energies

MDPI

Article

Upgrading BRICKS—The Context-Aware Semantic Rule-Based System for Intelligent Building Energy and Security Management

Gabriel Santos [1,2], Tiago Pinto [1,2,*], Zita Vale [2], Rui Carvalho [1,2], Brígida Teixeira [1,2] and Carlos Ramos [1,2]

[1] GECAD Research Group, 4249-015 Porto, Portugal; gajls@isep.ipp.pt (G.S.); rugco@isep.ipp.pt (R.C.); bccta@isep.ipp.pt (B.T.); csr@isep.ipp.pt (C.R.)
[2] Institute of Engineering-Polytechnic of Porto, 4249-015 Porto, Portugal; zav@isep.ipp.pt
* Correspondence: tcp@isep.ipp.pt; Tel.: +351-228-340-511

Abstract: Building management systems (BMSs) are being implemented broadly by industries in recent decades. However, BMSs focus on specific domains, and when installed on the same building, they lack interoperability to work on a centralized user interface. On the other hand, BMSs interoperability allows the implementation of complex rules based on multi-domain contexts. The *Building's Reasoning for Intelligent Control Knowledge-based System* (BRICKS) is a context-aware semantic rule-based system for the intelligent management of buildings' energy and security. It uses ontologies and semantic web technologies to interact with different domains, taking advantage of cross-domain knowledge to apply context-based rules. This work upgrades the previously presented version of BRICKS by including services for energy consumption and generation forecast, demand response, a configuration user interface (UI), and a dynamic building monitoring and management UI. The case study demonstrates BRICKS deployed at different aggregation levels in the authors' laboratory building, managing a demand response event and interacting autonomously with other BRICKS instances. The results validate the correct functioning of the proposed tool, which contributes to the flexibility, efficiency, and security of building energy systems.

Keywords: context-aware knowledge-base systems; intelligent control; interoperability; semantic reasoning; semantic rule-based systems

check for updates

Citation: Santos, G.; Pinto, T.; Vale, Z.; Carvalho, R.; Teixeira, B; Ramos, C. Upgrading BRICKS—The Context-Aware Semantic Rule-Based System for Intelligent Building Energy and Security Management. *Energies* **2021**, *14*, 4541. https://doi.org/10.3390/en14154541

Academic Editors: Jae-Weon Jeong and Wei-Hsin Chen

Received: 31 May 2021
Accepted: 22 July 2021
Published: 27 July 2021

1. Introduction

Several solutions have been developed in the building energy management area in the past decades [1], particularly with the emergence of new technologies, standards, and protocols. These solutions often focus on specific domains such as heating, ventilation, and air conditioning (HVAC) systems, lighting control systems, shading control systems, energy management systems, among others. However, there are no solutions capable of integrating all building management solutions into a single system, ensuring optimized management of the energy resources [2]. The only option being to manage each domain independently. Nowadays, buildings are key elements in the smart grid (SG) domain [1,3] and relevant sources of energy flexibility in the demand response (DR) context [4,5]. A building itself is not enough to provide the needed consumption reduction. However, a DR aggregator collecting each building's contribution reaches the required energy reduction [6,7].

Building management systems (BMSs) have limitations in terms of energy efficiency, such as the need for effective retrofitting of the building's equipment and systems and the need for the detailed identification of the energy consumption points [6,7]. It is even more relevant with the rising demand for dynamic and intelligent building management, enabling consumers to participate actively in energy consumption management, considering the priorities of the loads according to the current context [8]. Current BMSs rely on simplistic controls that require the user to enter the information manually. However, intelligent systems should calculate and understand all of these context variables autonomously [9]. Critical buildings (such as hospitals or airports) have a technological ecosystem composed of several closed systems to control equipment and functions of specific domain areas and, therefore, do not allow information exchange nor interoperability between them. It could be interesting to take advantage of the heterogeneous systems' knowledge to provide a more intelligent building management. Additionally, BMSs should integrate renewable energy sources (RES) components, adopting a dynamic resource management philosophy, taking into account several time horizons, and functioning in an isolated mode or connected to the distribution grid [10,11].

The SEmantic SmArt Metering–Services for Energy Efficient Houses (SESAME-S) [12] project developed a semantic smart home energy management system to provide residential users informed decision support for their energy consumption. It takes advantage of semantic web technologies to provide a personalized, intuitive, and efficient tool interoperable with heterogeneous devices and simple to extend, maintain, and upgrade. Constantin et al. [13] present an agent-based building comfort and energy management system for non-residential buildings, using semantic communications. The agents work to balance the trade-off between the occupant's comfort and the energy savings. Howell et al. [14] present a cloud-based building energy management system, supported by semantic middleware, integrating a sensor network with advanced analytics and accessible by a web interface. The proposed system promotes reusability and extensibility as it can be deployed in other buildings without having to redesign its underlying technologies. Muñoz López et al. [15] use ontologies to describe automation and combine it with a system's architecture to provide contextual services. Deployed in a smart office scenario, it connects devices through web services. The use of ontologies enables semantic interoperability and expressiveness in the automation rules' modeling and definition. In turn, Kučera and Pitner [16] present a semantic BMS, which allows using building automation data in facility benchmarking. The semantic BMS provides facility managers with user-friendly, flexible, and dynamic querying over the building automation data for benchmarking and decision support. Petrushevski et al. [17] concentrate on the semantic representation of building systems' information to support advanced data analytics algorithms and improve building energy efficiency. To this end, they extend the model developed in [18] and implement rules to find anomalies in the building monitoring data. However, their approach does not take advantage of semantic reasoners as there is the need to develop functions to execute the rules when they trigger. To reduce the energy gap near real-time, Yuce and Rezgui [19] propose a semantic mapping process to define the most prominent variables. It uses an artificial neural network (ANN) to learn semantic mapping patterns and a genetic algorithm-based optimization tool to generate the energy-saving rules in multiple objectives and constraints. Finally, Tamani et al. [20] introduce a rule-based model for the supervision and management of smart buildings. This proof of concept highlights the potential of declarative rules dealing with building management and supervision.

On one hand, some BMSs focus on buildings' energy efficiency [12,14], while others are pure building automation systems (BAS) without considering the energy management domain [15,16]. The work of Constantin et al. [13] goes a step further by adding the user's comfort to the equation. On the other hand, there are only a few semantic rule-based systems in the building's domain, and not all of them are taking advantage of semantic reasoners, as seen in [17]. Moreover, each of these systems focuses only on solving a specific problem. It would be advantageous to have a configurable tool gathering heterogeneous services for building management from energy efficiency to the building's automation and security, where the user decides which services to use and rules to apply. Although using ontologies, these solutions are not taking full advantage of semantic web technologies and hold onto the semantic models they use. In other words, these solutions are not agnostic to the ontologies they use, and when the model or business rules change, these systems must be coded accordingly to reflect those changes. Additionally, with the SG and DR advent, it would also be beneficial to have in the same BMS the possibility to add DR contracting and management, where the user could set different contracts and the devices available for each contract and their priorities. Even more interesting would be to have a tool able to automatically interact in DR events, respecting the users' preferences and contracts made.

