Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería
y Sistemas de Telecomunicación

CONTRIBUTION TO THE DESIGN, IMPLEMENTATION AND STANDARDIZATION OF SEMANTIC MIDDLEWARE ARCHITECTURES FOR THE SMART GRID

DOCTORAL THESIS

Jesús Rodríguez Molina

Master in Systems and Services Engineering

for the Information Society

2017

Centro de Investigación en Tecnologías de Software y
Sistemas Multimedia para la Sostenibilidad
Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación

CONTRIBUTION TO THE DESIGN, IMPLEMENTATION AND STANDARDIZATION OF SEMANTIC MIDDLEWARE ARCHITECTURES FOR THE SMART GRID

DOCTORAL THESIS

Jesús Rodríguez Molina

Master in Systems and Services Engineering

for the Information Society

Supervisor:

Prof. PhD. José-Fernán Martínez Ortega

Universidad Politécnica de Madrid

2017

DOCTORADO EN INGENIERÍA DE SISTEMAS Y SERVICIOS PARA LA SOCIEDAD DE LA INFORMACIÓN

Tesis Doctoral		
Título	Contribution to the design, implementation and standardization of semantic middleware architectures for the Smart Grid	
Autor	Jesús Rodríguez Molina	
Director	Dr. José-Fernán Martínez Ortega	V^{o}B^{o}.
Tribunal		
Presidente		
Secretario		
Vocal 1º		
Vocal 2º		
Vocal 3º		
Suplente		
Suplente		
Lugar y fecha de lectura	E.T.S.I. y Sistemas de Telecomunicación (U.P.M.)	
Calificación		

El Presidente El secretario Los vocales

Tesis Doctoral para la obtención del título de Doctor
por la Universidad Politécnica de Madrid

2017

A happy life is impossible; the best a man can attain is a heroic life, such as is lived by one who struggles against overwhelming odds in some way and some affair that will benefit the whole of mankind, and who in the end triumphs, although he obtains a poor reward or none at all

Arthur Schopenhauer

You promised me Mars colonies. Instead, I got Facebook

Buzz Aldrin

Now I must go to war. We must all believe we have a future. We must fight for those who aren't even born yet!

Terra Branford

Table of contents

List of figures

List of tables

Acknowledgments

This thesis is the culmination of a collection of works and efforts shared with many people in the Next Generation Networks and Services Group (GRyS) who have been true comrades in arms during the good and the bad times that I have gone through there. It is when trouble and stress happen that the true nature of a person is revealed, and I am grateful to have found workmates that have helped me through thick and thin, regardless of my opinions or comments.

I would like to start by thanking the supervisor of my thesis, Professor José-Fernán Martínez Ortega. He has given me lots of motivation and encouragement during all the stages of my education, from the Bachelor thesis to my doctoral studies.

I would also like to thank Pedro Castillejo for all the support that he has provided me during all these years. It has been a pleasure and a privilege working with him all this time, and it is great being able to count on such a knowledgeable and patient person. I also want to thank David Gómez for his positivity and good spirit. He has been a firm source of tranquillity in moments of utter chaos.

I would also like to thank Alexandra Cuerva, Carlos Estévez, Esther Moreno and José Antonio Sánchez for the times we have spent together, either working, writing papers or just sharing a laugh in the cafeteria of the university. I am sure that whatever is ahead for them in their lives will be great. I also want to mention the strong inspiration that the hard work that my Chinese colleagues Xin Li, Ning Li, and Xin Yuan give me. As for the newest additions to the group, I am sure that Yuwen Chen and Zhaoyu Zhai are on their way to become great researchers.

Thanks to the combined efforts of my supervisor, Professor José-Fernán Martínez Ortega, and Doctor Geir Mathisen I did a three month stay in the Information and Communication Technologies department of SINTEF, a large research institution located in Norway, where I made part of my thesis and learnt a bit about how things are done in that country. I got to meet brilliant researchers with very impressive backgrounds, and I am very grateful for that too.

I want to thank my girlfriend Corianne for her unconditional love during all this time, no matter how harsh circumstances have been for us sometimes. Also, I would like to use these lines to thank my parents for their positivity, regardless of how bad things looked.

Last but not least, I want to mention all my friends and people that are part of my life outside work: my friend Javiii, my university friends César, Félix, Adriana, Álvaro, Esther, Carlos, Diego, Guille, Randa, Fran, my little nieces Alejandra and Olivia, my brother-in-law José Carlos and my sister Mónica.

Thank you all and let us hope we have a bright future. *Tomorrow never knows*.

Abstract

The Smart Grid, conceived as the power grid enhanced with Information and Communication Technologies aimed to optimizing electricity consumption, enabling a bidirectional participation in the energy provided and improving the power grid with features like Demand Response, Demand Side Management or Optimal Power Flow, is becoming one of the most compelling Cyber-Physical Systems that is being currently developed. Its capacity to increase usable energy in a sustainable manner, along with collect more information about its actual usefulness, makes possible the improvement of living standards for many people all around the world in a more transparent and open way. What is more, since it is able to integrate the comparatively small power supplied by the Renewable Energy Sources provided by the end users that participate in the Smart Grid (which become “prosumers”, as they both produce and consume electricity), it democratizes access to energy and enables a higher degree of competitiveness between traditional actors in the energy markets and newcomers, thus granting the overall improvement of the services that can be provided by the utility companies.

However, there are still many open issues that must be solved before completely using the Smart Grid to our advantage. Among these open issues, interoperability of its installed devices is a major one. The equipment that is used in deployments of this kind (Advanced Metering Infrastructure, Phasor Measurement Units, Remote Terminal Units, etc.) is manufactured by different companies with different backgrounds and interests. Therefore, the implementation of their products has usually different ways to transmit information or even proprietary solutions with a low degree of compatibility with other pieces of hardware. In a way, the status quo is similar to the situation of computer networks before the first standards were released: a plethora of manufacturers offer their own solutions to provide services and connectivity, but they struggle to work cooperatively with developments of other equipment vendors that may have different perspectives on the technologies that can be used for data transmission.

In addition to that, the services that should be available for either a Smart Grid or one of its smaller scale counterparts (microgrids, nanogrids) are not made clear, neither in terms of what services they should be or where they should be located. While there are some high level functionalities that are usually regarded as almost mandatory (the aforementioned Demand Side Management, device registration, Demand Response, Optimal Power Flow), other more data-centric facilities are often portrayed in a way vaguer manner. Security specifications, the existence of semantic capabilities, how to access the capabilities in a specific deployment or even how the hardware devices become integrated is not described with enough detail. That issue jeopardizes the main purpose of installing and developing components in this area of knowledge, because it makes difficult a further integration of both legacy systems that may have been used by large utility companies for a long time and new developments done by smaller companies that want to play a role in the Smart Grid.

Fortunately, many of these challenges can be solved by implementing a software layer located between the applications that can be included for the benefit of the end users and the

hardware and network infrastructure installed for package and binary data interchange. This software layer, commonly referred as middleware, has as its main purpose abstracting the heterogeneity and complexity of the underlying distributed hardware components, so that it will offer to the high, more application-based layer a collection of facilities of homogeneous, centralized appearance, usually shaped as an Application Programming Interface that can be accessed by the application developers. Middleware is a very useful software tool for Cyber-Physical Systems and distributed solutions because it grants the integration of almost any kind of device, either by adding software components in the device itself or in another part of the system, which must be open enough to have the components installed or have the required computational capabilities to have those components installed.

The main original contribution to knowledge of this doctoral thesis is offering a proposal for a model of a semantic middleware architecture for the Smart Grid, based on software components for distributed solutions. This model is aimed to be used in any kind of deployment related to the Smart Grid, as well as providing a common set of components and interfaces to be observed in future implementations. This architecture has been called Common Middleware Architecture (CMA), as it aims to provide the necessary software components for middleware development under any imaginable use case within this application domain. It has been designed based on the experience accumulated from several research projects where the implementation of a middleware layer was one of the main achievements. CMA has been designed with the main needs of a middleware solution in mind, such as hardware abstraction, context awareness, device registration, interfaces for the upper level, securitization and device integration. While the main domain of CMA is the Smart Grid, and demonstrators based on the Smart Grid have been used to validate it, CMA can also be adapted to other environments.

All in all, the main objective of this thesis is creating a reliable framework for the development of middleware solutions for the Smart Grid, which can be used in other application domains where there are requirements of hardware abstraction and service availability resembling the ones that can be found in this area of knowledge. Another major objective of this thesis is making contributions to the standardization of middleware development for the Smart Grid, so that there will be a specific set of services to be developed in order to comply with the most important functionalities of middleware (hardware abstraction, homogeneous set of services for applications, encasing services based on semantic capabilities and security). These two objectives have been achieved with the contributions done in the study of the State of the Art, the inference of open issues and challenges, the establishment of a list of functional and non-functional requirements and the validation of the proposal put forward in this manuscript. The solutions developed can be regarded as the background of the architecture described here, and therefore its performance should be good enough for the functionalities carried out for this kind of software layer, which should be present in any distributed or Cyber-Physical System that uses a collection of deployed pieces of equipment with different capabilities. Besides, since this is a middleware solution solves problems for issues present in other distributed and/or Cyber-Physical Systems (the Internet of Things, underwater robotics) it can be ported to other domains with ease, as services as high level interface access or device registration are used in those situations as well.

Resumen

La Red Eléctrica Inteligente, concebida como el tendido eléctrico mejorado con las Tecnologías de la Información y las Comunicaciones dirigidas a optimizar el consumo de electricidad, permitir una participación bidireccional en la energía suministrada y mejorar la red eléctrica con características como la Respuesta ante la Demanda, Gestión de la Demanda o el Flujo de Potencia Óptimo, se está convirtiendo en uno de los sistemas ciberfísicos más convincentes desarrollado actualmente. Su capacidad para aumentar la energía utilizable de manera sostenible, así como obtener más información sobre su utilidad real, hace posible la mejora del nivel de vida de muchas personas en todo el mundo de una manera más transparente y abierta. Además, al poder integrar la relativamente pequeña potencia suministrada por fuentes de energía renovable aportadas por los usuarios finales que participan en la Red Eléctrica Inteligente (que se convierten en "prosumidores", ya que ambos producen y consumen electricidad), democratiza el acceso a la energía y permite un mayor grado de competitividad entre los actores tradicionales de los mercados energéticos y los recién llegados, garantizando así la mejora general de los servicios que pueden aportar las empresas de utilidades.

