an RI have terminals in the electrical domain?*

If the RI utilizes terminals in modelling, the terminals in electrical domain can be obtained using the following procedure, similar to question 1:

a) Find the electrical domains (or subdomains of interest). These can be queried by searching for all the domains that are subdomains of domain “electrical” using the following SQL statement:

```
WITH RECURSIVE domains (domain_id, name, parent_id) AS (
    SELECT domain_id, name, parent_id FROM domain WHERE name = 'electrical'
  UNION ALL
    SELECT d2.domain_id, d2.name, d2.parent_id
    FROM domains d1, domain d2
    WHERE d1.domain_id = d2.parent_id
  )
SELECT name FROM domains
```

b) Find the components, which have terminals in all electrical domains. These can be queried using the following SQL statement:

```
WITH RECURSIVE domains (domain_id, name, parent_id) AS (
    SELECT domain_id, name, parent_id FROM domain WHERE name = 'electrical'
  UNION ALL
    SELECT d2.domain_id, d2.name, d2.parent_id
    FROM domains d1, domain d2
    WHERE d1.domain_id = d2.parent_id
  )
SELECT c.comp_name, t.name
FROM terminal t
JOIN domain d ON t.domain_id = d.domain_id
JOIN component c ON t.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE (d.name IN (SELECT name FROM domains) AND ri.name = 'RI_name')
```

If terminals are not used in the modelling of the RI that is under examination, the user has to search directly for component types known to have electrical terminals. In the ERIGrid database, these include all subclasses of *DERComponent* and *ElectricalLineSegment, Transformer* and *ElectricalSwitch*. These components can be queried using the following SQL statement:

```
WITH components AS (
    SELECT c.comp_id
    FROM component c JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
    WHERE ri.name = ('RI_name')
    )
SELECT c.comp_name
```

```
FROM generating_unit gu JOIN component c ON gu.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM load l JOIN component c ON l.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM storage_unit su JOIN component c ON su.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM converter i JOIN component c ON i.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM ev_charging_station evsc JOIN component c ON evsc.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM electric_vehicle ev JOIN component c ON ev.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM source s JOIN component c ON s.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM amplifier a JOIN component c ON a.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM electrical_line_segment els JOIN component c ON els.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM transformer t JOIN component c ON t.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name
FROM electrical_switch es JOIN component c ON es.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
```

5. Which infrastructure offers the interconnection of largest number of DER?

The database includes several subclasses of *DERComponent* (*GeneratingUnit, Load, StorageUnit, Converter, EVChargingStation, ElectricVehicle, Source, Amplifier*). The following SQL statement presents how the number of generating units in each RI can be queried:

```
SELECT COUNT(gu.comp_id), ri.ri_id
FROM generating_unit gu
JOIN component c ON gu.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
GROUP BY ri.ri_id
```

The number of other types of DERs can be queried similarly.

6. Which RIs offer the coupling of DER to a heat network?

The coupling of DERs and thermal components is modelled in the database by referencing associated DER components in the *ThermalComponent* table. For RIs that have coupled thermal and electrical systems, the *ThermalComponent* table as well as *HeatExchanger, ThermalPipeline* and *ThermalCircuitEquipment* can be queried. The investigation should therefore consider two stages: First, to look for RIs that have *ThermalComponents* (paired with *DERcomponents*) using the following SQL statement:

```
WITH thermal_components (comp_id, comp_name, der_component) AS (
    SELECT c.comp_id, c.comp_name, tl.der_component
    FROM thermal_load tl JOIN component c ON tl.comp_id = c.comp_id
    WHERE tl.der_component IS NOT NULL
    UNION
    SELECT c.comp_id, c.comp_name, tst.der_component
    FROM thermal_storage tst JOIN component c ON tst.comp_id = c.comp_id
    WHERE tst.der_component IS NOT NULL
    UNION
    SELECT c.comp_id, c.comp_name, tso.der_component
    FROM thermal_source tso JOIN component c ON tso.comp_id = c.comp_id
    WHERE tso.der_component IS NOT NULL
    )
    SELECT thermal_components.comp_name, derc.comp_name, ri.name
    FROM thermal_components
    JOIN component derc ON thermal_components.der_component = derc.comp_id
    JOIN research_infrastructure ri ON derc.ri_id = ri.ri_id
    ORDER BY ri.name
```

Secondly, for each RI then it can be investigated what heat network infrastructure is available by querying the *HeatExchanger, ThermalPipeline* and *ThermalCircuitEquipment* per RI:

```
WITH components AS (
    SELECT c.comp_id
    FROM component c JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
    WHERE ri.name = ('RI_name')
    )
SELECT c.comp_name, c.comp_id
FROM heat_exchanger he JOIN component c ON he.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name, c.comp_id
FROM thermal_pipeline tp JOIN component c ON tp.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
UNION
SELECT c.comp_name, c.comp_id
FROM thermal_circuit_equipment tce JOIN component c ON tce.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
```

5.2.2 Use Case 2: Mapping of test system to RI

This section provides some example questions that can be relevant in the mapping process of the holistic testing procedure introduced in Section 2.1.

1. Which RIs have certain components available (e.g., a PV unit and a battery storage unit)?

A test case often requires some specific components to be available. For example, testing an algorithm that is used to optimize the operation of a PV unit and a battery storage unit in a household requires these two components. The RIs that have these two components installed can be queried using the following SQL statement:

```
WITH pv_ris AS (
    SELECT c.ri_id
    FROM generating_unit gu
    JOIN component c ON gu.comp_id = c.comp_id
    WHERE gu.gen_type = 'PV'),
battery_ris AS (
    SELECT c.ri_id
    FROM storage_unit su
    JOIN component c ON su.comp_id = c.comp_id
    WHERE su.type = 'battery')
SELECT pv_ris.ri_id
FROM pv_ris JOIN battery_ris ON pv_ris.ri_id = battery_ris.ri_id
GROUP BY pv_ris.ri_id
```

2. *What control modes does a specific DER component support?*

The control modes of a specific DER component can be found using the following procedure:

a) The control devices associated with the specific DER component are found through the *primaryComponent* attribute in *ControlDevice* table using the following SQL statement when the DER component *comp_id* attribute is known:

```
SELECT c.comp_name, c.comp_id
FROM control_device cd
JOIN component c ON cd.comp_id = c.comp_id
WHERE (cd.primary_component = der_id)
```

b) The control modes of all the control devices found in the previous step are found through the *controlDevice* attribute in *ControlModes* table using the following SQL statement:

```
SELECT mode, control_level
FROM control_mode
WHERE (control_device IN (cd_id1, cd_id2, …, cd_idn))
```

3. *Does the given RIs support measurement and data logging at the required time resolution?*

The data logging facilities are described in the *LabSCADA* table in attribute *dataLoggingFacilities*. In addition, the *employedStandard* field can be useful. The values in these fields for the given RIs can be queried using the following SQL statement:

```
SELECT c.comp_name, ls.data_logging_facilities, ls.employed_standard
FROM lab_scada ls JOIN component c ON ls.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE ri.name IN ('RI_name1', 'RI_name2')
```

4. *How can a control software for a specific type of DER be deployed in the given RI?*

To deploy new control functions/algorithms inside an RI, the *ControllerHostingCapacity* table describes the available platforms for embedding custom controllers, identifying e.g., the programming languages and interfaces available. To ensure the availability of an interface to control the respective DER, the *ControllerHostingCapacity* property should be associated with a *ControlDevice* that has direct control hierarchy access to the *DERComponent* in question. A first query can identify the *ControlDevices* that refer to the required *DERComponent* in their *primaryComponent* attribute, and then to list the associated entries from the *ControllerHostingCapacity* table. For example, the hosting capacities of control devices controlling any generating unit in a specific RI (‘RI_name’) can be queried using the following SQL statement:

```
SELECT chc.type, chc.control_level, chc.description, cd.comp_id, c.comp_name,
gu.comp_id
FROM controller_hosting_capacity chc
JOIN control_device cd ON chc.control_device = cd.comp_id
JOIN component c ON cd.comp_id = c.comp_id
JOIN generating_unit gu ON gu.comp_id = cd.primary_component
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE (ri.name = 'RI_name')
```

If the available platforms are not satisfactory, a second step can be to first search for the desired hosting platform in the *type* attribute of the *ControllerHostingCapacity* entries associated with the respective RI using the following SQL statement:

```
SELECT chc.type, chc.control_level, chc.description, cd.comp_id, c.comp_name
FROM controller_hosting_capacity chc
JOIN control_device cd ON chc.control_device = cd.comp_id
JOIN component c ON cd.comp_id = c.comp_id
JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
WHERE (ri.name = 'RI_name')
```

Once suitable entries are found, the associated entries from the *ControlDevice* table can be manually searched for an available control hierarchy path to the required DER component using the fields *lowerLevelControlDevice* and *upperLevelControlDevice:*

```
SELECT c.comp_name, cd.upperlevel_control_device, cd.lowerlevel_control_device
FROM control_device cd
JOIN component c ON cd.comp_id = c.comp_id
WHERE cd.comp_id IN (cd_id1, cd_id2, …, cd_idn)
```

6 Public RI Information at ERIGrid website

Component information extracted from the database instance filled with ERIGrid partner data is publicly available at the ERIGrid website. The whole database as such is not published at the website but the data is prepared in a way which is more easily understandable for domain experts. SQL queries might provide a too complex user interface for many users and, therefore, table format of the most relevant information was selected. The tables provide the information that is relevant when making the first selection for suitable RIs for a specific test e.g. preceding a TA application. After a successful TA application also, the whole database instance can be made available to TA users for more detailed planning of testing.