Although there are very important contributions in the literature on BMSs, there is a need to integrate the different systems for more intelligent and efficient management, taking advantage of combining knowledge from the various tools to provide users with advanced solutions adaptable to their needs. This highlights the need to develop an intelligent and secure system to integrate energy management while promoting interoperability between all services. To overcome these issues, the *Building's Reasoning for Intelligent Control Knowledge-based System* (BRICKS) [21] emerged, providing an intelligent, integrated, efficient, and optimized building management and control. It was designed to integrate both Supervisory Control And Data Acquisition (SCADA) systems and smart appliances and to provide intelligent building management by applying contextual rules given the knowledge available from the different sources. BRICKS is modular and can easily be reused in any building, significantly reducing development and deployment costs. The semantic context-aware rule-based system supporting it is agnostic to the semantic model and rules; this way, there is no need to reprogram the system each time the rules or the model update. BRICKS was initially developed as a Java library [21] to deploy both in a software agent or a web service, covering both simulation and real-world environments. The former version of BRICKS contributed:

- a flexible, configurable, and context-aware building management system;
- overcoming the complexity of interaction between heterogeneous devices in a building;
- keeping the system abstracted from the hardware installed and respective communication protocols to avoid reprogramming the system every time a device is installed or removed;
- a rule-based system agnostic to the rules and data, enabling advanced machine intelligence;
- a tool suitable to different buildings by using it or parts of it in other building facilities for which the same semantic model, semantic converter, or semantic rules apply; and
- a centralized interface to manage heterogeneous cross-domain monitoring and alarms.

As the world evolves towards advanced energy management and its efficient use, traditional electricity end-consumers are becoming prosumers who have access to renewable-based generation and energy storage equipment. Prosumers are active players in distributed energy generation (DER), local energy trading, and DR programs [22]. Utilities see industrial and commercial players as economically suitable players for DR due to the amount of energy of their controllable loads [23,24]. However, there are already residential consumers participating in DR programs through direct load control of their appliances [22]. It is hard to motivate residential users to participate in DR programs, mostly when the bill saving is not satisfactory enough to be worth the effort. Additionally, most end-users have limited knowledge or are not aware of the benefits of participating in such programs [25], nor that electricity prices vary according to periods [26]. DR aggregators play an important role as intermediates between utilities and electricity end-users. Aggregating and managing enough loads and DER from consumers and prosumers, aggregators provide a system capacity that fulfills the requirements to participate in wholesale electricity markets, ancillary services, capacity reserves, and balancing provisions [27,28]. These issues motivate the upgrade of BMSs to include DR aggregation and participation, taking advantage of building automation and energy management. Additionally, it would also be interesting to develop a BMS that is able to adapt to different players, from the aggregator to the building manager (industrial, commercial, or residential) to the end-user (residential or commercial), which will allow configuring multiple contracts and different aggregation levels. By aggregation levels, we mean, for instance, a building manager aggregating loads of residential consumers while being aggregated by a DR aggregator.

In the literature, there are several studies on the hierarchical aggregation of demand-side management. Amarasekara et al. [29] propose a hierarchical aggregation methodology for distribution grids composed of multiple microgrids that consider the consumers' constraints and preferences. Bazmohammadi et al. [30] introduce a hierarchical stochastic energy management system for the operation of interconnected microgrids coordinated by the aggregator. Ju et al. [31] present a hierarchical control strategy, from the independent system operator (ISO) to the aggregators and end-users, to coordinate aggregated air-conditioning loads for power regulation and system stability. Du and Li [32] propose a hierarchical market structure for the participation of microgrids, aggregated by the distribution system operator (DSO), in wholesale balancing markets. Yu and Hong [33], in turn, present a hierarchical incentive-based DR model from the grid operator to multiple service providers and their customers. Huang et al. [34] suggest a hierarchical DR control to optimize the operation of a large scale of buildings, where a virtual building represents an aggregation of buildings to be optimized, overcoming the computational load challenge. Tavakoli et al. [35] address a two-stage hierarchical control approach for energy management in DR programs of commercial building microgrids based on local wind power and plug-in electric vehicles. Finally, Wu et al. [36] introduce a hierarchical control framework consisting of a load aggregator, a central controller, and multiple local controllers of residential heating, ventilating, and air-conditioning (HVAC) loads for primary frequency regulation.

These are very relevant works presenting frameworks, models, or strategies for hierarchical DR. However, these are only tested in laboratory environments, leaving their real-world application in BMSs aside. Furthermore, BMSs focus only on specific domains, such as, for instance, HVAC, lighting, or DR. There is a need to overcome this gap so that BMSs become a reality in most buildings and contribute to the buildings' energy efficiency and, consequently, to the distribution system's efficiency. Additionally, to the best of the authors' knowledge, there are not yet BMSs that provide the flexibility of multi-level aggregation of DR in buildings with a building manager aggregating the loads of building apartments, offices, or departments. This work does not propose a new hierarchical algorithm, methodology, or strategy for DR aggregation or control. It presents a tool capable of integrating different services and algorithms for DR at different levels with a simple configuration. Furthermore, it has the flexibility of configuring multiple DR contracts at different levels, as the case study demonstrates, giving the user the versatility of defining its DR contracts with the aggregators that best reward them for each specific case. The new and upgraded version of BRICKS also aims to be flexible enough to be deployed at any aggregation level, from the aggregator to the building manager or end-consumer. This way, it can be used by any player of the domain independently of its role. This work extends the former version of BRICKS by adding new functionality layers to the previously published Java library. The goal is to extend the previously developed version to include tools for DR programs, including contracts, aggregation levels, user's comfort and priorities, and services integration. In addition to the above contributions, the new BRICKS version includes:

- a new *BRICKS Core Module* featuring:
  - an improved *BRICKS Engine* (formerly BRICKS [21]) Java library;
  - a hierarchical DR aggregation management library deployable at any aggregation level from the aggregator to the end-user;
  - the capability to aggregate other BRICKS instances, SCADA systems, and smart appliances as well;
  - the possibility of, as a client, defining multiple DR contracts with different aggregators and devices concurrently;
  - the automatic control of appliances, according to the aggregation contract, for automated DR;
  - the autonomous interaction between BRICKS instances in DR events;
  - a consumption and generation forecast web service;

  - a database for the system's configurations and real-time and historical measurement data;
  - the connection to external web services for players' aggregation, DR programs, players' remuneration, or forecast algorithms;
- an intuitive semantic configuration module; and
- a dynamic user web interface module for the building's management, monitoring, and DR participation, rendered at a runtime given the building's aggregation level.