Sin embargo, todavía hay muchas cuestiones abiertas que deben ser resueltas antes de utilizar completamente el potencial de la Red Eléctrica Inteligente en nuestro beneficio. Entre estas cuestiones, la interoperabilidad de los dispositivos instalados es una de los principales. El equipo que se utiliza en los despliegues de este tipo (Infraestructura de Medición Avanzada, Unidades de Medida de Fasores, Unidades Terminales Remotas, etc.) es fabricado por diferentes compañías con diferentes conocimientos e intereses. Por tanto, la implementación de sus productos tiene a menudo diferentes formas de transmitir información o incluso soluciones propietarias con un bajo grado de compatibilidad con otros dispositivos. De alguna manera, el statu quo es similar a la situación de las redes de ordenadores antes de que se publicaran los primeros estándares: un conjunto de fabricantes ofrecen sus propias soluciones para proporcionar servicios y conectividad, pero tienen problemas al cooperar con los desarrolladores de otros fabricantes de equipos que pueden tener diferentes perspectivas sobre las tecnologías que se pueden utilizar para la transmisión de datos.

Aparte de eso, los servicios que deberían estar disponibles para una Red Eléctrica Inteligente o una de sus contrapartes de menor escala (microrred, nanorred) no están claros, ni en términos de qué servicios deben ser o dónde deben estar ubicados. Si bien hay algunas funcionalidades de alto nivel que generalmente se consideran casi obligatorias (las anteriormente mencionadas Gestión de Lado de la Demanda, registro de dispositivos, Respuesta a la Demanda, Flujo Óptimo de Potencia), otras facilidades centradas en datos se describen de una manera más vaga. Las especificaciones de seguridad, la existencia de capacidades semánticas, cómo acceder a las capacidades en un despliegue específico o incluso cómo los dispositivos de hardware se integran no se describen con suficiente detalle. Esta cuestión pone en peligro el objetivo principal de instalar y desarrollar componentes en esta área de conocimiento, ya que dificulta una mayor integración de tanto sistemas heredados que pueden haber sido utilizados

por las grandes empresas de utilidades durante mucho tiempo como nuevos desarrollos realizados por empresas más pequeñas que quieren desempeñar un papel en la Smart Grid.

Afortunadamente, muchos de estos retos pueden resolverse mediante la implementación de una capa de software ubicada entre las aplicaciones que se pueden incluir para el beneficio de los usuarios finales y el hardware y la infraestructura de red instalada para el intercambio de datos binarios y de paquetes. Esta capa de software, comúnmente referida como middleware, tiene como objetivo principal abstraer la heterogeneidad y complejidad de los componentes de hardware distribuidos subyacentes, de manera que ofrecerá a la capa alta y más basada en aplicaciones una colección de facilidades de aspecto homogéneo y centralizado, comúnmente conformada como una Interfaz de Programación de Aplicaciones a la que pueden acceder los desarrolladores de aplicaciones. El middleware es una herramienta de software muy útil para sistemas distribuidos y ciberfísicos porque permite la integración de casi cualquier tipo de dispositivo, ya sea añadiendo componentes de software en el propio dispositivo o en otra parte del sistema, la cual debe ser lo suficientemente abierta para tener los componentes instalados, o bien debe tener las capacidades de computación necesarias para tener esos componentes instalados.

La principal contribución original al conocimiento de esta tesis doctoral es ofrecer una propuesta para un modelo de arquitectura de middleware semántico para la Red Eléctrica Inteligente, basado en componentes software para soluciones distribuidas. Este modelo está destinado a ser utilizado en cualquier tipo de despliegue relacionado con la Red Eléctrica Inteligente, así como a proporcionar un conjunto común de componentes e interfaces a ser tenidos en cuenta en futuras implementaciones. Esta arquitectura se ha denominado Common Middleware Architecture (CMA), ya que tiene como objetivo proporcionar los componentes de software necesarios para el desarrollo de middleware en cualquier caso de uso imaginable dentro de este dominio de aplicación. Se ha diseñado contando con la experiencia acumulada en varios proyectos de investigación en los que la implementación de una capa de middleware fue uno de los principales logros. La CMA ha sido diseñada teniendo en cuenta las principales necesidades de una solución middleware, como abstracción de hardware, conocimiento del contexto, registro de dispositivos, interfaces para el nivel superior, securización e integración de dispositivos. Aunque el dominio principal de la CMA es la Red Eléctrica Inteligente, y se han utilizado demostradores basados en la Red Eléctrica Inteligente para validarla, la CMA también puede adaptarse a otros entornos.

En general, el objetivo general de esta tesis es crear un marco fiable para el desarrollo de soluciones de middleware para la Red Eléctrica Inteligente que pueda ser utilizado en otros dominios de aplicaciones donde existen requisitos de abstracción de hardware y disponibilidad de servicio similares a los que se pueden encontrar en esta área del conocimiento. Otro objetivo principal de esta tesis es contribuir a la estandarización del desarrollo de middleware para la Red Eléctrica Inteligente, de modo que habrá un conjunto específico de servicios desarrollados para poder cumplir con las funcionalidades más importantes del middleware (abstracción del hardware, conjunto homogéneo de servicios para aplicaciones, encapsulación de servicios basados en capacidades semánticas y seguridad). Estos dos objetivos se han logrado con las aportaciones realizadas en el estudio del estado del arte, la inferencia de cuestiones y temas abiertos, el establecimiento de una lista de requisitos funcionales y no funcionales y la validación de la propuesta presentada en este manuscrito. Las soluciones desarrolladas pueden ser consideradas como los antecedentes de la arquitectura aquí descrita, por lo que su rendimiento debe ser lo suficientemente bueno para las funcionalidades realizadas para este tipo de capa de software, la cual debería estar presente en cualquier sistema ciberfísico o distribuido que use un grupo de equipos de diferentes capacidades que haya sido desplegado. Además, puesto que se trata de una solución de middleware que resuelve problemas para cuestiones presentes en otros sistemas distribuidos y / o ciberfísicos (Internet de las Cosas, robótica subacuática), puede ser portado a otros dominios con facilidad, ya que servicios como interfaces de acceso a alto nivel o registro de dispositivos también se utilizan en esas situaciones.

1. Introduction and objectives

This chapter offers information about the historical context where the Smart Grid and middleware started off and were developed. In addition to that, a description of the objectives that were established at the start of the thesis is present in this section as well. Finally, the background used for research activities and the outlines of the dissertation have been included as well.

1.1. Motivation

Electricity is pivotal in the development of all the goods and services that are manufactured, developed and traded since the Second Industrial Revolution, which happened around more than one hundred years ago [1]. It was during that time when fossil fuels began to be burned in an intensive manner to obtain electricity as an output, rather than using the generated heat to move mechanical parts of steam-powered machinery. While electricity generation was being crafted, the development of infrastructures that made possible its supply to the production centres where it was being used was also required. Later on, as standards of living improved, electricity became a service demanded in dwells and households. Consequently, a power grid was built to transport electricity from its production facilities to the clients and end users. Typically, a power grid will work by transferring high voltage electricity originated in a power plant (usually of an order of magnitude ranging in the hundreds of thousand Volts; High Voltage Direct Current or HVDC is the most common way to transport electricity at this stage [2]), which is better suited for travelling long distances, and it will be progressively converted to lower voltages until it is delivered to its consumption location, where voltage gets scaled down to a range from 125 to 240 Volts, depending on factors as the country or the age of electric infrastructure owned by the end user. Power lines will have different characteristics depending on the voltage they are expected to work with. In this kind of system, there are several actors that, while presenting some differences depending on the country and their legislation, play a role in the development and exploitation of such a system. These actors are:

- Distribution System Operator (DSO): it is responsible for electricity generation in the system. Commonly, DSOs own the power plants required for electricity generation, that may use renewable (hydropower, photovoltaic or thermal solar power, etc.) or fossil fuels (coal, oil, natural gas). Thus, a DSO plays a more important role in electricity generation rather than distribution.
- Transmission System Operator (TSO): it provides the required infrastructure to transfer electricity from the power plant to the consumption points. Power lines used for that purpose are usually property of the TSO.
- Aggregator/Retailer: it is the entity that purchases electricity from the DSO and sells it to the end users. Depending on how “smart” the power grid is, it will either carry out functionalities related to aggregation or it will be solely focused on electricity trade.
- End user/Client: it is the entity that purchases the electricity offered by the former actor and consumes it. As far as the conventional power grid is concerned, this is where the electricity flow ends, as the energy it carries is often transformed into other kinds (luminous, mechanical, thermal, etc.).

Overall, these actors tend to be the same in most of the world, even though there might be some major differences depending on the countries (in Spain, for example, it is common that the Distribution System Operator or DSO will also play an active role in the commercialization of electricity to end users via second party companies, so the DSO is also present as a retailer). The principles of the currently used power grid are based on the theoretical and practical work performed by researchers such as Nikola Tesla [3] [4] and its implementation and installation makes use of techniques that date back to at least the 1960s [5] [6]. As a direct consequence of the latter fact, the regular power grid has become outdated in many ways, revealing itself incapable of keeping pace with several technological innovations that have taken place during the last decades. To begin with, renewable energies are increasingly used to produce electricity as an alternative to non-renewable, fossil-based energy sources. Despite some issues such as its intermittent availability or aesthetic impact [7] [8], its friendliness towards the environment and availability for comparatively small users forecast a growing utilization of these resources in the immediate and distant future, even in the worst case scenarios that have been devised [9]. Secondly, the availability of Renewable Energy Sources (RESs) for the small users as Distributed Energy Resources (DERs) has the potential of significantly changing the traditional, one-way flow of the electricity that has been traditionally implemented. Last but not least, by means of using means of energy storage (such as home batteries), electric energy does not need to be consumed as it is received and can be turned into other kinds of energy (for example, chemical energy), stored and used during a more suitable time, which may be when it is actually needed (in case of blackouts or power failures) or when trade opportunities appear in the electricity market, if the user has technology capable of interacting with them. Consequently, the end user can be not just a consumer of energy, but an actor capable of producing its own electricity, thus becoming enabled to either use it for themselves or obtaining a profit from its trade; hence the term *prosumer*, which is heavily linked to the Smart Grid and is referred to an end user capable of producing and consuming energy at the same time (*pro*ducer and con*sumer* are merged as *prosumer*).