Due to different confidentiality practices between ERIGrid partners, the amount of publicly available data differs between RIs.

6.1 Selected Information

The information selected for the website consists of information regarding some selected component types and contains also the most important data attributes of the selected components. The components and attributes selected to be presented on the website are presented in Table 6.1, Table 6.2, Table 6.3 and Table 6.4. The selected components are considered to be the most relevant for the first assessment of an RI’s capability for performing a particular test and for instance all connecting components such as electrical line segments have been omitted from the list.

Table 6.1. DER components and their attributes represented at the website

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
Generating unit	Type	GeneratingUnit: type
	Name	Component: name
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	Description	Component: description
Load	Type	Load: type
	Name	Component: name
	Rated P [VA]	DERComponent: ratedP
	Rated Q [VA]	DERComponent: ratedQ
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	Description	Component: description
Storage unit	Type	StorageUnit: type
	Name	Component: name
	Rated P [VA]	DERComponent: ratedP
	Rated Q [VA]	DERComponent: ratedQ
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	Rated capacity [Wh]	StorageUnit: ratedCapacity
	Description	Component: description
Converter	Name	Component: name

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
	Type	Converter: convType
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	Description	Component: description
EV charging station	Name	Component: name
	Rated P [VA]	DERComponent: ratedP
	Rated U [V]	DERComponent: ratedU
	Description	Component: description
Electric vehicle	Type	ElectricVehicle: type
	Name	Component: name
	Description	Component: description
Source	Type	Source: type
	Name	Component: name
	Rated P [VA]	DERComponent: ratedP
	Rated Q [VA]	DERComponent: ratedQ
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	AC/DC	Source: ACDC
	Description	Component: description
Amplifier	Mode	Amplifier: AmplifierMode
	Name	Component: name
	Rated P [VA]	DERComponent: ratedP
	Rated Q [VA]	DERComponent: ratedQ
	Rated S [VA]	DERComponent: ratedS
	Rated U [V]	DERComponent: ratedU
	Description	Component: description

Table 6.2. Thermal components and their attributes represented at the website

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
Thermal source	Type	ThermalSource: type
	Name	Component: name
	Rated Pth [W]	ThermalComponent: ratedPth
	Description	Component: description
Thermal load	Type	ThermalLoad: type
	Name	Component: name
	Rated Pth [W]	ThermalComponent: ratedPth
	Description	Component: description
Thermal storage	Type	ThermalStorage: type
	Name	Component: name

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
	Rated Pth [W]	ThermalComponent: ratedPth
	Rated capacity [Wh]	ThermalStorage: ratedCapacity
	Description	Component: description

Table 6.3 Control and monitoring components and their attributes represented at the website

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
Control device	Name	Component: name
	Model type	ControlDevice: modeltype
	Description	Component: description
Protection device	Name	Component: name
	Model type	ProtectionDevice: modeltype
	Description	Component: description
Measurement device	Name	Component: name
	Model type	MeasurementDevice: type
	Description	Component: description

Table 6.4 ICT components and their attributes represented at the website

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
Simulation capacity	Name	Component: name
	Size of RAM [Byte]	SimulationCapacity: RAM
	Size of CPU [Hz]	SimulationCapacity: CPU
	Description	Component: description
Network impairment simulator	Name	Component: name
	Description	Component: description
Storage area network	Name	Component: name
	Size of HDD [Byte]	StorageAreaNetwork: hdd
	Description	Component: description
Virtual server environment	Name	Component: name
	Size of RAM [Byte]	VirtualServerEnvironment: ram
	Size of CPU [Hz]	VirtualServerEnvironment: cpu
	Size of HDD [Byte]	VirtualServerEnvironment: hdd
	Description	Component: description
Server PC	Name	Component: name
	Size of RAM [Byte]	ServerPC: ram
	Size of CPU [Hz]	ServerPC: cpu
	Size of HDD [Byte]	ServerPC: hdd
	Description	Component: description
Real-time simulator	Name	Component: name
	Max node number	RealTimeSimulator: maxNodeNmbr

Component	Data attribute title on the website	RI DB table: attribute (see annex 9.3)
	Description	Component: description
Service	Name	Component: name
	Type	Service: type
	Description	Component: description

The information is extracted from the public database with simple SQL queries. As an example, the query used to obtain generating unit data for a certain RI is represented below. Other queries are similar with the only differences in the table name and the attributes that are queried for.

```
WITH components AS (
    SELECT c.comp_id
    FROM component c JOIN research_infrastructure ri ON c.ri_id = ri.ri_id
    WHERE ri.name = ('RI_name')
    )
SELECT gu.gen_type, c.comp_name, gu.rated_s, gu.rated_u, c.description
FROM generating_unit gu JOIN component c ON gu.comp_id = c.comp_id
WHERE (c.comp_id IN (SELECT * FROM components))
```

6.2 Representation at the Website

The component information represented in the previous section has been added to the ERIGrid transnational access website [2] as an additional link under each RI as shown in Figure 6.1.