This way, BRICKS contributes to the evolution of the SG paradigm, the buildings' participation in DR events, and the integration of renewable-based energy sources. It optimizes the use of renewable energy, takes advantage of the loads' flexibility, allows buildings to become active players capable of reducing energy costs, and adopts business models returning profits through real-time management of the buildings' resources [37].

The remainder of the paper is structured as follows. Section 2 overviews the previous work. Section 3 presents the upgraded version of BRICKS, highlighting the novelty in comparison to the previously published work. The case study presented in Section 4 demonstrates the new features of BRICKS in a DR event deployed at the authors' laboratory building. Section 5 asserts the final conclusions and future work.

2. BRICKS Engine Overview

The first version of BRICKS introduced "*a context-aware semantic rule-based system considering context-based profiles for intelligent management of buildings' energy and security.*" [21]. It was developed as a Java library and designed to be easily extended and integrated with existing software and hardware. This library is now named *BRICKS Engine* in the newly upgraded version of BRICKS.

BRICKS Engine provides a centralized interface for building monitoring and alarms, integrating different SCADA systems and smart appliances using both Modbus TCP/IP (https://modbus.org/, accessed on 25 May 2021) protocol and the Hypertext Transfer Protocol (HTTP) with Representational State Transfer (REST) [38] architectural style. It uses appliances' and sensors' data to trigger notifications, alarms, or automatic control as defined by the system's administrator. The system's administrator defines [21]: the ontology, or ontologies, to use; the SPARQL Protocol and RDF Query Language (SPARQL) (SPARQL is a recursive acronym: https://www.w3.org/TR/rdf-sparql-protocol/, accessed on 25 May 2021) constructs templates to update the KB when translating raw data into the semantic model; the mappings between the raw data and the SPARQL templates tags; the SPARQL queries to get data from the KB; the timestep; and the SWRL or SPARQL semantic rules. The use of ontologies enriches the data gathered from the different devices. The use of semantic web technologies provides the necessary abstraction to avoid re-coding the system anytime the ontology or rules update. The rules are at the software level; thus, *BRICKS Engine* does not depend on any installed device nor communication protocol. To keep the system agnostic to the ontologies and rules, BRICKS uses JavaScript Object Notation (JSON) (http://www.json.org/, accessed on 25 May 2021) data models instead of Java classes, SPARQL CONSTRUCT templates to translate raw data to the semantic model, and SPARQL queries returning JSON strings respecting given JSON schemas. SPARQL CONSTRUCTs are queries returning RDF graphs constructed by substituting variables in a set of triple templates. SPARQL CONSTRUCT templates are SPARQL queries with tags to be replaced by data values. For such, mappings between tags and values' sources are configured in JSON files.

In addition to the devices' data, the *BRICKS Engine* takes advantage of different REST web services to obtain weather data and the correct device's profile for the current context. The data is added to the system's KB to validate the rules and check if any are triggered. To control the building's appliances, *BRICKS Engine* enables both using REST requests to smart devices or existing BAS and Modbus TCP/IP requests to SCADA systems. Currently, it uses a local Weather Service (https://meteo.isep.ipp.pt/, accessed on 25 May 2021) to get solar radiation data to validate photovoltaic (PV) generation; a Context-Profiles Service [39] for an intelligent profile definition; and a Device Reader and Controller Service [40] for the data acquisition and devices' control in the laboratory building. Regarding the semantic rules, it is the responsibility of the system's administrator to write the rules according to the ontologies defined and set the correct execution order. SWRL rules run first to take advantage of the SWRL engine inferences (which turn implicit knowledge explicit), and the SPARQL rules execute after, taking advantage of the inferred knowledge. The *BRICKS Engine* outputs two types of data, namely monitoring data at each timestep and a list of notifications, alarms, or control actions if any rule triggers. The notifications and alarms are meant to be shown to the system's administrator in the UI, while the control actions are automatically executed by *BRICKS Engine* accordingly.

Finally, the first version of BRICKS had two disadvantages that are improved by the new *BRICKS Engine* library. Firstly, its initial configuration was a costly job since it was manually made using a text editor to write the necessary configuration files. Secondly, the minimum acceptable time step was around 30 s and now is reduced to less than 5 s. This improvement is two-fold: primarily, the appliances' readings are made in parallel using threads; and formerly, we have updated all SWRL rules to SPARQL, which, besides being more flexible than SWRL, it allows us to add only the necessary knowledge to the KB, improving inference performance. To be continuously working, *BRICKS Engine* is fault-tolerant, avoiding runtime exceptions if, for instance, a reading fails due to a hardware communication failure. Specific details on the functioning of the different features of the library are available in [21].

3. Proposed BRICKS Upgrade

This section presents the upgrades made and the new features added to BRICKS—the context-aware semantic rule-based system for intelligent building energy and security management. This new version aims to be as flexible and inclusive as its former version. It does not change the previously presented system but uses it as its core module, incrementing new modules and layers to solve additional issues. Its development was carried out within the scope of projects with companies, being part of their commercial products, that do not allow its public availability as open source for copyright reasons. To address the gaps identified in Section 1 a new and improved BRICKS arose by extending the previously developed version to add new features. BRICKS' new version includes a semantic configuration module, a building's management and monitoring UI, a flexible and configurable DR management service, a database for BRICKS' configuration and real-time and historical data, and a triple-store for the KB, on top of all the features reviewed in Section 2. Figure 1 presents BRICKS architecture.