It is due to these facts that the conventional power grid can be turned into a *Smart Grid*. The Smart Grid can be defined as the power grid after becoming enhanced with Information and Communication Technology (ICT) with the aim of providing a rationalized usage of energy, more oriented to the supplied resources (thus attempting to make use of what is available at a particular moment) rather than solely taking demand into account (which can be satisfied by adding more power plants and energy resources, but not at a sustainable pace in a finite world). As stated in [10], "*Smart Grids increase the connectivity, automation and coordination between suppliers, consumers and network by modernizing grid features like demand side management, generation, real-time pricing, and automated meter activation and reading*". Figure 1 shows the changes that are introduced by adding Smart Grid capabilities to the overall system. As it can be seen, while many of the actors and roles participating in the Smart Grid are still the same, there are major changes in the functionalities performed by end users (or more accurately said, prosumers).

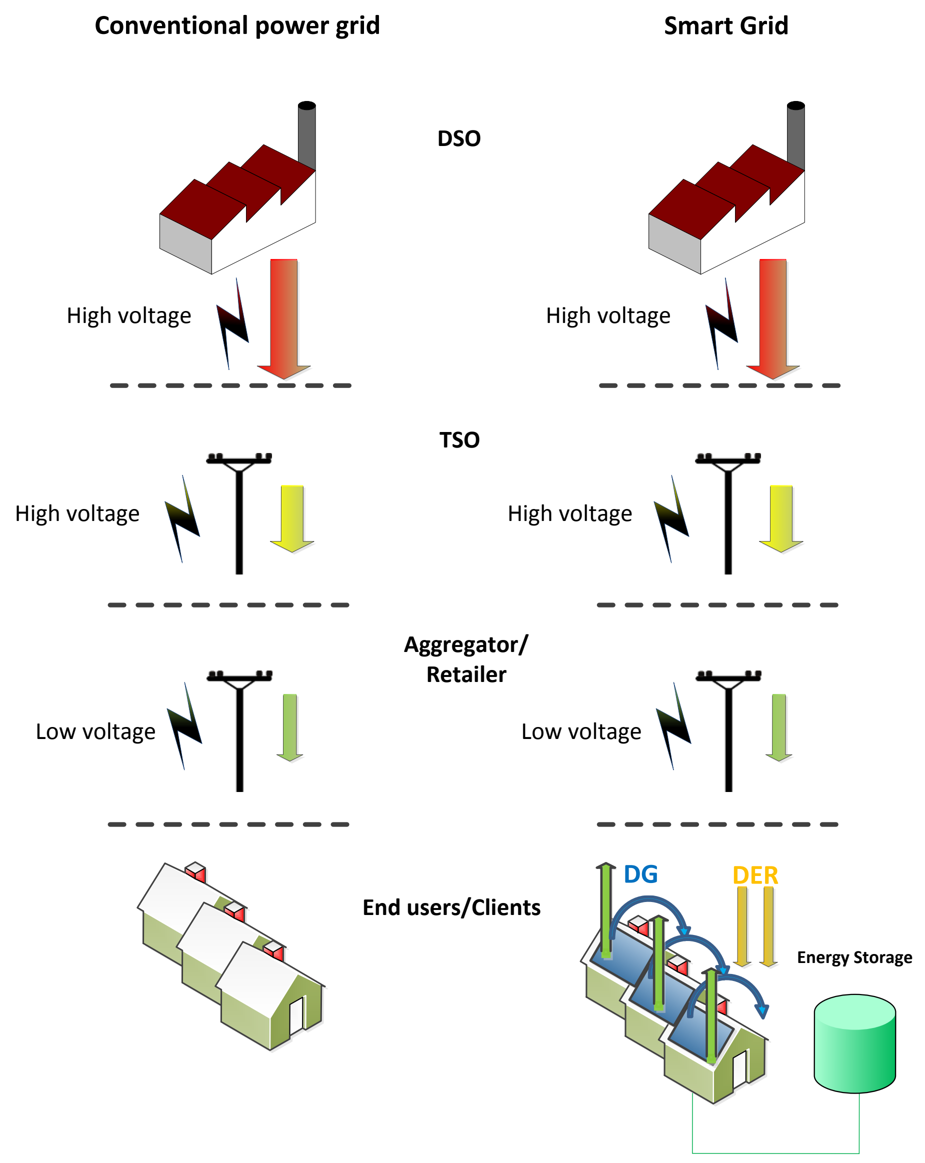

Figure 1. Graphical comparison between the conventional power grid and the Smart Grid, inspired from [11]

The decentralization of electricity production introduces some more significant changes in the whole behaviour of the Smart Grid. As it has already been mentioned, it adds a collection of small energy producers that require that the power flow will become bidirectional, as there is no longer a single major electricity supplier with the energy production centralized in one or a group of facilities supplying the electricity to peripheral consumers. Furthermore, the transition to a supply-based way to consume energy rather than a demand-based one is also encouraged by the adoption of the Smart Grid, where prosumers can trade their electricity surplus, taking into account how much and where it is needed or at what price it can be supplied. Prosumers as a new actor in the way electricity is produced, traded and consumed are a major innovation of the Smart Grid, with a great impact in the whole system due to several reasons:

1. Energy access democratization. Prosumers own means for energy production, which are commonly based on Renewable Energy Sources (RES). Consequently, they are able

to meet their power demands to an extent, depending on the extension of their available RESs and the energy they consume, and do not depend on DSOs as heavily as regular consumers. In this way, energy becomes available in a way that was not possible before.

2. Energy consumption patterns. Prosumers integrated in the power grid alter power consumption patterns dramatically; depending on the time of the day, month of the year and their consumption history, they will either inject power to the grid or will demand less than a regular consumer. Those two actions will greatly change how energy is consumed in a specific area, regardless of its geographical limits (nanogrid, microgrid or larger).
3. Trade for small users. Prosumers could be willing to trade their produced electricity as if it was any other commodity. Depending on their willingness or knowledge, trading algorithms could be used in the development of business models based on energy storage and delivery according to the demands of other prosumers, regular consumers or even the DSOs.
4. Environment-friendliness. Since the overwhelming majority of the prosumers use to produce electricity infrastructures based on renewable energies, more energy is available in an environment-friendly manner, with all the positive consequences that it carries (lower amount of carbon dioxide released to the atmosphere, less pollution, etc.).
5. Impact in other partners involved in the power grid. The integration of this new actor with the power grid will force a redefinition of the roles of other entities involved in the production, transfer and consumption of electricity, as they will require to adapt their offer to what prosumers can provide (DSOs) or gather information about consumption patterns (aggregators).

Overall, and as stated in [11], prosumers will be fully involved in the economic and technological parts of energy consumption, as well as creating new value for the enhanced power grid. A mathematical analysis that models the potential profitability of their activities has been included in the next chapter. The integration of prosumers in the Smart Grid is one of the benefits of enabling middleware in this application domain, for it will enable the necessary mechanisms for data transmission among the different hardware elements of a deployment where prosumers are contributing to the power flow with their own Renewable Energy Sources.

Figure 2 summarizes that new paradigm in the Smart Grid once prosumers have been fully integrated, along with the possible cooperation between entities that they offer: they are capable of receiving and providing assets (either electricity or information) to the Energy Services Company (ESCO) or to other prosumers, which can be organized as a cluster in a Virtual Power Plant (VPP) style. At the same time, DSOs can trade electricity with prosumers (either as a way to have energy or have it as a complementary resource) and with the aggregator in order to offer its resources and information. All these relations are included in the generation of electricity rather than its mere consumption (as regular end users do) and its transmission by means of the infrastructure owned by the Transmission System Operator.

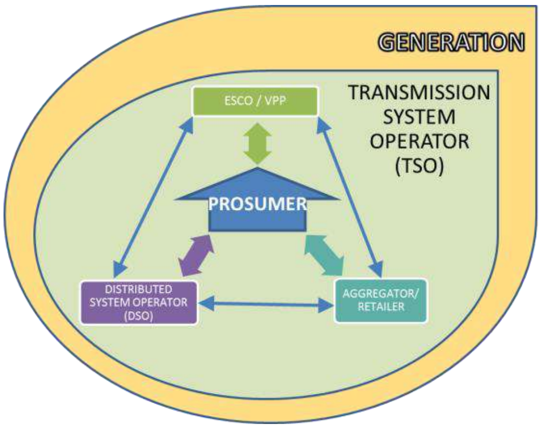
GENERATION
TRANSMISSION SYSTEM OPERATOR (TSO)
ESCO / VPP
PROSUMER
DISTRIBUTED SYSTEM OPERATOR (DSO)
AGGREGATOR/ RETAILER

Figure 2. Energy market value chain focused on the prosumer, as represented in [11]

The integration of prosumers and their small scale facilities used to generate electricity is a critical task for the Smart Grid, especially if all the changes that have to be made to guarantee it are taken into account. These latter can be defined by two different categories:

1. Technical: there are several activities that must be undertaken to offer integration at several layers, such as the physical one (providing all the necessary connections to the power grid and the information transferred from one piece of equipment to the other at the bit level), the network (so as to interchange packages between the existing prosumers, which will effectively behave as a distributed system) and the data-based ones (in order to exchange information about the services and features of the system that will be mutually understood by all the ends involved in the communication).
2. Legislative: prosumers are different from the entities traditionally involved in the generation of energy in the sense that their infrastructure is small-sized and is scattered in a certain area rather than centralized in a position. That would require the rearrangement of the legislative framework used to include the prosumers into the energy markets. For example, the European Energy Exchange includes, among other preconditions for admission, liable equity of at least 50,000 € [12]. This and other requirements might be challenging to be met by individual prosumers. Nevertheless, they could be offered an alternative, such as federating themselves into a larger cluster of energy providers making use of an aggregator.