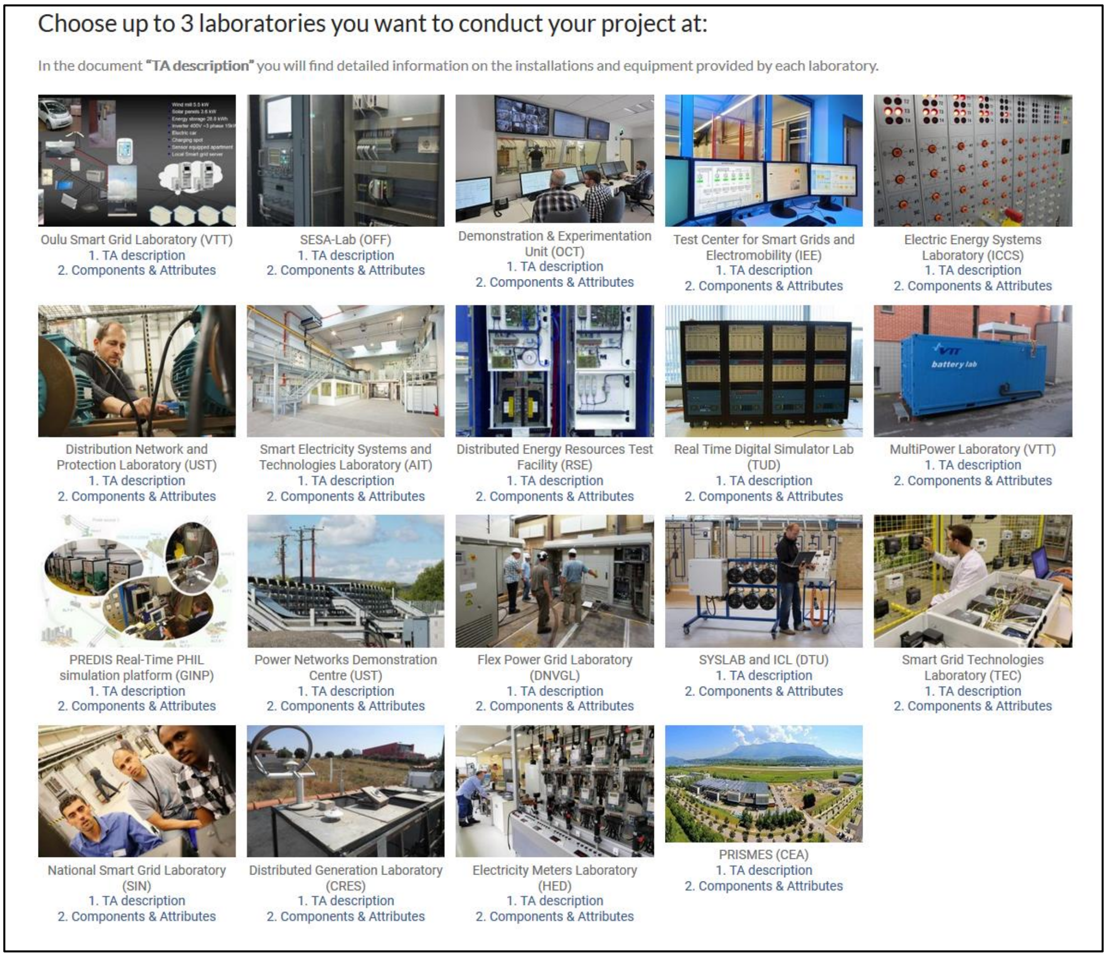
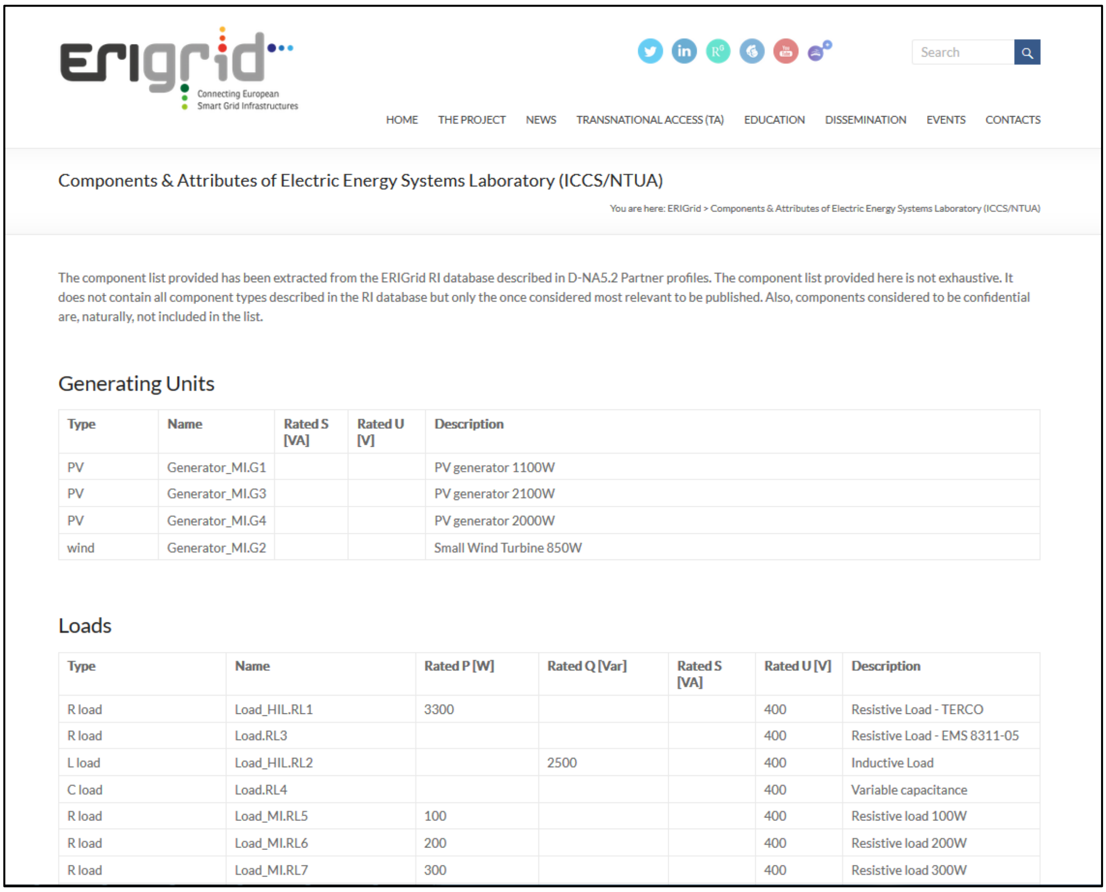


Figure 6.1. Additional links on the ERIGrid transnational access website

The additional links lead to new pages that include component information extracted from the database for each of the RIs. As an example, Figure 6.2represents the added page for Electric Energy Systems Laboratory (ICCS/NTUA).



Search

HOME THE PROJECT NEWS TRANSNATIONAL ACCESS (TA) EDUCATION DISSEMINATION EVENTS CONTACTS

Components & Attributes of Electric Energy Systems Laboratory (ICCS/NTUA)

You are here: ERIGrid > Components & Attributes of Electric Energy Systems Laboratory (ICCS/NTUA)

The component list provided has been extracted from the ERIGrid RI database described in D-NA5.2 Partner profiles. The component list provided here is not exhaustive. It does not contain all component types described in the RI database but only the once considered most relevant to be published. Also, components considered to be confidential are, naturally, not included in the list.

Generating Units

Type	Name	Rated S [VA]	Rated U [V]	Description
PV	Generator_MI.G1			PV generator 1100W
PV	Generator_MI.G3			PV generator 2100W
PV	Generator_MI.G4			PV generator 2000W
wind	Generator_MI.G2			Small Wind Turbine 850W

Loads

Type	Name	Rated P [W]	Rated Q [Var]	Rated S [VA]	Rated U [V]	Description
R load	Load_HIL.RL1	3300			400	Resistive Load - TERCO
R load	Load.RL3				400	Resistive Load - EMS 8311-05
L load	Load_HIL.RL2		2500		400	Inductive Load
C load	Load.RL4				400	Variable capacitance
R load	Load_MI.RL5	100			400	Resistive load 100W
R load	Load_MI.RL6	200			400	Resistive load 200W
R load	Load_MI.RL7	300			400	Resistive load 300W

Figure 6.2. Component information for Electric Energy Systems Laboratory (ICCS/NTUA)

7 Conclusions

This report outlines the ERIGrid approach for RI profiling. The RI profiles are stored in the ERIGrid RI profiles database, which has been designed, based on identified database usage requirements and is aligned with conventions and information models commonly used in ERIGrid. The document introduces a general data model for RI profiling and discusses the database implementation of ERIGrid. The database instance filled with ERIGrid partner RI data is an important part of the deliverable. This database is made publicly available on the ERIGrid website, containing the extracted public information of main components and their attributes for each ERIGrid laboratory.

The RI profiles database implemented in this deliverable is an important part of the holistic testing procedure introduced in [1] and further developed during the future work of ERIGrid [11]. The RI profiles can be utilized also in transnational access activities of ERIGrid to provide more detailed information on the RIs to the potential user groups. This is beneficial prior to submission of the user proposals to optimize the evaluation process whether a specific RI is suitable for a certain test and after acceptance of user proposals to facilitate preparatory work before the actual on-site tests. The public part of the database which is available on the ERIgrid website can be used by researchers who are interested to apply for transnational access provided by ERIGrid RIs as well as researchers who seek for a suitable RI in order to carry out specific experiments. The whole database can be made available for accepted TA users upon request.

The database structure is defined based on the system configuration definitions of [1] and borrows in many parts from CIM. This report includes both the general data structure definition (Section 3.2) and detailed data attributes for each database object (Annex 9.3). In addition, database implementation related things are discussed (Section 4.1). In ERIGrid, it is not required from all RIs to use the most detailed modelling possibilities of the database and minimum requirements for the data to be entered are documented (Section 4.2). If new cases for database use emerge or new technologies are taken into use, some parts of the database definition might need to be modified, new components added and/or data attributes updated. In addition, the minimum requirements for RI data can be completely different depending on the cases for database use.

8 References

[1] Deliverable D5.1, „D-NA5.1 Smart grid configuration validation scenario description method,“ H2020 ERIGrid project, 2017.

[2] „https://erigrid.eu/transnational-access/,“ [Online].

[3] Deliverable D9.1, „D-JRA3.1 Improved hardware and software interfaces,“ H2020 ERIGrid project, 2017.

[4] Deliverable D10.1, „D-JRA4.1 Report on JaNDER interconnections of the participating RIs,“ H2020 ERIGrid project, 2018.

[5] „https://erigrid.eu/dissemination/,“ [Online].

[6] „http://schemaspy.sourceforge.net/,“ [Online].

[7] Deliverable D7.1, „D-JRA1.1 ERIGrid scenario descriptions,“ H2020 ERIGrid project, 2016.