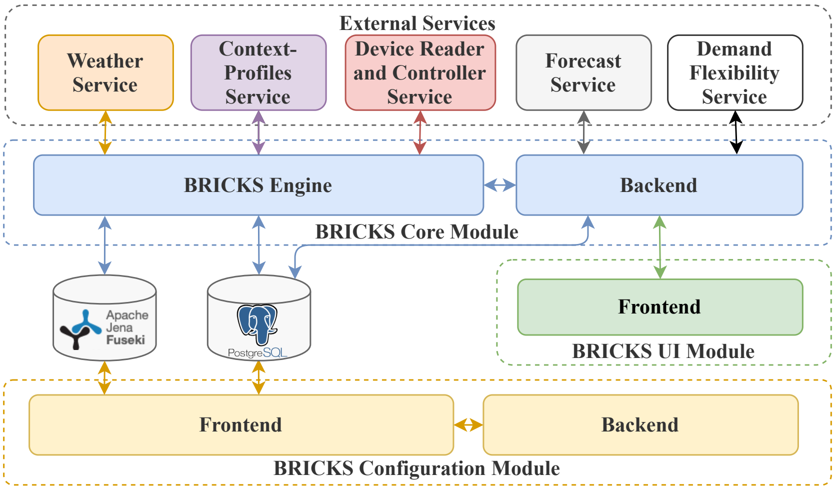
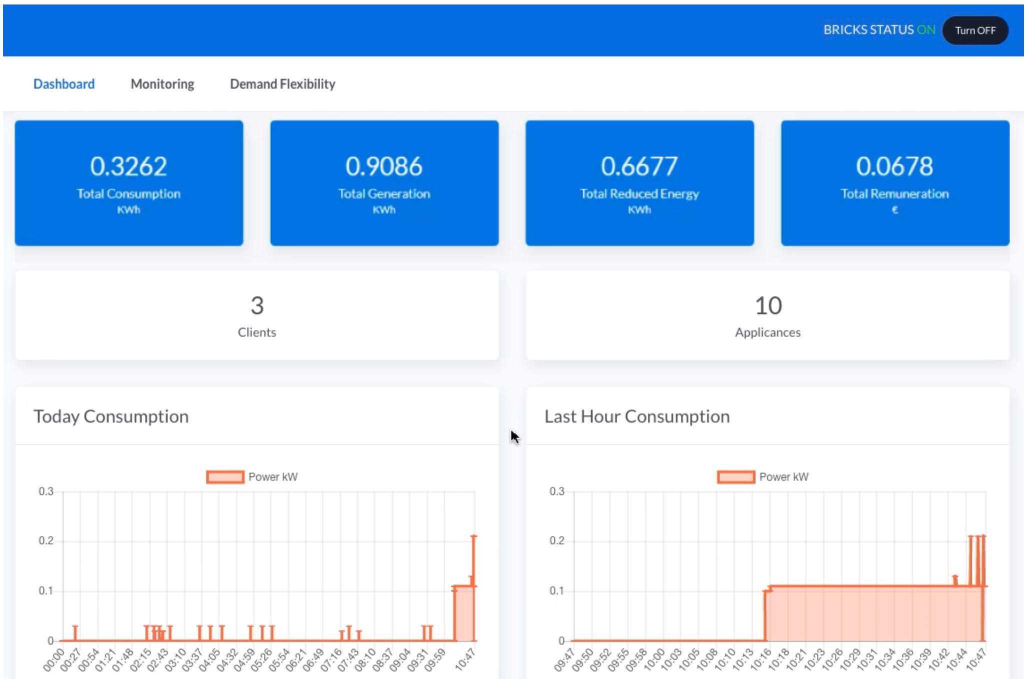


Figure 1. BRICKS' architecture.

BRICKS is now a containerized distributed system featuring three modules: the *BRICKS Core Module*, the *BRICKS UI Module*, and the *BRICKS Configuration Module*. The *BRICKS Core Module* is the main module of BRICKS and is composed of *BRICKS Engine* and *Backend* libraries. Following the initial design principles, this module uses external services for distinct purposes. To improve the *BRICKS Engine* library, the former configuration text files were dropped and replaced by database records and RDF triples in the KB. The system's administrator configures which database and RDF store to use. The relational database holds all static configuration data from *BRICKS Engine* that, formerly, was made on JSON or SPARQL files. These include SPARQL template files, the respective JSON template tags mappings files, SPARQL queries, and the SWRL and SPARQL rules (more details on these configuration files are available at [21]). Additionally, it also stores the real-time and historical measurement data gathered by the *Backend* library, as well as data related to the DR management, such as the appliances and their priorities, contracts, aggregation levels, among others. The triple-store KB, in turn, stores the ontologies and individuals (instances) configured in BRICKS, which were previously managed in both text files (static information) and in-memory (real-time measures and rules' inferences). Currently, the use of ontologies and semantic web technologies is limited to the *BRICKS Engine*.

The *Backend* library is a REST service, providing the necessary endpoints for the management and communication among different BRICKS instances. It includes the configuration and management of DR contracts, the appliances assigned to each contract, the assets' priority during a DR event, the Modbus and REST connections to control the devices, the communications with client BRICKS instances, the external services (including algorithms) to use, and the database and KB connections. The newly developed BRICKS aims the deployment at different aggregation levels, from the aggregator to the office or residential consumer. To this end, BRICKS can aggregate client BRICKS instances, smart appliances, and devices connected to programmable logic controllers (PLCs) using the Modbus protocol. Each BRICKS instance gathers data from its local measurements and from its clients without a BRICKS instance. As a client, BRICKS enables multiple DR contracts with different aggregators, assigning separate devices to each contract. BRICKS automatically controls its appliances and the contracted appliances of clients without BRICKS. During DR events, BRICKS interacts autonomously with other BRICKS instances from the aggregator to the end consumer. BRICKS allows the configuration of different services for the aggregator and client levels, such as the consumption/generation forecast service and the demand flexibility clustering and remuneration service. By default, BRICKS includes an artificial neural network (ANN)-based forecast service [41]. It is up to the system's administrator to use it or configure another option.

The *BRICKS UI Module* provides a user-friendly, configurable, and dynamic user web interface. This module is configured beforehand for the respective BRICKS instance and its clients (without BRICKS). To configure the *BRICKS UI Module*, the system's administrator only needs to define the BRICKS base Uniform Resource Locator (URL), the refresh rate timestamp (in milliseconds) to get updated monitoring measurements from BRICKS, and a unique identifier to the respective BRICKS instance or one of its clients. Only the system's administrator has access to the unique identifiers. With this configuration, the *BRICKS UI Module* builds the UI according to the data provided by the *BRICKS Core Module* for the respective unique identifier. The UI rendering depends on the aggregation level BRICKS is configured or on the client's configuration (in a client without BRICKS). This way, the aggregator's UI differs from the UI of its aggregated clients; the UI of a client with BRICKS (that also aggregates other clients) is also different from the UI of a client without BRICKS (e.g., a building manager aggregating various offices or apartments). Figure 2 shows an example UI of a business building.