Information and Communication Technologies must also be described. The Smart Grid makes use of very different appliances with a huge disparity regarding their features, manufacturers or data formats used to transfer data throughout the system. The lack of standardized procedures specific to the software in the Smart Grid or software solutions to interoperate these devices at the data level is one of the major challenges that are faced in this application domain, as interoperability of the devices used cannot be guaranteed and there is a chance for them to become isolated from the other devices used for data transmissions. Fortunately, this issue can be solved by means of *middleware*. Whereas middleware is of more recent conception than the power grid, it was firstly referenced in a NATO document that dates back to 1968 [13], so it is still almost 50 years old. Popularity of middleware grew at a slowly pace during a time where distributed systems were underdeveloped and mainframes were the most common location of most of the computational and logical resources in the firstly available computer networks (thus having a central hardware component providing most of the services, instead of a collection of computers). By the 1980s, though, middleware was being increasingly used as a way to interconnect legacy systems with new ones. As it will be described in subsequent chapters of this thesis, this functionality is still satisfied by middleware for distributed systems. Its usability and utility became greatly expanded during the late 1990s and early 2000s with the uprising of Internet usage, the first developments dealing with the Internet of Things (IoT) and mobile computing, as those are systems that are by definition distributed and heterogeneous [14] [15]. It was during that time that middleware for distributed systems was definitely established as the software entity capable of abstracting heterogeneity from an underlying collection of hardware devices in order to offer a set of homogenous, centralized-looking facilities (typically, an Application Programming Interface or API) to the application layer [16] [17]. Further developments, such as Cloud Computing, have made middleware a software entity of critical importance in the development of distributed systems [18] [19]. Figure 2 illustrates the typical location of middleware in the latter; due to this and how it is accessed by the application layer it effectively withholds the different (and quite often, opposite) hardware components present in a deployment, as well as their local firmware and operating systems. Even though these systems usually interconnect with each other by interchanging messages at the network layer, they still are lacking data formats that can be apprehended by a human being, as middleware does when sending information upwards.

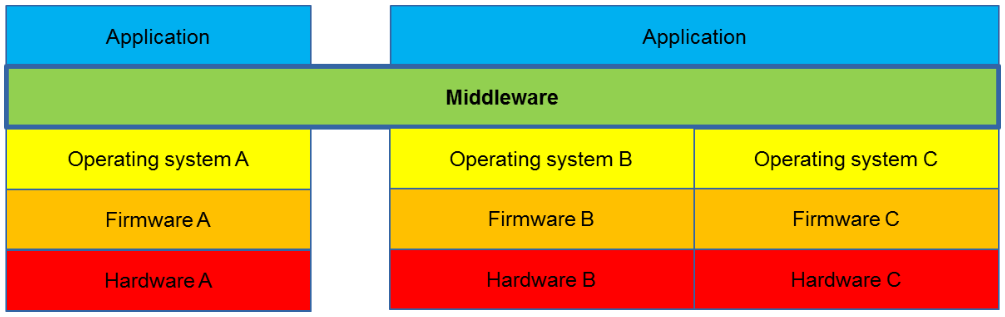

Figure 3. Middleware location in a distributed system

Since the Smart Grid can be regarded as a distributed system that must cope with most of its defined issues, the importance of having a middleware architecture that a) abstracts the heterogeneity of the appliances present in the Smart Grid, b) provides a collection of services that can be used by every entity making use of them, c) offers a way to access those services from the application layer in a homogeneous way and d) allows interoperability and scalability for all the parties involved in a deployment, becomes capital for the reasonable development and deployment of hardware, services and applications.

1.2. Objectives

There are several objectives that have been set for this thesis; all of them are related to a great extent to middleware as a software component of the Smart Grid. They have been listed as follows:

1. ***Providing a study on the state of the art with regards to existing middleware solutions for the Smart Grid where their strengths and weaknesses are acknowledged and assessed***. It will be made clear during this thesis that the existence of a semantic middleware in this application domain makes possible not only the abstraction of hardware heterogeneity (thus easing the integration of Renewable Energy Sources –referred to as RESs- in the Smart Grid) but also adding other services (semantic capabilities, security, context awareness, service composition) that result in the improvement of the information and applications that can be included at higher levels. However, an evaluation of the existing middleware solutions for the Smart Grid must be carried out in order to know how optimal the current solutions are. This step is required so as to know how the current status quo with regards to the scope of the thesis is looking like, and contributions can be focused in the areas where they are most needed, which have been identified as open issues to be dealt with.

2. ***Contributing to the standardization of design and implementation of semantic middleware architectures for the Smart Grid***. A proposal for a standard in semantic middleware architectures for the Smart Grid is put forward in this thesis; it has been called *Common Middleware Architecture* (CMA). The contributions that are made by CMA are mostly focused on a) the establishment of a collection of functional and non-functional requirements that can be regarded as mandatory for a good design of a semantic middleware architecture for the Smart Grid, b) listing a set of software components that must be present in any semantic middleware implementation that is planned to be used in this application domain, c) how these components must interact with each other in order to guarantee the expected services that a middleware for the Smart grid should provide and d) the steps to be carried out during implementation activities where software components are codified. In a more detailed way, contributions have been made in order to establish a collection of characteristics, software modules, interfaces and technologies that can be applied for efficient middleware implementation activities. These contributions are based on the developed middleware works that have been made in different European projects. While these works have been done following the waterfall software development cycle (analysis, design, implementation, testing and maintenance), it has been combined with incremental prototyping for the development works that have been carried out in the middleware architecture implementation. Although CMA could be used as a standard for the design and implementation of middleware architectures for the Smart Grid, its contributions can also be incorporated to any other standard that resembles the functionalities that have been put forward here and enjoys a higher level of popularity. Furthermore, the addition of a software layer that is performing several complex actions when dealing with the data requested that has to be transferred among the different components of a system could suppose that the pieces of equipment where it is installed would be hindered by its inclusion and demand of computational resources. Nevertheless, performance results show that including middleware in architectures providing service registration or composition does not affect the performance of a deployment, with the added befits of including services that can be utilized by an end user or a prosumer.

3. ***Proving that a semantic middleware architecture is a key element to create business models where new actors (especially prosumers) can join a new scenario where energy access and trade are democratized and more distributed than before***. Since middleware is expected to be a major agent to use in order to integrate distributed, small-scale RESs, it must be proved that its inclusions adds to the trend of including prosumers in this application domain that will make energy access an easier procedure that will enhance energy availability and competitiveness in the energy markets.

This thesis describes the latest progress made regarding middleware architecture designs that become implemented as middleware solutions for the Smart Grid. Again, it has to be noted that from the software engineering point of view, the Smart Grid can be regarded as a distributed system that will be enhanced by the usage of middleware, so it is very convenient to have an implementation like this. It is due to those presented progress works that a proposal for an intermediation architecture for the Smart Grid application domain can be formulated with a high level of detail. Taking software development into account, along with the environments where the resulting work is deployed, the following core procedures were used for the implementation:

1. Describing the actions used to formalize, design, implement and validate a middleware architecture that has been used with success in a Smart Grid-like development. This is an area of knowledge where inclusion of ICTs (let alone middleware) is often implemented in a very poor way, or ignored altogether. Consequently, advantages that can be offered by ICT (a variety of services, applications for end users, hardware seamless integration, etc.) are not fully developed for power grids. The work that is going to be presented here shows that it is not only feasible, but also desirable, including a middleware layer within a Smart Grid, as it offers several facilities that greatly improve its usability. To name but a few, heterogeneity abstraction, interoperability among pieces of equipment, software and hardware scalability, security or context awareness can all be offered by middleware. In addition to that, Smart Grid-specific functionalities such as demand Response or Demand Side Management can be provided in an easier, more efficient way.
2. Describing the common combined features with regards to the efforts done in this area of industry, so that the model previous referred to as Common Middleware Architecture (CMA) can be put forward as a way to offer fixed interfaces, services and data units to be implemented. This proposal describes a) the software modules that can be regarded as mandatory for a functional middleware architecture, b) where those modules are located, c) how they should be interfaced and d) how they should be implemented into a middleware solution. In addition to that, the model presented here might be usable for some other application domains with technical similarities, such as the Internet of Things (IoT), and generally speaking, any kind of Cyber-Physical System that implies several pieces of heterogeneous hardware cooperating with each other by sharing information. In this sense, the ideas and principles described in this thesis have been ported to another European project named SWARMs (*Smart and Networking UnderWAter Robots in Cooperation Meshes*, [20]) that has similar challenges regarding interoperability of Autonomous Underwater Vehicles (AUVs).

Figure 4 offers an illustrated example of this idea for an IoT-based development: despite having a small collection of features specific for each of the environments where middleware is deployed (such as the IoT), a significant amount of them is likely to keep reappearing. It is only natural for this to happen, as there are a group of functionalities that must be implemented on a constant basis.

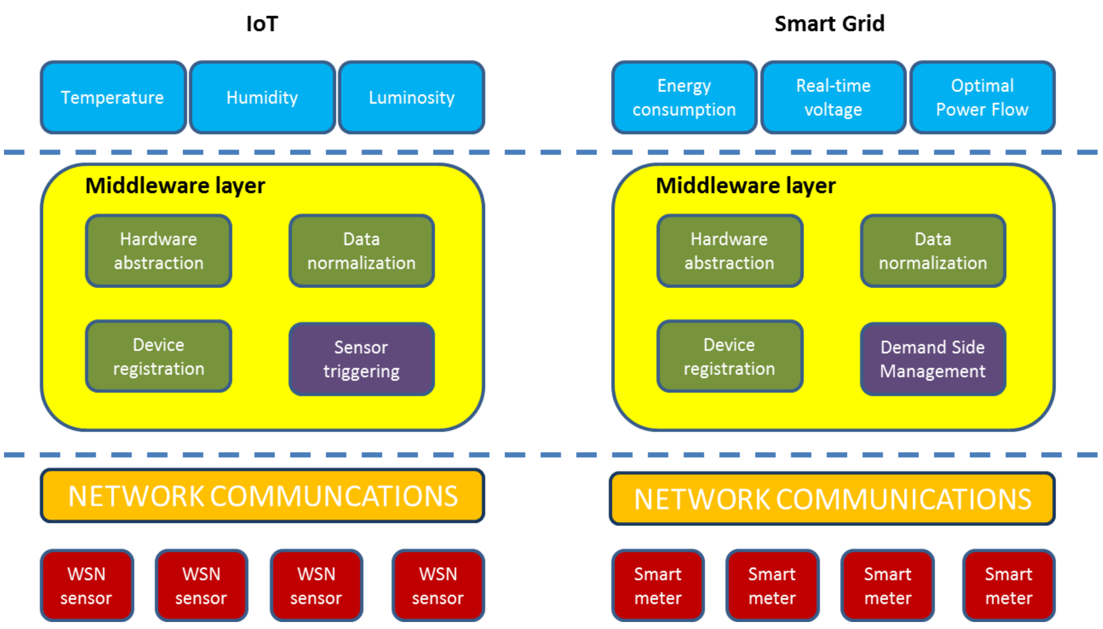

Figure 4. Comparison between common elements in the IoT and the Smart Grid

As a starting ground to fulfil these goals, there are several activities that have been conducted in this thesis to have an accurate grasp of the current state of the art in several characteristics where middleware architectures are implemented:

1. The state of the art in middleware solutions for the Smart Grid has been studied and a taxonomy has been applied to classify the possible middleware solutions that can be found as a way to locate the main topic of the thesis (that is, semantic middleware architectures for the Smart Grid) within the possible software developments related to middleware and distributed systems.
2. A state of the art regarding the tools that have been used for the middleware implementation works used to validate the proposal for a Common Middleware Architecture. Since the most prominent of them is the open source Enterprise Service Bus architecture that was used in the validation scenarios, the most profound study for the software tools has been carried out here.
3. Last but not least, the potential for business models in middleware architectures has been studied as well. This is a matter of major importance because the commercial point of view is a pivotal element in the diffusion and acknowledgment of the proposal for a Common Middleware Architecture. Currently, business models for middleware exploitation involve a reduced set of companies with proprietary solutions that extend the closeness of their development works to the systems where they are used. Additionally (or sometimes as a consequence), middleware is often regarded as a nuisance rather than an opportunity to provide profitable

services. Effective exploitation of middleware under different perspectives so that it will be a decisive agent for business models where prosumers can play a prominent role is described as part of the thesis.