[8] Deliverable D3.1, „D-NA3.1 General rules for the ERIgrid trans-national access,“ H2020 ERIGrid project, 2016.

[9] Deliverable D7.3, „D-JRA1.3 Use case implementation plans,“ H2020 ERIGrid project, 2017.

[10] „https://www.postgresql.org/,“ [Online].

[11] Deliverable D5.3, „D-NA5.3 Harmonization, lessons learned and improvements report,“ H2020 ERIGrid project, 2020.

[12] NIST, „Special Publication 800-82, Guide to Industrial Control Systems Security,“ 2011.

[13] „http://infrastructure.der-lab.net/,“ [Online].

9 Annex

9.1 List of Figures

9.2 List of Tables

9.3 Detailed Data Attributes

The logical structure of the database is presented in Figure 3.3 and the implemented database in detail in the additional material on the ERIGrid website [5]. The detailed data attributes for each of the database tables are defined in this annex. The structure of the annex is based on the logical data structure of Figure 3.3.

Every entity in the database has a primary key and author information. These are not included in the tables below. The author information consists of two fields:

- author: The person who added or modified the data last (default value will be the database user)
- modifiedDate: The time when the data was modified last (default value the present time)

9.3.1 ResearchInfrastructure

The *ResearchInfrastructure* table gives the basic information on each RI. In this table, a set of resources that can be used together is represented as one RI. If a partner has separate RIs that cannot be connected to each other, they should be separated as individual RIs even if they are located at the same place.

Table 9.1: Data attributes in the ResearchInfrastructure table

Data attribute	Description	Unit	Mandatory data
name	Name of the RI.	-	X

Data attribute	Description	Unit	Mandatory data
organization	Owner of the RI.	-	X
descriptionPurpose	Verbal description of the purpose of the RI.	-	X
descriptionSetup	Verbal description of the physical setup of the RI.	-	X
voltageLevel	The RI voltage level. Acceptable values: {LV; MV; none}	-	X
securityChecks	Verbal description of the procedures needed to get access to the research infrastructure. A more detailed description will be provided in a separate document outside the database.	-	
externalConnectivity	Real-time connectivity of the RI with other RIs. Acceptable values: {None; JaNDER L0; JaNDER L1; JaNDER L2; Other}	-	X
confidential	If all RI information is confidential, set as true.	-	X
namingConvention	Description of the naming chosen hierarchy (interpretation aid)	-	X

The JaNDER interfaces referenced in *externalConnectivity* data attribute represent different ERIGrid implementations for real-time data exchange between RIs and will be explained in detail in [4].

9.3.2 Component

The *Component* table is the superclass for all laboratory components. The components can be of any type e.g., electrical components such as generators, control components such as voltage controllers or virtual components such as real-time simulators. The components are connected to each other through their terminals similarly as in CIM. One component can have terminals in several domains in which case the component acts as an interface between the different domains.

Table 9.2: Data attributes in the Component table

Data attribute	Description	Unit	Mandatory data
name	Name of the component	-	X
RI	Foreign key referring to the RI table.	-	X
description	Verbal description	-	X
confidential	If information related to this component is confidential, set as true.	-	X
isContainer	“Non-physical“components such as substations and bays can be modelled using plain Component objects i.e. without specializing subobjects	-	X

The subclasses of the *Component* class are presented below. The components are divided in six different categories. Some of the categories serve as superclasses in the DB logical structure. These superclasses are abstract classes, which means they cannot be instantiated by themselves in object or a database. The purpose of these classes is to store common attributes for all other classes, which are extending them. In the ERIGrid database implementation presented in Section 4.1 and the additional material on the ERIGrid website [5], the superclass attributes are included in the subclass tables and the superclasses (e.g., *DERComponent*) do not have dedicated tables in the DB.

The components should have a subclass that defines their attributes in a bit more detail. If a suitable subclass is not, however, available components can be modeled also using only the *Component* table and describing the modeled component in the description field.

9.3.2.1 DER Components

DERComponent is the superclass for the electrical domain energy resources (*GeneratingUnit, Load, StorageUnit, Converter, EVChargingStation, ElectricVehicle, Source, Amplifier*). The components can be real physical components or emulated (e.g., a PV *GeneratingUnit* can be a real PV panel or a PV emulator).

Table 9.3: Data attributes in the DERComponent table

Data attribute	Description	Unit	Mandatory data
physicalOrEmulated	Is the component a physical component or emulated? Acceptable values: {Physical; Emulated; Simulated}	-	X
ratedU	Nominal voltage	V	X
ratedP	Nominal real power	W	X
ratedQ	Nominal reactive power	VAr	X
ratedS	Nominal apparent power	VA	X
ratedf	Nominal frequency	Hz	X
phases	Phase number (1 or 3)	-	X

9.3.2.1.1 GeneratingUnit

Different types of generators are represented in the *GeneratingUnit* table, which is a subclass of the *DERComponent* table.

Table 9.4: Data attributes in the GeneratingUnit table

Data attribute	Description	Unit	Mandatory data
type	Type of the generating unit. Acceptable values: {Wind; PV; Internal Combustion Engine; Microturbine; Fuel cell; Other}	-	X
gridConn	Type of grid connection. Acceptable values: {Synchronous generator; Asynchronous generator; Inverter}	-	X
ratedEfficiency	Efficiency of the generating unit (needed especially for the fossil fueled generators)	pu	
gridFormingCapable	Is the generating unit grid-forming capable?	-	X

9.3.2.1.2 Load

Different types of loads are represented in the *Load* table, which is a subclass of the *DERComponent* table. The loads can be impedance loads, motors or inverter interfaced loads.

Table 9.5. Data attributes in the Load table.

Data attribute	Description	Unit	Mandatory data
type	Type of load. Acceptable values: {R load; L load; C load; RL load; RC load; LC load; Synchronous motor; Asynchronous motor; Inverter interfaced load}	-	X

Data attribute	Description	Unit	Mandatory data
continuousOrStepped	Can the load be controlled continuously? False = stepwise True = continuously	-	X
stepSizeP	Step size for real power load	W	
stepNmbrP	Number of steps for real power load	-	
stepSizeQ	Step size for reactive power load	VAr	
stepNmbrQ	Number of steps for reactive power load	-	

9.3.2.1.3 StorageUnit

Different types of electrical energy storages are represented in the *StorageUnit* table, which is a subclass of the *DERComponent* table.

Table 9.6: Data attributes in the StorageUnit table

Data attribute	Description	Unit	Mandatory data
type	Type of the storage unit. Acceptable values: {Battery; Li-ion battery; Ni-Cd battery; Lead acid battery; Redox flow battery; Flywheel; Ultracapacitor; Power to gas}	-	X
ratedCapacity	Nominal energy capacity	Wh	X
remainingCapacity	Current energy capacity. This field can be used to document the real storage capacity if it is known to be something different from the nominal value (e.g., old batteries).	Wh	
peakPCharge	Peak power for charging	W	
peakPDischarge	Peak power for discharging	W	
ratedEfficiency	Round trip efficiency of the storage unit	pu	

9.3.2.1.4 Converter

Converters that are not represented as a part of other components (e.g., *GeneratingUnit*) can be presented in the *Converter* table, which is a subclass of the *DERComponent* table.

Table 9.7: Data attributes in the Inverter table

Data attribute	Description	Unit	Mandatory data
PHILCapable	Can the inverter be used to form a closed-loop simulation?	-	X
convType	Type of the converter	-	X

9.3.2.1.5 EVChargingStation

The EV charging stations are represented in the *EVChargingStation* table, which is a subclass of the *DERComponent* table. If the EV charging station is operating according to IEC 61851 this should be defined in the *Standards* field.

Table 9.8: Data attributes in the EVChargingStation table

Data attribute	Description	Unit	Mandatory data
peakP	Peak power	W	X
V2G	V2G capable or not	-	X
Standards	IEC standards according to which the charging station operates (text field).	-	

9.3.2.1.6 ElectricVehicle

Electric vehicles are represented in the *ElectricVehicle* table, which is a subclass of the *DERComponent* table.

Table 9.9: Data attributes in the ElectricVehicle table

Data attribute	Description	Unit	Mandatory data
type	Type of the electric vehicle. Has to be some of the following {Full EV; PHEV}	-	X
ratedCapacity	Nominal energy capacity of the car batteries	Wh	X
peakPCharge	Peak power for charging	W	X
peakPDischarge	Peak power for discharging	W	X
V2G	V2G capable or not	-	X

9.3.2.1.7 Source

Different types of sources are modelled in the *Source* table which is a subclass of the *DERComponent* table. This table is used to model the main grid connection and e.g., DC sources. Also, grid emulators that can be used to decouple the laboratory network (or part of it) from the feeding network so that voltage and frequency are controllable are represented in the *Source* table.