BRICKS STATUS ON

0.3262 Total Consumption KWh

0.9086 Total Generation KWh

0.6677 Total Reduced Energy KWh

0.0678 Total Remuneration €

3 Clients

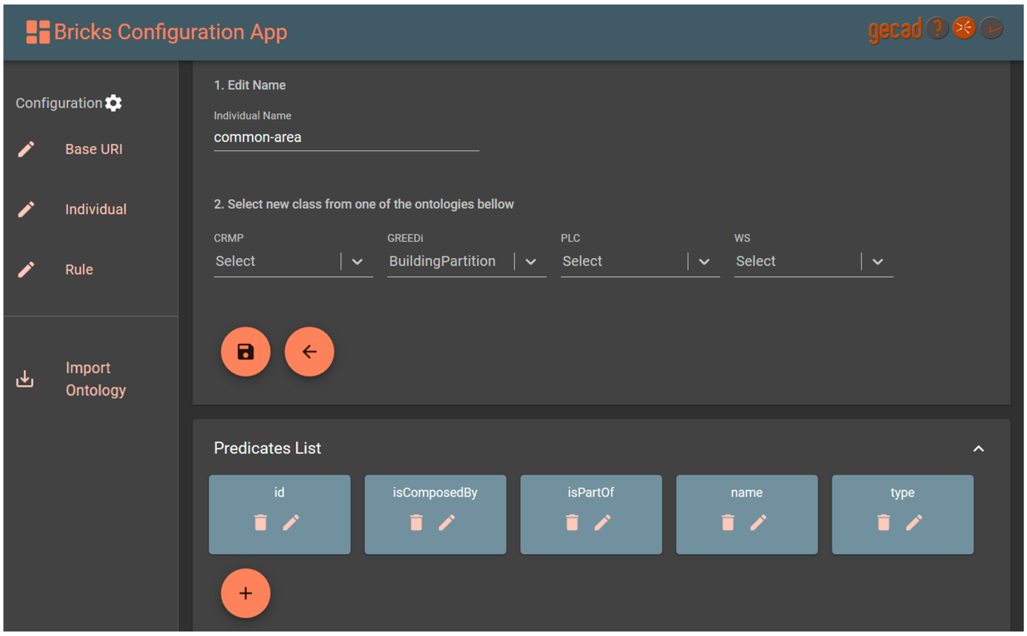
10 Applicances

Figure 2. The business building dashboard.

The business building of Figure 2 aggregates three clients, controlling a total of 10 appliances. Due to privacy issues, only the aggregator can visualize the consumption/generation of clients without BRICKS. Additionally, the total consumption/generation available in the dashboard of an aggregator only considers their measurements (without clients' data).

Finally, the *BRICKS Configuration Module* provides a web-based and intuitive administration service to configure the semantic models, instances, and rules of the *BRICKS Engine*. The need to develop this module arises so that any user can configure the system, even without any knowledge about ontologies or semantic web technologies. Besides, it also aims to avoid human errors when writing text files and reduce the time needed to configure a building and its rules. Although this tool abstracts the user from ontologies and semantic web technologies, an advanced mode is also available for expert users. At the first deployment, the system's administrator must import the ontologies to BRICKS' KB. The system allows importing both local ontology files and online using the ontology's URL. The system provides, by default, a base URI for the ontology individuals that the administrator can change at their will. Using the uploaded ontologies, BRICKS UI dynamically provides the user with the available classes, properties, and relations to create the required individuals to represent the building's topology, the devices, including how to read data from them and control them, the services, and the semantic rules. The next step is to create the ontology individuals describing: the building, its areas, the devices per area, how to read devices' measurements, the commands for controllable devices, among others. Figure 3 depicts a snippet of the configuration of a building partition instance using the *BRICKS Configuration Module*.

Observing Figure 3, one can see, after the Individual Name input field (filled with *common-area*), four select fields with the classes of each uploaded ontology, so the user can pick the class of the new instance to create (set as *BuildingPartition* class). In the predicates list, the user determines the data properties of each individual (e.g., *common-area name "Common Area"*) and the relations of it with other instances (e.g., *common-area isPartOf office-building*). Finally, the system's administrator can define SWRL and SPARQL rules for alarms, notifications, and the automatic control of appliances.



1. Edit Name

Individual Name: common-area

2. Select new class from one of the ontologies bellow

CRMP	GREEDi	PLC	WS
Select	BuildingPartition	Select	Select

Predicates List

- id
- isComposedBy
- isPartOf
- name
- type

Figure 3. The configuration of a building's partition using the *BRICKS Configuration Module*.

4. Case Study

The following case study aims to show the functioning of the newly added features of the upgraded version of BRICKS, namely the multi-level aggregation of consumption flexibility for DR, the ability to manage contracts with multiple aggregators considering different assets, and the integration of external services for distinct purposes. It demonstrates BRICKS' execution in the authors' research lab during a DR event, considering different aggregation levels and contracts. It does not address the *BRICKS Engine* configuration and functioning since it is available in our previous work [21], and the building's assets, including the reading and control hardware, are the same. With other systems, building administrators must install them independently, program the rules according to each system, and use the services provided by the company. Besides not taking advantage of the available knowledge, these tools do not interoperate with external services.

The case study scenario considers an energy flexibility Aggregator with 20 clients, namely a Business Building, two offices (Office 1 and Office 3), a Hospital Building, and 16 dummy clients. From Office 1, he is aggregating the air-conditioning (AC) consumption, and from Office 3, a Smart Plug. From the Business Building, he is aggregating the building's photovoltaic generation and the consumption of the common areas and aggregated offices. From the Hospital Building, he is aggregating the heating, ventilation, and air conditioning (HVAC) system of a non-critical nursery floor. The Business Building, in turn, besides managing the building's common areas, aggregates assets from three offices, namely: the lights of Office 1; the lights and AC of Office 2; and the AC of Office 3. This type of aggregations is only possible due to the proposed system, including using heterogeneous services for specific purposes for distinct types of buildings and devices. Office 1, Office 2, and Office 3 represent three office zones of the authors' research lab, while the Business Building represents the left side of the same building. The Hospital Building, in turn, is emulated in the lab [42]. Additionally, these players use data measured in real-time, while the dummy clients use real-measured data from different departments of the authors' institute. Figure 4 illustrates the case study scenario.

As Figure 4 shows, BRICKS is deployed at the Aggregator, Business Building, Hospital Building, and Office 1. The Aggregator's BRICKS instance interacts with the BRICKS instances of the Business Building, the Hospital Building, and Office 1. It is also configured to monitor and control the Smart Plug of Office 3 autonomously. Similarly, the Business Building monitors and controls the lights and AC of Office 2 and the AC of Office 3 and interacts with the BRICKS instance of Office 1. The Business Building demonstrates a BRICKS instance at an intermediate level, interacting with the systems of the Aggregator and Office 1. Office 1 and Office 3, in turn, demonstrate multiple contracts at different levels. As a provider, BRICKS aggregates other BRICKS instances, smart appliances, and devices connected to PLCs, gathering their measurement data. Furthermore, it automatically controls its appliances and the appliances of clients without a BRICKS instance. As a client, BRICKS enables assigning different devices to separate DR contracts with distinct aggregators.

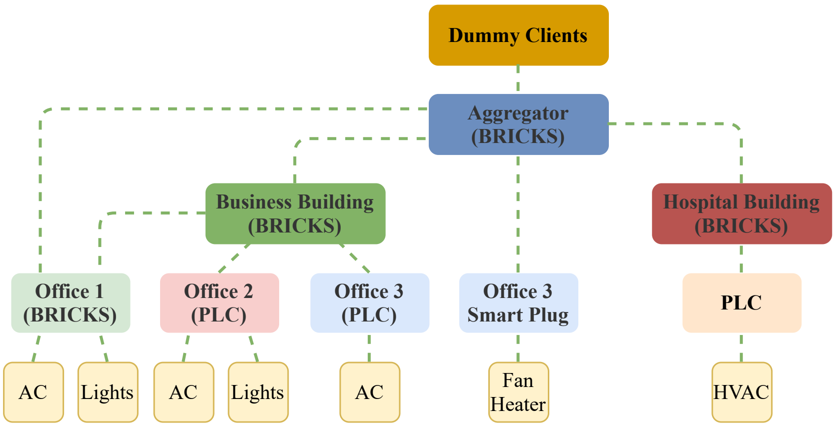


Figure 4. The case study scenario.