The most prominent pieces of information described have been presented to the research community in varied forms, such as contributions for European research projects, scientific papers or software implementations, as they will be shown in a more extensive manner in the following sections.

1.3. Thesis framework and background

The concepts displayed are done according to two major parameters that must be taken into account: on the one hand, the location, from a software and layered architecture point of view, of the research that has been done during the period of time required to complete the assignments related to middleware. On the other hand, the experience acquired in research European projects from different call (ITEA-3, ARTEMIS, FP7, ECSEL).

1. If the different layers that have been previously described are taken into account, the research activities will be located in the middleware layer and slightly extended beyond the boundaries of this layer both upwards (to the application layer) and downwards (to the network layer interconnecting the hardware components). The reason to include these parts is due to the fact that interfacing the middleware layer, along with the technologies used to develop those interfaces, has to be taken into account when designing the different software modules to be programmed in the middleware solution. In a more accurate manner, the background of the area of the research done is in Figure 5.

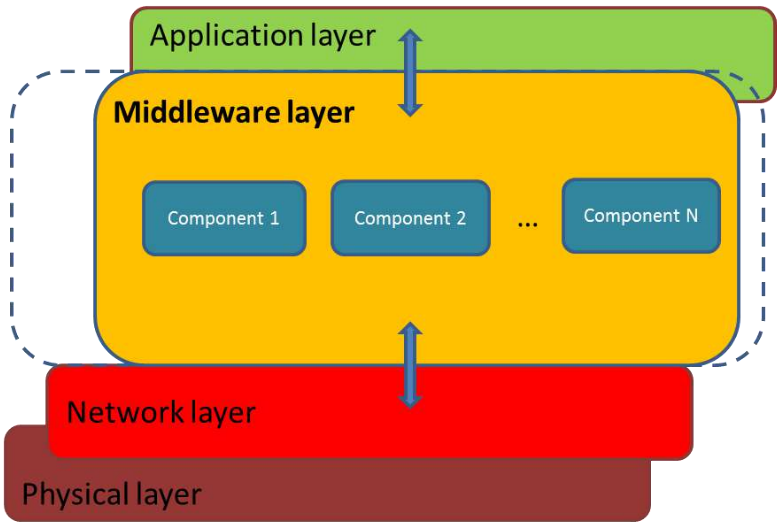

Figure 5. Area of interest of the thesis

2. At the same time, the implementation works done on this thesis have been done within the framework of European research projects. This is a pivotal feature of the presented research, as it has been validated by means of the scenarios that have been used in those

projects. The most prominent contributions have been done in the ARTEMIS-funded e-GOTHAM project (Sustainable-Smart Grid Open System for the Aggregated Control, Monitoring and Management of Energy [21]) and in the FP7-funded I3RES project (ICT-based Intelligent management of Integrated RES for the smart grid optimal operation [22]), which have obtained overall satisfactory results with acknowledged positive contributions regarding research and development in many different areas, including middleware solutions. Some of the ideas of middleware for distributed systems (hardware abstraction, information distribution, generic API for the application layer) were also implemented in the ITEA-3 project called LifeWear (LifeWear: Mobilized Lifestyle With Wearables [23]). Logos of these projects can be seen in Figure 6. In addition to that, contributions regarding middleware have also been done in ITEA-3 Web of Objects project [24] and in the currently undergoing ECSEL SWARMs project (Smart and Networking UnderWAter Robots in Cooperation Meshes [20]). e-GOTHAM and I3RES are projects strongly focused on the Smart Grid, whereas LifeWear and WoO deal with the Internet of Things and SWARMs with underwater robotics.

e-GOTHAM
Sustainable-Smart Grid Open System for the Aggregated Control, Monitoring and Management of Energy
i3RES
intelligent energy integration
life wear
SWARMs)))

Figure 6. Project logos where middleware-related research activities have been carried out

The experience acquired with these European projects also provides a good starting point for future works that will be done regarding middleware architectures in the future.

1.4. Dissertation outline

This thesis has been structured in six different sections, each of them dealing with different characteristics:

1. Chapter one introduces the most high-level features of the thesis, such as the background where it is developed, the main issues that are tackled and a first approach to the development works that have been carried out.
2. Chapter two displays a study on the state of the art done on the major features that imply the research works done, that is to say, middleware architectures for the Smart Grid, tools

for their implementation and business models for their exploitation. Open issues and challenges to be tackled are inferred as a result.

3. Chapter three includes the dissertation and a complete description about the main contributions done by this thesis: an outline of the Common Middleware Architecture for the Smart Grid is put forward, the lower and upper layer technologies used so that hardware and software components will be interfaced with ease, the description of the features that are used for implementation, and the functional and non-functional requirements that resulted in this proposal.
4. Chapter four deals with the validation scenarios used to prove that the proposal is capable of, showing good results in terms of performance both in the intermediate efforts and the final ones.
5. Chapter five describes the conclusions reached with the works presented in this thesis and puts forward some future works that can be carried out in the future. In addition to that, it shows the publications done in indexed journals (using the Journal Citation Report, JCR, as the chosen index), international conferences and other dissemination works done. Awards received are also mentioned.
6. Chapter six shows a list of all the references that have been used to complete this thesis.
7. Two appendixes have been added afterwards; one is used to offer a short description of some of the implementation works that have been carried out and the next one shows part of the API that was generated during the e-GOTAHM project, as it is directly related to the capabilities expected from CMA.
8. A table of acronyms closes the manuscript.

2. State of the art

2.1. Introduction

The main motivation for having a state of the art is that it is mandatory to have a good grasp on the status of the technologies and solutions that are being studied, as it can be considered as the starting ground for a researcher to become aware of the open issues and challenges that can be found in the corresponding application domain. Taking into account the existing middleware architectures for the Smart Grid, an accurate idea can be obtained regarding the available solutions in terms of research and marketable solutions. Middleware architectures for the Smart Grid usually involve their deployment in pieces of equipment of different features and capabilities, along with the set of services that can be offered to the prototype, demonstrator or permanent location where the Smart Grid is located. It is likely that the middleware will be added to a specific smaller structure, called *microgrid*, which basically represents a lower-scale Smart Grid (often covering a small, yet still sizable area, such as an industrial complex or a village [25]) that can be connected or disconnected to another larger entity (thus behaving in an island mode if required).

Other major aspect to be considered is the kind of software architecture that is used in the distributed model. Taking into account the existing solutions, one of the most suitable ones to be used is an Enterprise Service Bus or ESB. An ESB can be regarded as a model for a software architecture capable of transmitting messages throughout the entities connected to the bus used for data interchange, or as mentioned in [26] "*a service computing platform based on event driven mechanism. Its task is to offer an integration management platform to the Service Providers (SP) and the Service Consumers (SC), which decreases the coupling degree of SP and SC and eliminates the mess relationship between SP and SC*". Considering that an ESB is mainly used as a software abstraction layer that withholds the differences among software applications, it can be easily used as a way to locate all the middleware services and components. While any other software architecture capable of embedding software components is still suitable, there are several more reasons to have come to the conclusion that an ESB is one of the best solutions for middleware implementation. The most prominent ones are as follows:

1. Service storage: when an Enterprise Service Bus is installed and run, locations are created as a way to store in a non-volatile way the different software packages that are created. Those software packages (which are often referred to in ESB terminology as bundles) can afterwards be transferred to another machine –which might even be of the same family of ESBs solutions but not the specific same one-, thus enabling system interoperability (since the software packages can be used regardless of the underlying hardware, as long as it is capable of having the ESB working without performance issues). It must be noted that a system that uses an Enterprise Service Bus will therefore be capable of containing a certain amount of services, so it will be used as more than a mere mean to transfer messages among the devices participating in a distributed system.
2. Design: an Enterprise Service Bus has been conceived as a software system that justifies its usage by allowing interoperability among different software-based

applications that can be programmed in different languages [27]. This can be proven by the fact that ESBs have been ported to a variety of systems with different applications, such as healthcare [28] or C4I (Command, Control, Communications, Computers, and Intelligence) systems [29].

3. Middleware concept: middleware architectures´ most prominent functionality (device heterogeneity abstraction) matches very accurately the purpose of having a software tool used to contain all the services that can be collected from the system, regardless of the hardware that is used to obtain them. When installed, ESBs will usually be turned into the embodiment of the middleware that has been designed for the system.
4. Distribution: ESBs make use of a bus to transfer information from one entity to the other. Consequently, there are no centralized locations that contain most of the information when an ESB solution is deployed in a system. This has several advantages, such as not having critical points that would leave the system seriously damaged in terms of performance whenever there is a flaw in the software or hardware use to run it in a specific location, or easing the addition of new peers in an already existing system, such as prosumers equipped with technology focused on obtaining electricity in a small scale (Distributed Energy Resources or DERs).

Overall, an Enterprise Service Bus solution will be therefore used as an intermediation level between applications (that in the scope of this thesis will be based on what can be obtained from the Smart Grid, like Demand Response, Demand Side Management, Optimal Power Flow, etc.) and information resources, such as pieces of information or databases. The justification of this statement will be shown during this chapter of the thesis.

Last but not least, it must also be taken into account that middleware architectures can be used as a way to either support or provide business models that will result in the profitable exploitation of either software solutions or devices that encased middleware. It has been stated that sometimes middleware is often deemed as a lesser evil that has to be dealt with [30], yet the potential for services and applications that middleware can offer in different application domains it is usually not acknowledged. This thesis proves that middleware can be used as a backbone for business related to the Smart Grid, even though distributed and Cyber-Physical Systems will largely benefit from the implementation of a middleware layer that enhances the usability of most of the entities participating in a system of those features (the Internet of Things, Autonomous Underwater Vehicles, etc.).