Table 9.10: Data attributes in the Source table

Data attribute	Description	Unit	Mandatory data
type	Type of the source. Acceptable values: {Main grid; Linear; Nonlinear}	-	X
ACDC	Is the source AC or DC? Acceptable values: {AC; DC}	-	X
PHILCapable	Can the source operate as a part of a closed-loop simulation?	-	X
minU	Minimum voltage	V	
maxU	Maximum voltage	V	
minf	Minimum frequency	Hz	
maxf	Maximum frequency	Hz	

9.3.2.1.8 Amplifier

Amplifiers necessary e.g., in many HIL arrangements are represented in the *Amplifier* table.

Table 9.11: Data attributes in the Amplifier table

Data attribute	Description	Unit	Mandatory data
IONmbr	Number of channels	-	X
AmplifierMode	Acceptable values: {Switched Mode; Synchronous Generator; Linear}	-	X
InputBoundary	Limits on inputs		X
OutputBoundary	idem		X
GalvanicIsolation	(Y/N)		
SlewRate	Maximum rate at which the amplifier can respond to an abrupt change of input level.		
IdentifiedTransferFunc-tion	Is the transfer function identified? (Y/N). If yes, please provide the transfer function in description field.	-	
InterfaceType	Current type or VoltageType.	-	
InterfaceAlgorithm	Acceptable values: {ITM (Ideal Transformer Model); PCD (Partial Circuit Duplication); DIM (Damping Impedance Method); Other}	-	
realTimeSimulator	Foreign key referring to the RealTimeSimulator table. If the amplifier is associated with a real-time simulator, the simu-lator is indicated in this attribute. If not, the field is left empty.	-	

9.3.2.2 Thermal Components

ThermalComponent is the superclass for the thermal domain energy resources (*ThermalSource, ThermalLoad, ThermalStorage*). The components can be real physical or emulated/simulated (e.g., a boiler can be a real boiler or a boiler emulation/simulation).

Table 9.12: Data attributes in the ThermalComponent table

Data attribute	Description	Unit	Mandatory data
physicalOrEmulated	Is the component a physical component or emulated/simulated? Acceptable values: {Physical; Emulated, Simulated}	-	X
DERComponent	Foreign key referring to DERComponent table (in particular for CHP and heat pumps)	-	
ratedPth	Nominal Thermal Power	W	X
MinPth	Minimum Thermal Power	W	
ratedTemp	Heated fluid nominal temperature	°C	
ratedEfficiency	Efficiency of the ThermalComponent at nominal power(for heat pumps indicate COPheating/cooling, for storage round trip effi-ciency)	pu	
HeatFluid	Type of heated fluid. Acceptable values: {Water; Steam; Air; Oil; Other}	-	
continuousOrStepped	Can the ThermalComponent be controlled continuously or step-wise?	-	
stepSizePth	Step size for thermal power (if stepped)	W	

9.3.2.2.1 ThermalSource

Different types of heaters are represented in the *ThermalSource* table, which is a subclass of the

ThermalComponent table.

Table 9.13: Data attributes in the ThermalSource table

Data attribute	Description	Unit	Mandatory data
type	Type of the heating/cooling unit. Acceptable values: {Boiler, CHP, Heat pump, Solar system, others}	-	X
Primary Energy	Type of primary energy/fuel. Acceptable values: {Electricity; Natural gas; Oil; Solar; Other)	-	

9.3.2.2.2 ThermalLoad

Different types of loads are represented in the *ThermalLoad* table, which is a subclass of the *ThermalComponent* table. The loads can be domestic heating/cooling circuits, industrial loads, heat sinks (as thermal load emulators) etc.

Table 9.14: Data attributes in the ThermalLoad table

Data attribute	Description	Unit	Mandatory data
type	Type of Thermal load. Acceptable values: {Domestic heating; Industrial load; Heat sink; Other)	-	X

9.3.2.2.3 ThermalStorage

Different types of energy storages are represented in the *ThermalStorage* table, which is a sub-class of the *ThermalComponent* table.

Table 9.15. Data attributes in the ThermalStorage table.

Data attribute	Description	Unit	Mandatory data
type	Type of the storage unit. Acceptable values: {Tank; Phase change storage; Chemical storage; Other}	-	X
ratedCapacity	Nominal energy capacity	Wh	X
peakPCharge	Peak power for charging	W	
peakPDischarge	Peak power for discharging	W	
avDiss	Average storage heat losses: watts per temperature difference (Tfluid-Tamb)	W/(°C)	

9.3.2.3 Connecting Components

Components that connect components together have been grouped as one category. The following components are included in this category: *ElectricalLineSegment, Transformer, ElectricalSwitch, CommunicationLineSegment, CommunicationInterchange, HeatExchanger, ThermalPipeline, ThermalCircuitEquipment*. These components do not have any common attributes and, therefore, the categorization is not visible in the database structure as a superclass.

9.3.2.3.1 ElectricalLineSegment

Electrical line segments are modelled in the *ElectricalLineSegment* table. If the line impedances can be regarded as negligible and there is no need to model the electrical network in detail, *ElectricalLineSegment* table can remain empty.

Table 9.16: Data attributes in the ElectricalLineSegment table

Data attribute	Description	Unit	Mandatory data
length	Length of the line segment	m	X
phases	Phase number (1 or 3)	-	X
resistance	Line resistance	Ω	X
reactance	Line reactance	Ω	X
ratedCurrent	Rated current of the line segment	A	X

9.3.2.3.2 Transformer

Transformer data is represented in the *Transformer* table.

Table 9.17: Data attributes in the Transformer table

Data attribute	Description	Unit	Mandatory data
ratedU1	Primary side nominal voltage	V	X
ratedU2	Secondary side nominal voltage	V	X
ratedS	Nominal apparent power	VA	X
Xk	Leakage reactance	%	
Rk	Winding resistance	%	
tapChanger	Is there a tap changer in the transformer?	-	X
stepSize	Tap changer step size	pu	
tapNmbrDown	Number of tap steps downwards (decreasing secondary side voltage)	-	
tapNmbrUp	Number of tap steps upwards (increasing secondary side voltage)	-	

9.3.2.3.3 ElectricalSwitch

Switches can be modelled in the *ElectricalSwitch* table if needed.

Table 9.18: Data attributes in the ElectricalSwitch table

Data attribute	Description	Unit	Mandatory data
phases	Phase number (1 or 3)	-	X
ratedCurrent	Rated current of the switch	A	
statusMonitor	Can the status of the switch be remotely monitored? {TRUE or FALSE}	-	

9.3.2.3.4 CommunicationLineSegment

Communication line segments are modelled in the *CommunicationLineSegment* table. If communication can be assumed to be very fast and reliable and there is no need to model the communication network in detail, *CommunicationLineSegment* table can remain empty. If the RI foresees specific testing on communication, modelling communication line segments is mandatory.

Table 9.19: Data attributes in the CommunicationLineSegment table

Data attribute	Description	Unit	Mandatory data
length	Length of the line segment	m	X
medium	The medium of the communication line.	-	X
bandwidth	Bandwidth of the communication line as defined by the application.	bit/s	X

9.3.2.3.5 CommunicationInterchange

CommunicationInterchange table covers all communication devices, which facilitate communication by connecting several communication links, without being an endpoint. This includes hubs, switches, routers, but also e.g., media converters or gateways, which translate between domains. If the RI foresees specific testing on communication, modelling communication interchange components is mandatory.

Table 9.20: Data attributes in the CommunicationInterchange table

Data attribute	**Description**	**Unit**	**Mandatory data**
type	The type of the component. Has to be some of the following: {Hub; Switch; Router; Gateway; Media converter; Other}	-	X
primaryDomain	Foreign key referring to Domain table. If the device interconnects communication links of the same type (e.g., an Ethernet switch or a USB hub), the primary domain is the only one that exists. If the device converts between domains (e.g., a webservices to Modbus/RTU gateway), the primary domain will be on the upstream side if an upstream side can be identified (e.g., towards the master in a master-slave system). If the device converts between domains and no upstream side can be identified (e.g., CAN bus extender over Ethernet), primary and secondary can be assigned at random.	-	X
secondaryDomain	Foreign key referring to Domain table. Empty if the device interconnects communication links of the same type, otherwise the domain of the upstream link if upstream can be identified (see under primaryDomain), otherwise the domain of the link not entered as primary.	-	X

9.3.2.3.6 HeatExchanger

Heat exchanger data is represented in the *HeatExchanger* table.