The DR events occur every 15 min and, for our case study, we will be looking at the event from 16:00 to 16:15. The period duration of DR events is configurable at the Aggregator's level. For each event, the Aggregator requests their players for the load flexibility available for the next period. This request is performed automatically by BRICKS to the aggregated clients' systems in cascade as, during DR events, it autonomously interacts with other BRICKS instances from the Aggregator to the end consumer. To determine the load flexibility for the next period, each client runs a forecast algorithm. BRICKS allows the configuration of external services for both the aggregator and client levels, such as the consumption/generation forecast service and the demand flexibility clustering and remuneration service. This case study considers BRICKS' default ANN-based forecast service [41] for the consumption/generation forecast and the Demand Flexibility service published by [43] for the aggregator's demand flexibility clustering and remuneration. BRICKS runs the forecast of clients without a BRICKS instance. Thus, when the BRICKS system of the Business Building receives the load flexibility request, it automatically sends a forecast request to the BRICKS system of Office 1 and forecasts the load flexibility of Office 2 (lights and AC), Office 3 (AC), and the building's common areas. Likewise, the Aggregator forecasts the load flexibility of Office 3's Smart Plug. Table 1 presents the load forecasts sent to the Aggregator for the next 15 min.

Table 1. Load forecasts received by the Aggregator.

Client	Load Forecast (kW)
Office 1 (AC)	0.3748
Business Building	2.9665
Hospital Building (HVAC)	2.0843
Office 3 (Smart Plug)	0.7046

After receiving the forecasts, the Aggregator's BRICKS instance runs a Demand Flexibility Service [43] to classify players by clusters, determine the load reduction of each client, and set the remuneration tariff per cluster. The service responds with the Load Forecast, the Reduction, the Cluster, and the Remuneration Tariff for each client. With the results, the Aggregator's BRICKS instance automatically sends reduction requests to client

BRICKS instances and acts accordingly in the appliances of the clients without a BRICKS instance. Table 2 shows the Demand Flexibility Service results.

Table 2. Demand Flexibility Service results.

Client	Load Forecast (kW)	Reduction (kW)	Cluster	Remuneration Tariff (EUR/kWh)
Dummy Client 1	1.2300	1.2300	2	0.1426
Dummy Client 2	0.9021	0	1	0.1652
Dummy Client 3	1.3306	1.3306	2	0.1426
Dummy Client 4	6.6394	0	3	0.1016
Dummy Client 5	5.8934	0	3	0.1016
Dummy Client 6	1.7092	0	2	0.1426
Dummy Client 7	0.5923	0.5923	1	0.1652
Dummy Client 8	3.9525	0	3	0.1016
Dummy Client 9	12.0639	0	3	0.1016
Dummy Client 10	4.6157	0	3	0.1016
Dummy Client 11	7.1017	0	3	0.1016
Dummy Client 12	2.3030	0.2186	2	0.1426
Dummy Client 13	12.0112	0	3	0.1016
Dummy Client 14	3.4986	0	3	0.1016
Dummy Client 15	2.8493	0	2	0.1426
Dummy Client 16	0.0236	0	1	0.1652
Office 1 (AC)	0.3748	0.3748	1	0.1652
Business Building	2.9665	2.9665	2	0.1426
Hospital Building	2.0843	2.0843	2	0.1426
Office 3 (Smart Plug)	0.7046	0.7046	1	0.1652

Observing Table 2, it is perceptible that the demand flexibility algorithm returned three clusters and a remuneration tariff for each. Furthermore, all of the four clients are requested to reduce their total amount of forecasted load flexibility. The load reduction is only effective from 16:00 until 16:15. In this period, each BRICKS system reduces the requested amount by turning off or shifting devices or decreasing the lights' intensity, for example, respecting the appliances' priorities and the user's preferences. At the end of the DR event, BRICKS calculates the remuneration according to the effective reduction provided. The results of the DR event, the remuneration, and its tariff are displayed in the clients' BRICKS UI.

Table 3 presents the Effective Reduction and Remuneration for each player.

Table 3. Effective Reduction and Remuneration.

Client	Effective Reduction (kW)	Remuneration (EUR/kWh)	Remuneration Tariff (EUR/kWh)
Office 1 (AC)	0.3748	0.0155	0.1652
Business Building (Common areas)	0.6677	0.0238	0.1426
Office 1 (Lights)	0.8201	0.0292	0.1426
Office 2 (AC and Lights)	0.7415	0.0264	0.1426
Office 3 (AC)	0.7372	0.0263	0.1426
Hospital Building (HVAC)	0.1523	0.0054	0.1426
Office 3 (Smart Plug)	0.7046	0.0291	0.1652

Analyzing Table 3, only the Hospital Building was unable to accomplish the total reduction required by the Aggregator (2.0843 kW). The remaining players fulfill the requested amount since the load forecast of the Business Building results from the sum of the forecasts of its clients and the building's common areas. A hospital's context, in turn, changes very rapidly, and the load flexibility forecasted 5 min earlier may no longer be valid. This way, the Hospital Building was only able to reduce a small amount of energy consumption. The results demonstrate the advantage of using BRICKS, and it was only possible due to the aggregation of different levels and devices introduced by the tool proposed in this paper.

5. Conclusions

Nowadays, there are several BMS solutions available for several domains, such as: building security, user comfort, and energy management, to name a few. However, solutions aggregating different systems in the same framework in an interoperable manner are expensive. To the best of the authors' knowledge, there is not yet a solution able to interoperate with heterogeneous equipment, gathering their data and taking advantage of this knowledge for cross-domain context-based rules, with the flexibility of seamlessly adding/removing devices and services, making available DR programs and contracts at different levels. To overcome this gap arises BRICKS, an improved and extended semantic rule-based system considering context-based profiles for intelligent building energy and security management.