2.2. State of the art in middleware architectures for the Smart Grid

When surveying middleware architectures for the Smart Grid, it has to be taken into account that they share a significant amount of features with the middleware architectures for other distributed or Cyber Physical Systems, such as the fact that they have software components often included in hardware devices working in a cooperative manner, or how they often use messages to transmit information from one element of a deployment to another different one.

Therefore, there are several features that can be ported to other application domains as well. On the other hand, the variety on different middleware solutions applied to the area of knowledge of the Smart Grid is wide, with different backgrounds and legacy works that have been included, so there are several criteria that have to be taken into account to grasp an accurate idea of how the main challenges addressed by middleware are solved in each of the proposals. Consequently, in order to classify the different middleware proposals that have been used for distributed systems, a hierarchy or a way to identify common features among will have to be established, as the scope of this manuscript attempts to cover different proposals of differing areas of knowledge. Therefore, some characteristics have been established as a way to organize the knowledge that has been included. Although sometimes proposals will be sharing features from each of the features used in the same context, in the end there will be a more prominent one characterizing the proposal. These features are:

1. Service availability. This is used to describe how there are some distributed systems that rely on middleware just as a way to abstract hardware heterogeneity when compared to other proposals that define a more complete set of functionalities for middleware (high level interfaces, inner services, etc.). From a layered software point of view, this feature defines how "thick" a middleware proposal is, depending on the number of services that have been implemented.
2. Computational capabilities of deployed hardware. The hardware capabilities of the systems running middleware must be taken into account when the solutions are being evaluated, as they will have to work with a higher level of optimisation if they are installed in low capability devices that will impose strict non-functional requirements for its usage. It can often happen that middleware, being a software development capable of abstracting hardware features, will be deployed in devices of heterogeneous software capabilities. This particular case will also be taken into account, as using one kind of device over another one should be dictated by the actual needs of the middleware architecture.
3. Message coupling level. Depending on the specific needs of the platform where middleware solutions are deployed, it could happen that different degrees of message coupling are required. When describing message coupling, it must be understood as the amount of time that typically takes place when sending a message and receiving it at the other end of the communication. Requiring a longer time to do the transmissions does not necessarily imply that the middleware solution is worse, as it could happen that it has been designed with a high decoupling level.
4. Middleware distribution. While middleware is usually conceived for its usage in distributed systems, there are different degrees of distribution that imply the existence (or absence) of centres of information in the system. The level of relevance when they are present will be considered by means of this feature.

Overall, these features will be used as the backbone of the classification used to characterize the most prominent middleware proposals for the Smart Grid that have been researched. Each of them can be conceived as an axis that will showcase "minimum" and "maximum" values that, depending on the performance and the overall features of the proposal, will result more or less optimized for the usual functionalities expected from middleware. Thus, the chosen criterion will be further divided in intermediate levels that define their partial status and are encased within two values that represent the extreme, end values. Another way to conceive this is as a matrix where all the features will be categorized as a collection of rows and each of the intermediate states of the features will be located as part of the columns of the matrix. Both depictions will be further described in the next subsection.

2.2.1. Service availability

The quantity and extension of the services that have been included in the architecture will determine the capabilities of the middleware designed and the level of responsibilities assigned to it. It is not uncommon that a distributed system using low capability devices as a way to extract information will rely on services located in the middleware to perform operations that would otherwise be made at the applications or in the hardware components. On the contrary, sometimes most of the services will be offered outside the middleware and it will be deployed just for message interchanges or as an access layer for other parts of the deployed system. Four different levels of service availability have been defined to be included in the developed middleware taxonomy:

- Abstraction middleware. This kind of middleware offers the required set of operations to perform hardware abstraction, so that all the data will be presented to the application-based, upper levels in a homogenous-looking way that will make it easy to be utilized by end users and application developers.
- Intermediation middleware. Aside from hardware abstraction, this type of middleware offers an additional sublayer used to provide access points to the middleware for the applications that are located right above it. This is done so due to the fact that the applications may be too light or have too few software resources to run several procedures related to special characteristics (machine learning, big data) that will require them to locate part of their functionalities in the underlying middleware.
- Message-Oriented Middleware (MOM). In addition to the former capabilities, these middleware solutions are capable of interchanging messages with relevant content regarding the information that is transferred throughout the system, thus adding some knowledge to it. The main concept behind MOM is that messages will be interchanged among the participants in the distributed system, regardless of their location or the application iteration that are running in a particular machine [31].
- Middleware architecture. This middleware solution is characterized by offering a wide set of services that are of common usability for the system, ranging from security to semantic capabilities. It is the most complex and complete possibility available according to the studied middleware designs and implementations, and relies on a structure to encase the services and features of a software-based system. One example of this concept is Enterprise Service Bus architectures [32].

According to the description previously offered, the axial representation of this feature is as shown in Figure 7. Note that the subjacent feature that establishes the existing levels is the service availability in the middleware proposal.

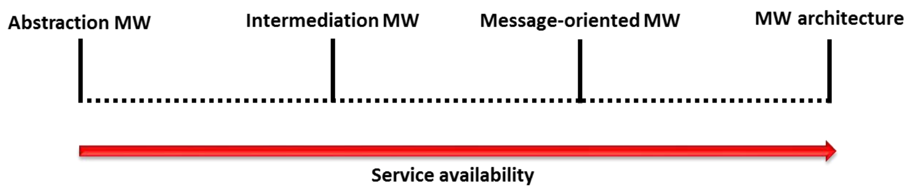

Figure 7. Levels of service availability for middleware

2.2.2. Computational capabilities of deployment hardware

Even though hardware will be an underlying element that middleware will abstract in order to withhold its complexity for end users and clients alike instead of part of the middleware itself, it is something that must be taken into account when designing it, for it will determine the kind of services that can be run in the system. When all is said and done, computational capabilities must be born in mind when designing the complexity of the services to be used, as the processing power provided by the hardware will determine if a service can be executed locally, in a distributed manner, or cannot be offered to higher levels at all due to its excessive workload demand. The levels that have been defined for computational capabilities are:

- End user domain devices. These are pieces of hardware that are usually included in the Smart Grid as equipment mounted in an end user´s dwell or facility. The most common ones are Advanced Metering Infrastructure (AMI), which sometimes can be used and deemed as if it was part of an Internet of Things development [33] or home batteries for energy storage, which are becoming increasingly popular because of the energy they can provide when buying it from the electricity markets at a particular moment is disadvantageous [34]. The capability of these end user devices to have software components installed in them is usually acceptable, but issues related to Intellectual Property Rights (IPRs) or security may happen when proprietary solutions are used.
- Aggregator domain devices. These are the devices that will be used by the aggregator/retailer present in a deployment to, depending on how smart the power grid is, either trade with electricity among different clusters of users (in case the aggregator has the capabilities to do so) or just offer it to be sold to the end users. Commonly, these devices will imply some hardware with a database in them to store information about end user profiles or energy scheduling algorithms used to determine when energy should be purchased [35].
- TSO/DSO domain devices. These are devices that are used in a Smart Grid, but are not accessed by end users or the aggregator, as they are usually owned by the DSO or the TSO; if something, they will be accessed by the technicians and engineers participating in their deployment. Two examples of this kind of equipment are Remote Terminal Units (RTUs), used for functionalities such as executing demand response programs between the end users and the DSO [36], and Phasor Measurement Units (PMUs) focused on synchronizing the measurements on an electrical grid for monitoring and control purposes through the

power grid [37]. As it happened in previous cases, issues related with Intellectual Property rights are prone to happen as well.

- Power plant domain. This term encases the different hardware devices used in the power plant management, which may or may not have middleware software components installed in them, depending on how services have been distributed in all the other hardware locations. The pieces of equipment present here usually require large computational capabilities or generate large amounts of data for power grid management (big data, [38]) or execute complex algorithms for overall knowledge inference (machine learning, [39]).

Considering the description of the different available levels shown before, Figure 8 shows them in a more graphical manner.

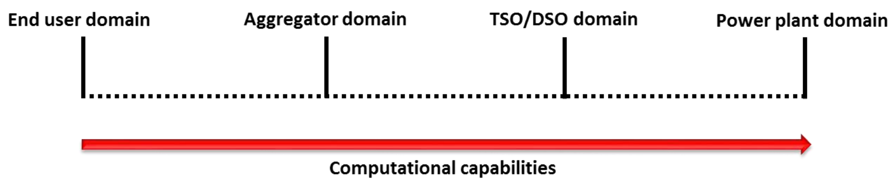

Figure 8. Levels of computational capabilities for middleware

2.2.3. Coupling level

This feature must be understood in the context of this manuscript as the level of interdependence between the element that produces data when a communication is established and the element that, at the other end, consumes the data that has been served to the system for any kind of purpose. Coupling level is closely linked to the need of providing the data as soon as it has been produced in the system. The four levels that have been defined for this feature are as follows:

- Real-time. In this case, data are served to the consumer as immediately as possible, to the point that it is virtually assumed that no delays are acceptable. Real-time systems are often prone to severe constrains in the requirements that are regarded as such [40].
- Client/Server. In this case, the system follows a pattern where data are located in one end of the communication (the server) and are retrieved by another end making requests to the server (the client) in the typical fashion used in distributed systems [41]. This model used for interchanging information is the one used in widespread systems such as the Internet and is followed by many middleware proposals as well.
- Polling. Under this paradigm, data are provided to a storage location where it is kept until it is retrieved by the data client that is expected to consume it [42]. As it can be inferred, the time required for the user to retrieve the data is less important, as long as the information is still available.
- Publish/Subscribe. Under this paradigm, several entities subscribe to a collection of services that can be offered by the system, so that whenever data is available for those

services they will be retrieved by the subscribed elements. The main difference between polling and publish/subscribe is that when an entity subscribes to a service, it will receive the data without any further action, whereas under a polling paradigm the entity that needs the data will have to actively demand it to the entity that stores it.

As it happened with the other two criteria, there is a progression between the lowest message coupling level and the highest one, as shown in Figure 9.