Table 9.21: Data attributes in the HeatExchanger table

Data attribute	Description	Unit	Mandatory data
type	Type of the Heat exchanger. Acceptable values: {Tube/shell; Tube in tube; Plate; heat pipes; Other}	-	
ratedPth	Nominal Thermal Power	W	X
HeatingFluid	Type of heating fluid (water, steam, air, oil, other) -inlet	-	
HeatedFluid	Type of heated fluid (water, steam, air, oil, other) -outlet	-	
ratedEfficiency	Heat exchanger efficiency at nominal thermal power	pu	
MaxTempIn	Heating fluid max temperature	°C	
MaxTempOut	Heated fluid max temperature	°C	

9.3.2.3.7 ThermalPipeline

As a detailed description of each pipelines segment is out of scope of the DB, only overall information are requested; a simplified scheme of the overall thermal circuit with different line segments should be included in the *DomainTopology* table. Thermal pipeline data is represented in the *ThermalPipeline* table.

Table 9.22: Data attributes in the ThermalPipeline table

Data attribute	Description	Unit	Mandatory data
type	Type of the Thermal Pipeline. Acceptable values: {Not insulated; Insulated; Thermal heated}	-	
nomD	Nominal Thermal Pipeline Diameter (average value)	m	
length	Overall length of the Thermal pipeline	m	X
nomT	Nominal fluid temperature	°C	
HeatFluid	Type of heat fluid. Acceptable values: {Water; Steam; Air; Oil; Other}	-	
avDiss	Average pipeline heat losses: watts per meter length and temperature difference (Tfluid-Tamb)	W/(m °C)	
nomFR	Nominal Flow rate	m3/s	
nomDP	Overall pressure losses at Nominal Flow rate	bar	
DHconnection	Stand-alone Thermal circuit or part of a thermal network connected to District Heating	0/1	X
DH_Pth_in	If DHconnection=1, Max inlet thermal power	W	
DH_Pth_out	If DHconnection=1, Max outlet thermal power	W	
DH_T_in	If DHconnection=1, Nominal inlet temperature	°C	
DH_T_out	If DHconnection=1, Nominal outlet temperature	°C	
DH_T_FR	If DHconnection=1, Nominal inlet/outlet Flow rate	m3/s	

9.3.2.3.8 ThermalCircuitEquipment

As a detailed description of each heat circuit equipment installed in the thermal circuit (in particular pumps and valves) is out of scope of the DB, only general information are requested; a simplified scheme of the overall thermal circuit with the installed main equipment should be included in the *DomainTopology* table. Thermal circuit equipment is represented in the *ThermalCircuitEquipment* table.

Table 9.23: Data attributes in the ThermalCircuitEquipment table

Data attribute	Description	Unit	Mandatory data
type	Type of the Heat Equipment. Acceptable values: {Pump; Valve; Other}	-	X
DERcomponent	Foreign key referring to DERComponent table (in particular for pumps as electrical loads)	-	
ratedFlow	Nominal Flow rate	m3/s	
continuousOrStepped	Can variable be controlled continuously or stepwise?	-	

9.3.2.4 Control and Monitoring Components

Control and monitoring components are represented as one component category. The following components are included in this category: *ControlDevice, ProtectionDevice, MeasurementDevice, LabSCADA*. These components do not have any common attributes and, therefore, the categorization is not visible in the database structure as a superclass.

9.3.2.4.1 ControlDevice

Control devices are represented in the *ControlDevice* table. Controllers operating on different levels can be modelled. The different levels are defined in [3] and are introduced in Section 3.1.2. Section 3.1.2 also describes what a control device in this deliverable. The *ControlDevice* table is used to model also purely manual control of for instance loads or switches.

One control device can have several control modes and can host third-party software on many levels. The control modes are represented in a separate *ControlModes* table and the hosting capacities in a separate *ControllerHostingCapacity* table.

The *ControlDevice* table provides a possibility to input control hierarchy and communication protocol data as lists in three text fields (*upperlevelControlDevice, lowerlevelControlDevice, horizontal-levelControlDevice*). Another, more comprehensive, option to model the control hierarchies is through the terminal-connectivity node representation.

Table 9.24: Data attributes in the ControlDevice table

Data attribute	Description	Unit	Mandatory data
modeltype	Modeltype of the controller. In case of manual control this should be set to “Manual”.	-	X
primaryComponent	Foreign key referring to Component table. The controller can be linked to a primary component if desired. The primary component means the physical component which the controller is controlling. This attribute is used only at the lowest levels of the control hierarchy (D3-D5) and refers to one primary component only. It can be used, for example, to associate a generator voltage controller to the generating unit.	-	
upperlevelControlDevice	If there is a hierarchical control structure, the upper level control devices and protocols that are used to communicate with them are listed here. This is a text field and the format to input data should be as follows:{ControlDevice 1, protocol 1; ControlDevice 2, protocol 2; …; ControlDevice n, protocol n}	-	

Data attribute	Description	Unit	Mandatory data
lowerlevelCon-trolDevice	If there is a hierarchical control structure, the lower level control devices and protocols that are used to communicate with them are listed here. This is a text field and the format to input data should be as fol-lows:{ControlDevice 1, protocol 1; ControlDevice 2, protocol 2; …; ControlDevice n, protocol n}	-	
horizontallevelCon-trolDevice	If there is a hierarchical control structure, the control devices at the same control level and protocols that are used to communi-cate with them are listed here. This is a text field and the format to input data should be as fol-lows:{ControlDevice 1, protocol 1; ControlDevice 2, protocol 2; …; ControlDevice n, protocol n}	-	

9.3.2.4.1.1 ControlModes

The *ControlModes* table lists all possible control modes. The control modes are linked to control devices and each control device can have multiple possible control modes.

Table 9.25: Data attributes in the ControlModes table

Data attribute	Description	Unit	Mandatory data
controlMode	Control mode	-	X
controlDevice	Foreign key to ControlDevice table.	-	X
controlLevel	The level on which the control operates. The levels are based on the definitions in [3] and discussed in Section 3.1.2. Acceptable values: {D1; D2; D3; D4; D5}	-	X
description	Verbal description.	-	X

9.3.2.4.1.2 ControllerHostingCapacity

The *ControllerHostingCapacity* table can be used to model where and how third-party software can be used. The controller hosting capacities are linked to control devices and each control device can have multiple hosting capacity relations.

Table 9.26: Data attributes in the ControllerHostingCapacity table

Data attribute	Description	Unit	Mandatory data
type	Type of the hosting capacity, e.g., “Java code”, “Labview” etc.	-	X
controlDevice	Foreign key to ControlDevice table.		X
controlLevel	The level on which the control operates. Acceptable values: {D1; D2; D3; D4; D5}	-	X
description	Verbal description.	-	X

9.3.2.4.1.3 LabSCADA

The central LabSCADA system is represented in the *LabSCADA* table. *LabSCADA* is a subclass of the *ControlDevice* class.

Table 9.27: Data attributes in the LabSCADA table

Data attribute	Description	Unit	Mandatory data
type	The type of the SCADA system. Has to be some of the following {DCS; SCADA}	-	X
Capacity	Maximum Number of nodes	-	X
employedStandard	Interface protocol to applications of same level (e.g., OPC). The characteristics of the protocol can also be described.	-	X
DataLoggingFacilities	Text field to explain the data logging facilities of the SCADA.	-	X
HMI	Description field.	-	X
Platform	Platform/language on which the SCADA was built (e.g., Lynx, PCVue, etc).	-	X

9.3.2.4.2 ProtectionDevice

Protection devices are represented in the *ProtectionDevice* table. The protection devices relevant for ERIGrid database are the ones that can be used as a part of a test setup. The protection devices that are protecting the lab components as an inseparable part of them do not need to be modelled in the database unless some completely new cases for using the database appear in addition to those identified in Section 2. Different protection functionalities are represented in the separate *ProtectionModes* table.

Table 9.28: Data attributes in the ProtectionDevice table

Data attribute	Description	Unit	Mandatory data
modeltype	Modeltype of the protection device.	-	X
primaryComponent	The protection device can be linked to a primary component if desired. The primary component is the breaker/switch, which the protection device is controlling.	-	
intProtocol	Interface protocol of the unit.	-	

9.3.2.4.2.1 ProtectionModes

The *ProtectionModes* table lists all possible protection modes. The protection modes are linked to protection devices and each component can have multiple possible protection modes.