BRICKS provides an intelligent, integrated, efficient, and optimized building management and control solution. It can be deployed independently or at different DR aggregation levels for integrated building management. It also supports the direct aggregation of appliances of clients without a BRICKS system. It overcomes the main building automation and management issues, such as high costs, installation, complexity, and compatibility. The use of semantic web technologies allows BRICKS to be agnostic to the ontologies used and semantic rules applied. Inference may result directly in actions over the building assets and notifications or alarms to the system manager. The system does not need to be reprogrammed or recompiled for new building equipment since it is independent of the used devices and communication protocols. It can be reused for other buildings or parts of buildings for which the same semantic model and/or rules apply. Additionally, this new version of BRICKS allows the definition of multiple DR contracts with different aggregators simultaneously; the automatic cut, shift, or reduction of devices' consumption; the autonomous interaction among BRICKS systems; and the configuration of external services for distinct purposes. Furthermore, it also provides a user-friendly building monitoring UI and an intuitive semantic configuration interface.

The case study demonstrates BRICKS' use at various levels during the execution of a DR program. It focuses on the autonomous interactions of BRICKS with other BRICKS instances and appliances deployed in the authors' laboratory building using real-time data. The results confirm the expected functioning of the system regarding the various types of aggregation contracts and levels, expressing the system's autonomy in executing DR events while accomplishing the users' needs and priorities. With the proposed system, it is easier to apply DR, meeting worldwide initiatives in terms of flexibility and efficiency of electrical energy consumption and security of the energy system.

There are still some limitations and future work to do to improve BRICKS. A relevant issue to deal with is the amount of data gathered from real-time measurements. To this end, the use of a time-series database is one of the next steps to improve the system's performance. A significant upgrade in progress is the extension of the *BRICKS Configuration Module* to enable the complete configuration and deployment of BRICKS in a new site, as it currently only allows to configure *BRICKS Engine*. *BRICKS UI Module* can also be improved to provide management tools to end-users, and not only to system administrators, which additionally requires the inclusion of users' access management. Another valuable feature would be to give end-users the option of defining an external service to choose which appliances to control at a DR event in a given context, considering the devices' priorities and users' preferences. Finally, the system should be flexible to provide a broader range of aggregation contracts. For instance, at the moment, the remuneration tariff is always set by the aggregator's algorithm, and it would be interesting to have different options where the client and aggregator could negotiate the remuneration tariff.

Author Contributions: Conceptualization, G.S. and Z.V.; methodology, G.S.; software, G.S., R.C. and B.T.; validation, G.S., T.P., Z.V. and C.R.; formal analysis, G.S. and Z.V.; investigation, G.S., R.C. and B.T.; resources, G.S., T.P., Z.V., R.C., B.T. and C.R.; data curation, G.S., R.C. and B.T.; writing—original draft preparation, G.S. and T.P.; writing—review and editing, T.P., Z.V. and C.R.; visualization, G.S. and B.T.; supervision, T.P., Z.V. and C.R.; project administration, Z.V.; funding acquisition, Z.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from FEDER Funds through COMPETE program and from National Funds through Fundação para a Ciência e a Tecnologia (FCT) under projects CEECIND/01811/2017 and UID/EEA/00760/2019. Gabriel Santos was supported by the PhD grant SFRH/BD/118487/2016 from National Funds through FCT.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Calvillo, C.; Sánchez-Miralles, A.; Villar, J. Energy management and planning in smart cities. *Renew. Sustain. Energy Rev.* **2016**, *55*, 273–287. [CrossRef]
2. Lee, S.S.; Kwon, B.; Lee, S.S. Joint Energy Management System of Electric Supply and Demand in Houses and Buildings. *IEEE Trans. Power Syst.* **2014**, *29*, 2804–2812. [CrossRef]
3. Lawrence, T.M.; Boudreau, M.C.; Helsen, L.; Henze, G.; Mohammadpour, J.; Noonan, D.; Patteeuw, D.; Pless, S.; Watson, R.T. Ten questions concerning integrating smart buildings into the smart grid. *Build. Environ.* **2016**, *108*, 273–283. [CrossRef]
4. Aduda, K.O.; Labeodan, T.; Zeiler, W.; Boxem, G.; Zhao, Y. Demand side flexibility: Potentials and building performance implications. *Sustain. Cities Soc.* **2016**, *22*, 146–163. [CrossRef]
5. Yu, Z.J.; Haghighat, F.; Fung, B.C. Advances and challenges in building engineering and data mining applications for energy-efficient communities. *Sustain. Cities Soc.* **2016**, *25*, 33–38. [CrossRef]
6. Marzband, M.; Fouladfar, M.H.; Akorede, M.F.; Lightbody, G.; Pouresmaeil, E. Framework for smart transactive energy in home-microgrids considering coalition formation and demand side management. *Sustain. Cities Soc.* **2018**, *40*, 136–154. [CrossRef]
7. Park, S.; Ryu, S.; Choi, Y.; Kim, J.; Kim, H. Data-driven baseline estimation of residential buildings for demand response. *Energies* **2015**, *8*, 10239–10259. [CrossRef]
8. Wijayasekara, D.; Linda, O.; Manic, M.; Rieger, C. Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1829–1840. [CrossRef]
9. Domingues, P.; Carreira, P.; Vieira, R.; Kastner, W. Building automation systems: Concepts and technology review. *Comput. Stand. Interfaces* **2016**, *45*, 1–12. [CrossRef]
10. Wang, Y.; Wang, B.; Chu, C.C.; Pota, H.; Gadh, R. Energy management for a commercial building microgrid with stationary and mobile battery storage. *Energy Build.* **2016**, *116*, 141–150. [CrossRef]
11. Burmester, D.; Rayudu, R.; Seah, W.; Akinyele, D. A review of nanogrid topologies and technologies. *Renew. Sustain. Energy Rev.* **2017**, *67*, 760–775. [CrossRef]
12. Fensel, A.; Tomic, S.; Kumar, V.; Stefanovic, M.; Aleshin, S.V.; Novikov, D.O. SESAME-S: Semantic smart home system for energy efficiency. *Inform. Spektrum* **2013**, *36*, 46–57. [CrossRef]
13. Constantin, A.; Loewen, A.; Ponci, F.; Mueller, D.; Monti, A. Design, Implementation and Demonstration of Embedded Agents for Energy Management in Non-Residential Buildings. *Energies* **2017**, *10*, 1106. [CrossRef]
14. Howell, S.K.; Wicaksono, H.; Yuce, B.; McGlinn, K.; Rezgui, Y. User Centered Neuro-Fuzzy Energy Management through Semantic-Based Optimization. *IEEE Trans. Cybern.* **2019**, *49*, 3278–3292. [CrossRef] [PubMed]
15. Muñoz López, S.; Fernández, A.; Coronado, M.; Iglesias, C.A. Smart Office Automation based on Semantic Event-Driven Rules. In Proceedings of the Workshop on Smart Offices and other Workplaces, Colocated with 12th International Conference on Intelligent Environments (IE'16), London, UK, 14–16 September 2016; Novais, P., Konomi, S., Eds.; Ambient Intelligence and Smart Environments; IOS Press: Amsterdam, The Netherlands, 2016; Volume 21, pp. 33–42. [CrossRef]
16. Kučera, A.; Pitner, T. Semantic BMS: Allowing usage of building automation data in facility benchmarking. *Adv. Eng. Inform.* **2018**, *35*, 69–84. [CrossRef]
17. Petrushevski, F.; Gaida, S.; Beigelböck, B.; Sipetic, M.; Zucker, G.; Schiefer, C.; Schachinger, D.; Kastner, W. Semantic Building Systems Modeling for Advanced Data Analytics for Energy Efficiency. *Build. Simul.* **2017**, *2017*, 622–627.
18. Schachinger, D.; Kastner, W. Semantics for smart control of building automation. In Proceedings of the IEEE International Symposium on Industrial Electronics, Santa Clara, CA, USA, 8–10 June 2016; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2016; Volume 2016-Novem, pp. 1073–1078. [CrossRef]