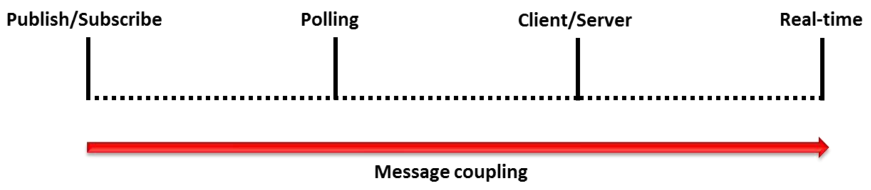

Figure 9. Levels of message coupling

2.2.4. Middleware distribution

This feature assesses the level of distribution that has been designed for each of the solutions. Four different levels have been defined by bearing in mind how many devices will have significant parts of the middleware installed:

- Fully centralized. The middleware solution has been installed in a single device (which implies that it will be powerful enough to run it). Although there might be other devices connected to the one with the middleware installed, the solution is still installed in a single one.
- Mostly centralized. The middleware has been distributed among several devices, but there is still a more prominent centre where most of the services or the system intelligence runs.
- Mostly decentralized. Most of the logic and the prominent features are located on the periphery of the system, despite the existence of a central node used to transfer information from one part of the middleware solution to the other.
- Peer-to-peer. No central elements are available in the solution; all the interchanged data is done so in a fully distributed manner. This is the paradigm that, for example, has been used the most for file sharing from a historical point of view [43].

Figure 10 shows the different levels that have been set for middleware distribution, ranging from next to no distribution to a peer-to-peer paradigm.

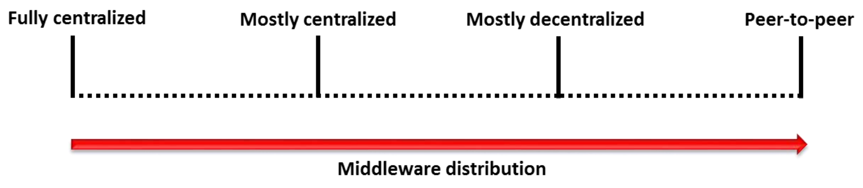

Figure 10. Levels of middleware distribution

2.2.5. Taxonomy for middleware in distributed systems

The previous assessment of the criteria put to a use to classify and have a holistic perspective of the middleware solutions can also be taken into account from different perspectives. A direct one, where all the internal categories for each of the criterion become represented, has been used in Figure 11. It effectively implies that a taxonomy has been introduced as a result of the study on the state of the art of middleware architectures for the Smart Grid.

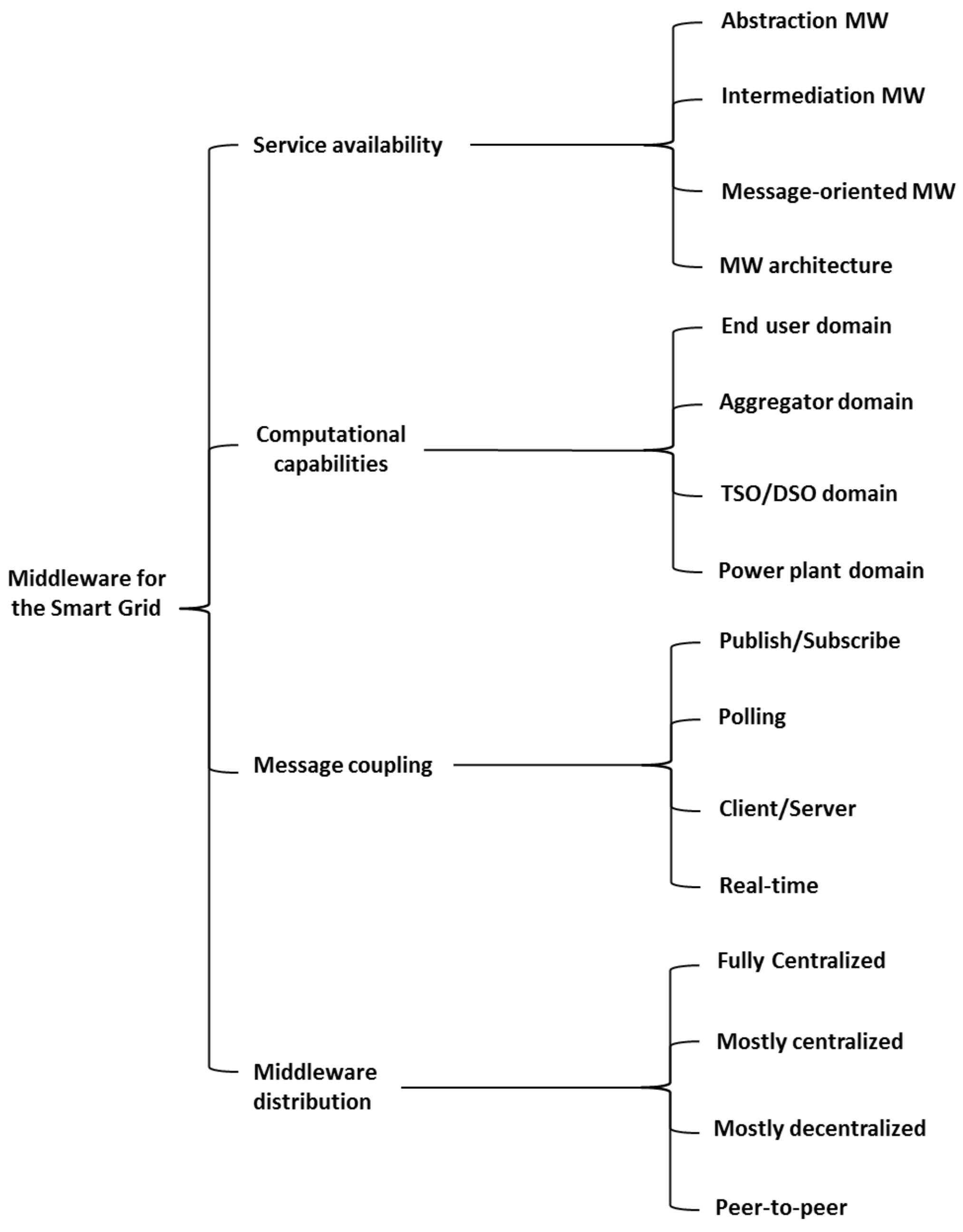

Figure 11. Taxonomy for middleware classification

In addition to that, there is another way that can be used to represent the taxonomy; rather than having a separated branch for each of the features, they can be placed altogether in a matrix where each of the rows will be occupied by the feature and each of the columns will be made up by the intermediate levels of every characteristic of the ones that are represented. This latter representation is more appealing from the mathematical point of view, as each of the features can be located as one element matching a row and column position, as depicted in Figure 12.

$$\textbf{Middleware Solutions for the Smart Grid} = \begin{pmatrix} \textbf{Abstraction MW} & \textbf{Intermediation MW} & \textbf{Message-oriented MW} & \textbf{MW architecture} \\ \textbf{End user domain} & \textbf{Aggregator domain} & \textbf{TSO/DSO domain} & \textbf{Power plant domain} \\ \textbf{Publish/Subscribe} & \textbf{Polling} & \textbf{Client/Server} & \textbf{Real-time} \\ \textbf{Fully centralized} & \textbf{Mostly centralized} & \textbf{Mostly decentralized} & \textbf{Peer-to-peer} \end{pmatrix}$$

Figure 12. Matrix for Smart Grid middleware taxonomy

If the taxonomy is born in mind the way it has been depicted in the previous figure, the accurate definition of the features of each of the middleware architectures could be regarded a matter of choosing features that will match a collection of four positions in the given matrix, one for each row (feature established as a way to define the kind of middleware), so that the optimal middleware solution will be defined as the result of a mathematical equation. If, for example, a solution where a) a middleware architecture for b) end user devices based on c) a Publish/Subscribe paradigm that d) requires a peripheral information processing procedure but still has a centre to perform certain functionalities, then the solution could be described as follows:

Smart Grid Middleware = Service Availability (element no.3) + Computational Capabilities (element no. 0) + Message Coupling (element no. 0) + Middleware Distribution (element no. 2).

Therefore, it can be represented as:

$$SGM = SA\ (3) + CC\ (0) + MC\ (0) + MD\ (2) \tag{1}$$

In this way, a precise idea can be obtained of the most prominent aspects of a middleware solution that has been conceived for the Smart Grid. In addition to that, once the characteristics of a particular architecture have become clear, an assessment of its advantages and disadvantages can also be done by considering how middleware functionalities are performed in each of the presented proposals.

2.2.6. GridStat

The proposal elaborated by Harald Gjermundrød, et al. [44] focuses on several aspects inherent to middleware and distributed systems: it has been conceived to be used by scattered pieces of hardware involving Smart Grid deployments, Quality of Service (QoS) capabilities and a collection of functionalities within its very architecture. It was firstly presented as a way to distribute timestamped, time-synchronous data from Phasor Measurement Units [45] and further developed into a way to support RPC (Remote Procedure Call) mechanisms to use its own QoS semantic features [46]. The main purpose of this middleware proposal is creating a framework where data can be interchanged in a distributed manner under a Publish/Subscribe paradigm. In this way, messages will be transferred from one entity to another by mean of intermediate hardware appliances and software components. The following information can be inferred for each of the previously formulated criterion:

Service availability: while GridStat is regarded as a middleware framework by its authors and is referred to as an architecture, its functionalities are aimed at transferring information among different locations rather than providing any extra service, so it fits better a Message Oriented Middleware type of service availability. The authors claim that it offers secure, robust and flexible data communications that provides Quality of Service features, such as QoS-managed multicast. One of the most notorious features of the architecture is that it is divided in two different levels that are referred to as planes: the upper plane is the *management plane*; its main purpose is establishing the forwarding rules of the information sent. The other plane is called the *data plane*; it transfers data from the publishers to the subscribers, regardless of their location. Management plane uses *QoS brokers* to determine where information is sent in the data plane. QoS brokers in touch with the management plane (called *leaf QoS brokers*) directly manage the data plane *clouds* (a set of routers with the same resource management and cyber security policies) where the *Status Routers* are located; the latter have a flat organization within the clouds and forward the incoming data to the suitable communication link. Figure 13 depicts the appearance of the architecture.