Table 9.29: Data attributes in the ProtectionModes table

Data attribute	Description	Unit	Mandatory data
protectionMode	Protection mode	-	X
protectionDevice	Foreign key to ProtectionDevice table.		X
description	Verbal description.	-	X

9.3.2.4.3 MeasurementDevice

Measurement devices are represented in the *MeasurementDevice* table. Different measurement types are represented in the separate *MeasuredData* table.

Table 9.30: Data attributes in the MeasurementDevice table

Data attribute	Description	Unit	Mandatory data
type	Type of the measurement device. Acceptable values: {Remote terminal unit; Smart meter; PMU; Other measurement device}	-	X
dataSampling	Sampling interval of data.	s	X
dataHandling	Type of data. Acceptable values: {Instantaneous value; RMS}	-	X
upstreamProtocol	Interface protocol of the unit.	-	X

9.3.2.4.3.1 MeasuredData

The *MeasuredData* table lists all possible measurable data. The measured data are linked to measurement devices and each component can have multiple possible measured variables.

Table 9.31: Data attributes in the MeasuredData table

Data attribute	Description	Unit	Mandatory data
measuredData	Measured variable	-	X
measurementDevice	Foreign key to MeasurementDevice table	-	X
dataQuality	Data quality	-	X
unit	Unit of the measured data	-	X
description	Verbal description	-	X

9.3.2.5 Cybersecurity Control Components

CyberSecurityControl is the superclass for components related to cyber security. The components in this category represent security mechanisms that are implemented in an RI and include the following components: *SecureCommunication, Firewall, IntrusionDetectionSystem, Antivirus, NetworkSegmentation, AccessControls*.

Table 9.32: Data attributes in the CyberSecurityControl table

Data attribute	Description	Unit	Mandatory data
threatDefence	Description of threats for which this security control has been deployed.	-	X
version	Version of the software of hardware security control.	-	

9.3.2.5.1 SecureCommunication

Normally communication between components and devices in the testbed facilities occurs using dedicated communication protocols whereas the information is exchanged in plaintext. Confidential or sensitive information that is transmitted with these protocols can easily be intercepted, read and modified unless the information is not protected. The class *SecureCommunication* shall be used for describing authentication and encryption technologies used in the testbeds. Secure communication protocols provide a way to authenticate clients and servers (or simply to entities/components in the testbed) to protect confidentiality and integrity of communication. A variety of secure communication standards have been developed, including IP security (IPsec), SSL and TLS. This class allows user to specify which protocol is used between to two end-points and describe encryption scheme (and/or authentication schemes) which are used.

Table 9.33: Data attributes in the SecureCommunication table

Data attribute	Description	Unit	Mandatory data
protocolType	Description of protocol type which is used for communication between two entities in the testbed, e.g., Modbus/TCP, IEC 61850, etc.	-	X
ComponentServer	First end-point (server) which is involved in data exchange the secure protocol. Foreign key referring to a component in the testbed, e.g., ServerPC (defined in ICT Components).	-	X
ComponentClient	Second end-point (client) which is involved in data exchange over secure protocol. Foreign key referring to a component in the testbed, e.g., ControlDevice.	-	X
secureProtocol	Type of the secure protocol used for data exchange. Acceptable values: {SSL; IPsec; VPN; Other}	-	X
encryptionScheme	Description of encryption scheme used for exchange of data.	-	
authentication	Specify whether a secure protocol involves authentication Acceptable values: {No; Client Authentication; Server Authentication; Mutual Authentication}	-	
authenticationKey	Description of keys which are used for authentication, e.g., password, digital certificates, etc.	-	

9.3.2.5.2 Firewall

Network firewalls are devices or systems that control the flow of network traffic between networks employing different security postures. By employing firewalls to control connectivity to different network areas, an organization can prevent unauthorized access to the representative systems and resources within the more sensitive network. There are three general classes of firewalls [12]:

- **Packet Filtering Firewalls** – the most basic type of firewall that is essentially routing devices that include access control functionality for system addresses and communication sessions.
- **Stateful Inspection Firewalls** – stateful inspection firewalls are packet filters that beside basic packet filtering also incorporates added awareness of the data in transport layer, e.g., they can determine whether session packets are legitimate and evaluate content of packets at the transport layer (e.g., TCP or UDP).
- **Application-Proxy gateway firewalls** – this class of firewalls examines packets at the application layer and filters traffic based on specific application rules.

Table 9.34: Data attributes in the Firewall table

Data attribute	Description	Unit	Mandatory data
productName	Name of the firewall which is installed in the testbed.	-	X
type	The type of the firewall which is used in the facility. Acceptable values: {Packet filtering firewall; Stateful Inspection Firewall; Application-Proxy gateway firewall}	-	X
integratedSecurityServices	Description of other security services which are integrated within the firewall, e.g., intrusion detection and prevention, virtual private network, malware protection, etc.	-	

9.3.2.5.3 IntrusionDetectionSystem

An intrusion detection system (IDS) is a device or software application that monitors network or systems for malicious activity or policy violations. Any detected activity or violation is typically reported to administrator. In this class, users can specify whether type of IDS is network or host based. A system that monitors important operating files is an example of a host-based IDS, while a system that analyses incoming traffic is an example of a network-based IDS. Moreover, the data attribute *componentReference* allows a user to specify other components of the testbed on which this IDS is installed.

Table 9.35: Data attributes in the IntrusionDetectionSystem table

Data attribute	Description	Unit	Mandatory data
productName	Name of the intrusion detection system which is installed in the testbed.	-	X
type	The type of the intrusion detection system which is used in the facility. Acceptable values: {Host-based IDS; Network-based IDS}	-	X
componentReference	Foreign key referring to the Component table. The reference to the component in the testbed on which the IDS is installed on, e.g., firewall, router, server PC, etc.	-	

9.3.2.5.4 Antivirus

The *Antivirus* class can be used to describe antivirus software used in the testbed facility. Antivirus is a software component used to prevent, detect and remove malicious software. Besides the *productName*, this class should also reference the testbed component to which the antivirus software is installed on using *componentReference* attribute.

Table 9.36: Data attributes in the Antivirus table

Data attribute	Description	Unit	Mandatory data
productName	Name of the anti-virus product which is installed in the testbed.	-	X
componentReference	Foreign key referring to the Component table. The reference to the component in the testbed on which the antivirus software is installed.	-	X

9.3.2.5.5 NetworkSegmentation

Partitioning the network into security domains and separating the operational network from other networks is a common security control used in many network designs for critical infrastructures. The aim of network segmentation is to minimize access to sensitive information for those systems that do not need it. This can be achieved using a number of techniques and technologies depending on the networks' architecture and configuration. The *NetworkSegmentation* class is used to describe different technologies used in the testbed to implement network segmentation and describe their usage. For example, firewall based separation could be used to segment corporate network and control network. Only network segmentation implemented for security reasons should be described in this table. For example, VLANs implemented for performance reasons should not qualify as a security measure.

Table 9.37: Data attributes in the NetworkSegmentation table

Data attribute	Description	Unit	Mandatory data
typesOfNetworkSegmentation	The type of network segmentation which is used to isolate the testbed networks of different sensitivity. Acceptable values: {Physical isolation; Logical isolation (VLAN); Unidirectional gateways; Firewall based separation between different network zones}.	-	X

For each chosen value the usage of network segmentation type within the research facility must be described in the description field inherited from *Component* superclass.

9.3.2.5.6 AccessControls

Access controls are concern with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system. A given IT infrastructure of the ERIGrid testbed facility can implement access control systems in many places and at different levels. For example, operating systems and digital electronic devices can use access control systems to protect files and directories. The *AccessControls* class describes access controls implemented within the testbed facility, including also physical access control.

Table 9.38: Data attributes in the AccessControls table

Data attribute	Description	Unit	Mandatory data
typeOfAccessControl	The type of access control which are being used in the system or on the specific component. Acceptable values: {Access Control List; Role-based access control (RBAC); Rule-based access control (RAC); Physical access; Other}	-	X
identificationUsed	Description of identifications used to authenticate the user to the access control system, e.g., user names/passwords, digital certificates, ID cards, personal card, etc.	-	X
componentRef	Optional foreign key referring to the Component table. A component which implement access controls can be referenced, e.g., a firewall component.	-	

9.3.2.6 ICT Components

ICT components are represented as one component category. The following components are included in this category: *SimulationCapacity, NetworkImpairmentSimulator, StorageAreaNetwork, VirtualServerEnvironment, ServerPC, RealTimeSimulator, Service*. These components do not have any common attributes and, therefore, the categorization is not visible in the database structure as a superclass.