19. Yuce, B.; Rezgui, Y. An ANN-GA Semantic Rule-Based System to Reduce the Gap Between Predicted and Actual Energy Consumption in Buildings. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 1351–1363. [CrossRef]
20. Tamani, N.; Ahvar, S.; Santos, G.; Istasse, B.; Praca, I.; Brun, P.E.; Ghamri, Y.; Crespi, N.; Becue, A. Rule-Based Model for Smart Building Supervision and Management. In Proceedings of the 2018 IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 2–7 July 2018; pp. 9–16. [CrossRef]
21. Santos, G.; Vale, Z.; Faria, P.; Gomes, L. BRICKS: Building's reasoning for intelligent control knowledge-based system. *Sustain. Cities Soc.* **2020**, *52*, 101832. [CrossRef]
22. Ponds, K.T.; Arefi, A.; Sayigh, A.; Ledwich, G. Aggregator of demand response for renewable integration and customer engagement: Strengths, weaknesses, opportunities, and threats. *Energies* **2018**, *11*, 2391. [CrossRef]
23. SWECO. *Study on the Effective Integration of Distributed Energy Resources for Providing Flexibility to the Electricity System Final Report to The European Commission*; Technical Report; European Commision: Brussels, Belgium, 2015.
24. Cappers, P.; Mills, A.D.; Goldman, C.A.; Wiser, R.H.; Eto, J.H. *Mass Market Demand Response and Variable Generation Integration Issues: A Scoping Study | Electricity Markets and Policy Group*; Technical Report; Lawrence Berkley National Library: Berkley, CA, USA, 2011.
25. Annala, S. Households' Willingness To Engage in Demand Response in the Finnish Retail Electricity Market: An Empirical Study. Ph.D. Thesis, Lappeenranta University of Technology, Lappeenranta, Finland, 2015.
26. Huq, Z.; Islam, S. Home Area Network Technology Assessment for Demand Response in Smart Grid Environment. In Proceedings of the 20th Australasian Universities Power Engineering Conference (AUPEC 2010) "Power Quality of the 21st Century", Christchurch, New Zealand, 5–8 December 2010.
27. Brown, T.; Newell, S.A.; Oates, D.L.; Spees, K. *International Review of Demand Response Mechanisms*; Technical Report; The Brattle Group Inc.: Sydney, Australia, 2015.
28. Ikäheimo, J.; Evens, C.; Kärkkäinen, S. *DER Aggregator Business: The Finnish Case*; Technical Report; VTT Technical Research Centre: Espoo, Finland, 2010.
29. Amarasekara, B.; Ranaweera, C.; Evans, R.; Nirmalathas, A. Hierarchical aggregation method for a scalable implementation of demand side management. *Comput. Oper. Res.* **2018**, *96*, 188–199. [CrossRef]
30. Bazmohammadi, N.; Tahsiri, A.; Anvari-Moghaddam, A.; Guerrero, J.M. A hierarchical energy management strategy for interconnected microgrids considering uncertainty. *Int. J. Electr. Power Energy Syst.* **2019**, *109*, 597–608. [CrossRef]
31. Ju, P.; Jiang, T.; Li, H.; Wang, C.; Liu, J. Hierarchical Control of Air-Conditioning Loads for Flexible Demand Response in the Short Term. *IEEE Access* **2019**, *7*, 184611–184621. [CrossRef]
32. Du, Y.; Li, F. A Hierarchical Real-Time Balancing Market Considering Multi-Microgrids with Distributed Sustainable Resources. *IEEE Trans. Sustain. Energy* **2020**, *11*, 72–83. [CrossRef]
33. Yu, M.; Hong, S.H. Incentive-based demand response considering hierarchical electricity market: A Stackelberg game approach. *Appl. Energy* **2017**, *203*, 267–279. [CrossRef]
34. Huang, P.; Fan, C.; Zhang, X.; Wang, J. A hierarchical coordinated demand response control for buildings with improved performances at building group. *Appl. Energy* **2019**, *242*, 684–694. [CrossRef]
35. Tavakoli, M.; Shokridehaki, F.; Marzband, M.; Godina, R.; Pouresmaeil, E. A two stage hierarchical control approach for the optimal energy management in commercial building microgrids based on local wind power and PEVs. *Sustain. Cities Soc.* **2018**, *41*, 332–340. [CrossRef]
36. Wu, X.; He, J.; Xu, Y.; Lu, J.; Lu, N.; Wang, X. Hierarchical Control of Residential HVAC Units for Primary Frequency Regulation. *IEEE Trans. Smart Grid* **2018**, *9*, 3844–3856. [CrossRef]
37. GREEDi Project. Homepage. 2017. Available online: http://greedi.ipbrick.com/ (accessed on 21 May 2021).
38. Fielding, R.T.; Taylor, R.N. Architectural Styles and the Design of Network-Based Software Architectures. Ph.D. Thesis, University of California, Irvine, CA, USA, 2000; AAI9980887.
39. Silva, F.; Santos, G.; Praça, I.; Vale, Z. A context-based building security alarm through power and sensors analysis. *Energy Inform.* **2018**, *1*, 41. [CrossRef]
40. Gomes, L.; Vale, Z.; Corchado, J.M. Microgrid management system based on a multi-agent approach: An office building pilot. *Meas. J. Int. Meas. Confed.* **2020**, *154*, 107427. [CrossRef]
41. Jozi, A.; Ramos, D.; Gomes, L.; Faria, P.; Pinto, T.; Vale, Z. Demonstration of an energy consumption forecasting system for energy management in buildings. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11804 LNAI, pp. 462–468. [CrossRef]
42. Vale, Z.; Faria, P.; Abrishambaf, O.; Gomes, L.; Pinto, T. MARTINE—A Platform for Real-Time Energy Management in Smart Grids. *Energies* **2021**, *14*, 1820. [CrossRef]
43. Silva, C.; Faria, P.; Vale, Z. Multi-period observation clustering for tariff definition in a weekly basis remuneration of demand response. *Energies* **2019**, *12*, 1248. [CrossRef]