9.3.2.6.1 SimulationCapacity

Servers, virtual machines etc. are represented in the *SimulationCapacity* table.

Table 9.39: Data attributes in the SimulationCapacity table

Data attribute	Description	Unit	Mandatory data
RAM	Size of RAM	Byte	X
CPU	Size of CPU	Hz	X
singleServer	Is it a single server or integrated in a cluster	-	X
LoadBalancing	Is it possible to balance the load over more than one server or via the usage of virtual machines?(true or false)	-	X

9.3.2.6.2 NetworkImpairmentSimulator

A network impairment simulator can be used for testing the network performance and connection under realistic network conditions (e.g., GSM, LTE profile).

Table 9.40: Data attributes in the NetworkImpairmentSimulator table

Data attribute	Description	Unit	Mandatory data
ports	Number of Ethernet ports	-	X
profiles	Description of available profiles	-	X
timeAccuracy	Temporal accuracy	s	X
sizeSimult	Amount of simultaneous emulations	-	X
accessRights	User groups, roles and rights management	-	X

9.3.2.6.3 StorageAreaNetwork

Central storage network for servers with high availability and failure safety is modelled in the *StorageAreaNetwork* table.

Table 9.41: Data attributes in the StorageAreaNetwork table

Data attribute	Description	Unit	Mandatory data
hdd	Size of HDD	Byte	X
connectionTech	Connection technology for servers (e.g., Fibre Channel)	-	
serverCount	Count of servers that can be connected	-	X
redundancy	Information about redundancy of the SAN	-	

9.3.2.6.4 VirtualServerEnvironment

Environment for server operation as virtual machines is modelled in the *VirtualServerEnvironment* table.

Table 9.42: Data attributes in the VirtualServerEnvironment table

Data attribute	Description	Unit	Mandatory data
cpu	CPU capacity of the cluster	Hz	X
ram	RAM capacity of the cluster	Byte	X
hdd	HDD space of the cluster	Byte	X
resourcePools	Management of available resources in pools (for a couple of virtual machines)(true or false)	-	
highAvailability	Information about failover in case of faulty hosts	-	
accessRights	User groups, roles and rights management	-	

9.3.2.6.5 ServerPC

The *ServerPC* table is used to model server/PC for simulations or other software components.

Table 9.43: Data attributes in the ServerPC table

Data attribute	Description	Unit	Mandatory data
isVirtual	Information if server is hardware or virtual (true or false)	-	X
cpu	CPU capacity	Hz	X

Data attribute	Description	Unit	Mandatory data
ram	Size of RAM	Byte	X
hdd	Hard drive space	Byte	X
os	Operating system	-	X
software	Installed software	-	X
ip	IP parameters	-	
accessRights	User groups, roles and rights management	-	

9.3.2.6.6 RealTimeSimulator

Real-time simulators such as the RTDS or OPAL-RT are represented in the *RealTimeSimulator* table.

Table 9.44: Data attributes in the RealTimeSimulator table

Data attribute	Description	Unit	Mandatory data
maxNodeNmbr	Maximum number of network nodes that can be modelled	-	X
analogInputNmbr	Number of analog input ports	-	X
digitalInputNmbr	Number of digital input ports	-	X
analogOutputNmbr	Number of analog output ports	-	X
digitalOutputNmbr	Number of digital output ports	-	X

9.3.2.6.7 Service

Services such as weather forecast available at the RI are represented in the *Service* table.

Table 9.45: Data attributes in the Service table

Data attribute	Description	Unit	Mandatory data
type	Type of the Service, e.g., "Weather Forecast", "Generation Fore-cast", "Load Forecast", "Market price", "State Estimation", etc. Detailed verbal description for the service (including data, time window etc.) should be given in the description data attribute.	-	X

9.3.3 Terminal

In the ERIGrid system configuration language, as in CIM, components are connected through terminals and connectivity nodes. The terminals are represented in the *Terminal* table. In contrast to CIM, this connectivity notion is extended to IT and other physical domains. It is mandatory to fill all data attributes if a CIM compatible description is adopted.

Table 9.46: Data attributes in the Terminal table

Data attribute	Description	Unit	Mandatory data
name	Name of the terminal	-	X
component	Foreign key referring to the Component table	-	X
domain	Foreign key referring to the Domain table	-	X

Data attribute	Description	Unit	Mandatory data
connectivityNode	Foreign key referring to the ConnectivityNode table.	-	
type-D	Direction type of terminal: {bidirectional; outwards; inwards; acausal} *outward, inward, bidirectional* : only one flow direction, or it can be controlled *acausal* : flow direction follows physical laws / depends on con-nectect systems e.g., electrical AC connections	-	X
description	Verbal description. (optional)	-	
type-E	Connector/endpoint type; may be specific to domain (optional); e.g., *electrical*: {not-grounded; grounded}; *communication*: {mas-ter; slave}; {client; server}	-	X

9.3.4 Domain

The *Domain* table lists all possible domains and enables also domain hierarchy modelling.

Consistent modelling of domains is aimed for in ERIGrid and, therefore, only a restricted group of DB users will have rights to modify the *Domain* table. Domain categorization work has started already in [1] and the domains used in the database will be derived from that previous work. Modifications will be made during the data filling process and also throughout the whole project when a need to model new domains arises.

Table 9.47: Data attributes in the Domain table

Data attribute	Description	Unit	Mandatory data
domain	Domain name	-	X
parentDomain	Foreign key referring to the Domain table. Refers to Domain's parent in domain hierarchy	-	
description	short description to facilitate the interpretation of domains	-	X

9.3.5 Connectivity

There are two options to represent the connectivity in a lab: utilizing connectivity nodes or using a graphical representation.

9.3.5.1 ConnectivityNode

Connectivity nodes are needed in the detailed topology modelling approach that has been developed based on CIM. The original CIM models only electrical domain networks but in ERIGrid the concept is extended also to other domains. It should be noted that all the terminals connected to one connectivity node have to be at the same domain. It is mandatory to fill all data attributes if a CIM compatible description is adopted.

Table 9.48: Data attributes in the ConnectivityNode table

Data attribute	Description	Unit	Mandatory data
name	Name of the connectivity node	-	X
RI	Foreign key referring to the ResearchInfrastructure table.	-	X

9.3.5.2 DomainTopology

The DomainTopology table is used to present graphically the network topologies in different domains (e.g., electrical, ICT, district heating). The database includes also another way of presenting the topology through terminals and connectivity nodes. In many cases, the graphical representation is, however, adequate and also easier to interpret for a human user. Each row in the DomainTopology table i.e., topology figure is related to a laboratory and to a domain as shown in Figure 3.3. Graphical representations are required even if also the connectivity node based model is utilized. The picture format has to be PNG.

Table 9.49: Data attributes in the DomainTopology table

Data attribute	Description	Unit	Mandatory data
picture	Picture of the topology.	-	X
RI	Foreign key referring to the ResearchInfrastructure table.	-	X
domain	Foreign key referring to the Domain table.	-	X
description	Verbal description.	-	X
confidential	If the topology picture is confidential, set as true.	-	X

For electrical networks, the network picture should preferably be a single-line diagram in which network parts that cannot be directly interconnected are highlighted e.g., by using different colors (e.g., transition from a 3ph to a 1ph sub-grid, transition from an AC to a DC sub-grid, different voltage levels). In the network diagram, the components must be labeled and component names must be consistent with the data entered into the database. In addition, the most significant measurement locations should be indicated in the diagram. At least all measurement devices listed in the DB have to be marked in the diagram and, preferably, all measurement locations should be indicated.

For thermal networks, the main components must be clearly indicated and component names must be consistent with the data entered into the database. A thermal circuit P&ID is the preferred solution.

For communication networks, the picture should include the main communication lines and devices (as hubs, routers, gateways), the connection to main components and the adopted protocols. For communication networks, the picture should be a graphical representation of control devices (with implemented control modes). Control relations must be clearly defined regarding hierarchy direction (downstream/upstream/horizontal) and associated communication protocols. Component names must be consistent with the data entered into the database.

9.3.6 ModelFile

It is possible to store also binary files related to an RI in the database. If an RI has, for instance, a CIM representation of its facility, this CIM file can be stored in the *LabModelFile* table. All data attributes are mandatory if such a file is included.

Table 9.50: Data attributes in the ModelFile table

Data attribute	Description	Unit	Mandatory data
file	Binary file.	-	X
RI	Foreign key referring to the ResearchInfrastructure table.	-	X
description	Verbal description.	-	X
confidential	If the file is confidential, set as true.	-	X