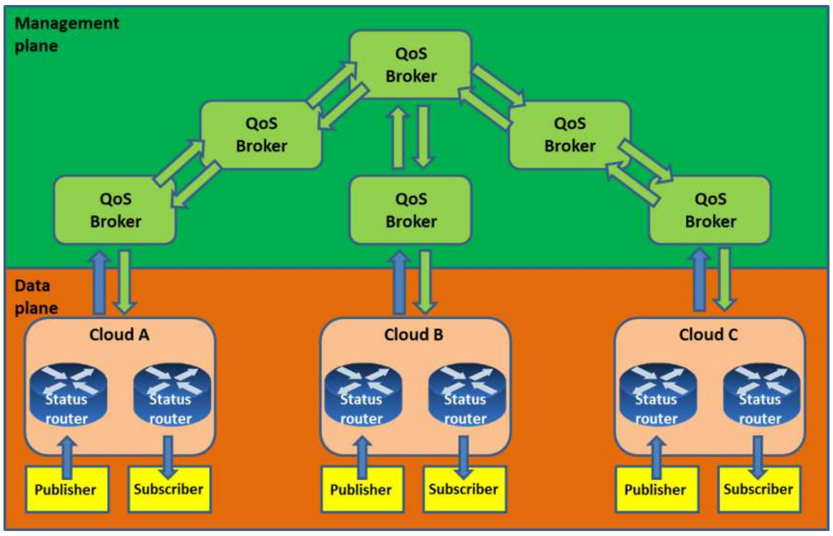

Figure 13. Gridstat proposal, as described in [43]

Computational capabilities: The proposal is primarily aimed to provide substation automation (which implies that the software components can be located in the TSO/DSO infrastructures, as both entities have substations in the power grid). However, it is not explicitly mentioned that it cannot be equipped elsewhere, as the software elements that are described, such as QoS brokers or publishers and subscribers, are not necessarily located in one specific kind of hardware. The implementation that was used to provide experimental results uses Java as programming language (even though a C-written version, which should offer a higher performance, were under development at the time the proposal was introduced to the research community), whereas the hardware used implied Dell Power Edge 1750s servers for data transmission, Fedora 2 Linux as the operating system and a Java 2 Platform standard edition, so the requirements were more than acceptable to run the middleware architecture.

Message coupling: GridStat relies heavily on a Publish/Subscribe model in the data plane to send and receive information among the different entities where it is installed. This is clear from what has been shown of the data plane, which implies groups of routers receiving information from publishers and forwarding it to subscribers that will be dependent on either the same cloud or a different one (thus requiring forwarding commands from the management plane). Along with Publish/Subscribe, some other paradigms have been used as well: it is mentioned that Common Object Request Broker Architecture (CORBA), Client/Server communications are used for command interactions interchanged among the entities of the management plane.

Middleware distribution: The proposal offers a prominent degree of decentralization, but rather than following a peer-to-peer model, it has a hierarchy that keeps high level management decisions in an upper location. It is mentioned in the proposal that the management plane must readapt the network to changing conditions, like power system configurations or network failures, so a degree of control is given to some entities that others do not have.

If the matrix for the middleware in the Smart Grid is considered, the proposal can be defined as:

$$SGM = SA\ (2) + CC\ (2) + MC\ (0) + MD\ (2) \qquad (2)$$

Advantages of the proposal: GridStat offers a detailed framework not only from the theoretical but also from the practical point of view, as there are implementation works to prove the design that has been conceived by the authors and the performance results show the feasibility of the proposal. The fact that it does not require many computational resources to have it running and its significant distribution degree are also appealing, as it can be used in different low capability devices the Smart Grid usually involves, such as smart meters.

Disadvantages of the proposal: The proposal does not have a collection of services that can be used by applications; instead of that, GridStat seems to have been conceived for the mere transmission of data one level above network capabilities. Even though security is mentioned several times in the proposal, not enough detail is provided (such as how access is securitized or whether cryptography is used). Lastly, important features used for middleware and information enhancement (ontologies, information models) are not mentioned in the proposal.

2.2.7. Service-Oriented Middleware for Smart Grid

L. Zhou and J. Rodrigues put forward a proposal that puts its stress on services for the Smart Grid and how they get to be applied in a deployment [47]. The authors claim that they have developed an efficient and integrated middleware solution for heterogeneous services of the Smart Grid, aiming to obtain a high level of software sustainability and stability. The idea of offering efficient services in the proposal is a major one for the authors, as they consider that serving for power allocation and consumption is closely linked to the idea of service applied to a middleware solution for the enhancement of the power grid. The proposal deals with one usual problem of middleware solutions, that is to say, the difficulty to be reused from one development to another, due to the different nature of the services that are requested in each application domain. Consequently, the service-oriented middleware has been decoupled according to several application-based functionalities. Its most prominent characteristics are:

Service availability: As it usually happens with middleware, this proposal is divided into several layers that split the functionalities in different levels, so that they will be accessed and used in different ways. As it has been displayed in Figure 14 there are three levels: a *user part*, a *control part* and a *transmission layer*. The user part is responsible for providing both QoS and Quality of Experience (QoE) for the end user (which usually determines bandwidth, delay, reliability or jitter parameters); it also takes care of scheduling flexibility to offer the best available QoS possible. Secondly, the control part connects the user part with the transmission layer; its functionalities are the ones more typical of a middleware solution, as it is expected to handle the different devices interoperating in the system and transferring information via transmission layer while maintaining an acceptable level of QoS and QoE in the user part. Lastly, the transmission layer is regarded as the foundation of the whole system and uses its services in the Advanced Metering Infrastructure of any deployment where the proposal is used. Data interchanges are carried out both among contiguous levels and between the user part and the transmission layer. Due to the separation of the proposal in different layers, and its differentiated services, it can be regarded as a middleware architecture.

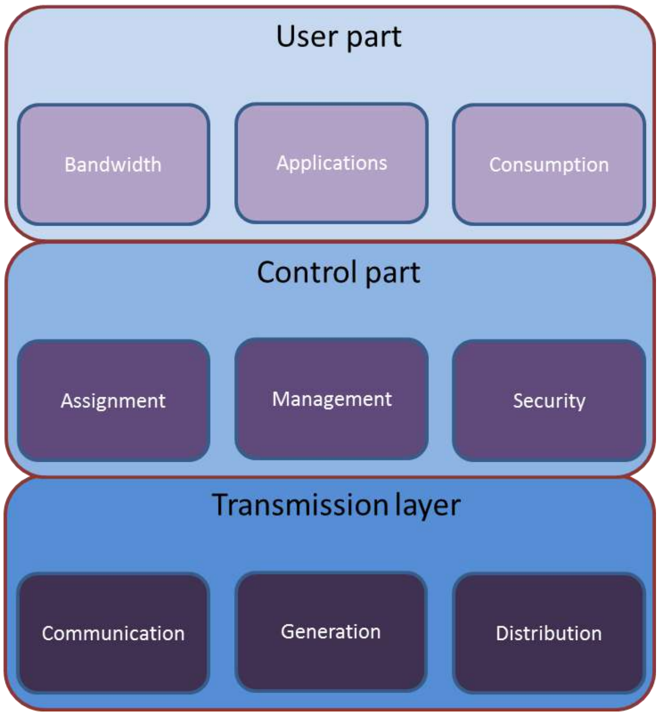

Figure 14. Service-Oriented Middleware proposal, as described in [47]

Computational capabilities: according to the results that have been obtained from the tests, the middleware solution offers a high Mean Option Score (MOS) in four different situations (access control, message transmission, power allocation and service quality) without compromising its performance. As for the tools that have been used, the network simulator ns-2 was used to test the proposal. The collecting node has been modeled with a regular personal computer. Each of the smart meters has been modeled as a node with an ARM processor. It can be estimated that end user pieces of equipment and aggregator infrastructure would be the most likely to receive the developed middleware.

Message coupling: the proposal mentions that they have developed an application access control principle; hence, there is a homogeneous way to access the middleware. Additionally, it is also said that message transmission was tested, so the proposal can be regarded as a Message Oriented Middleware.

Middleware distribution: due to the tests that have been done and the fact that the proposal is expected to be mounted in devices such as smart meters, it can be argued that it is a mostly decentralized proposal, as it will be used in many devices of comparatively low computational resources but still follow the hierarchy to be expected from the Smart Grid. Specific information about this criterion is scarce in the proposal, as simulation tests are provided rather than having the software running in actual machines.

If the matrix for the middleware in the Smart Grid is considered, the proposal can be defined as:

$$SGM = SA\ (3) + CC\ (0\,||\,1) + MC\ (2) + MD\ (2) \qquad (3)$$

Advantages of the proposal: there is a collection of services provided clearly stated by the authors. Apart from that, performance tests have been made in order to offer actual data about the behavior of the middleware solution. Even they have been done in the controlled environment of a simulator, they offer an idea of the capabilities of the middleware solution. Finally, it is also mentioned that symmetric algorithms have been considered for security implementation.

Disadvantages of the proposal: overall, the services that are provided are not that prominent or complex. In addition to that, it is not mentioned how distributed it can be or the devices that would be able to have installed the software packages it is made of. In addition to that, the monitoring system that is referred to in the proposal is mainly composed by elements that, although are used for communication purposes, can also be regarded as part of levels usually assigned to physical and network-based ones (such as Wi-Fi, satellite networks, the Internet, etc.). Overall, few data are provided with regards of the nature of each of the services present in the three layers of the proposal; information about the functionalities of each of them is missing too.

2.2.8. Ubiquitous Sensor Network Middleware (USN)

Zaballos et al. offer their own interpretation about what a middleware architecture for the Smart Grid should look like [48]. It is mainly focused on adapting the framework of the ITU ubiquitous sensor architecture to the application domain of the Smart Grid, which is conceived by the authors of the proposal as a network of Intelligent Electronic Devices, sensors, distributed generators, dispersed loads and smart meters requiring a heterogeneous communication paradigm. It is also mentioned how by using the framework referred to as the Ubiquitous Sensor Network architecture at defining interoperability, along with a Next Generation Network (NGN) as the backbone where the proposal is deployed, end-to-end integration of devices can be achieved. The proposal relies heavily on Internet of Things-like features, such as lightweight transmission protocols (such as IEEE 802.15.4, used for bit transmission among constrained devices like nodes from a Wireless Sensor Network, even when adding security capabilities to it [49]). In addition to that, it also makes use of technologies like Power Line Communication (PLC) or Worldwide Interoperability for Microwave Access (WiMAX).

Service availability: there are several kinds of services that have been conceived, as it can be seen in Figure 15, so the proposal can be regarded as a middleware architecture. There are three different levels closely related to the kind of level they are closest to. The higher level has a directory containing information about the sensor network. Furthermore, it has an open API that can be accessed by the application developers in order to access the middleware architecture. The middle level is used to process either events related to the deployment or the sensing data that can be mined. Finally, the lower level is devoted to interface and monitor the sensor network that is present in the lower levels. Interestingly enough, there is a security manager module that is used in every level of the architecture.

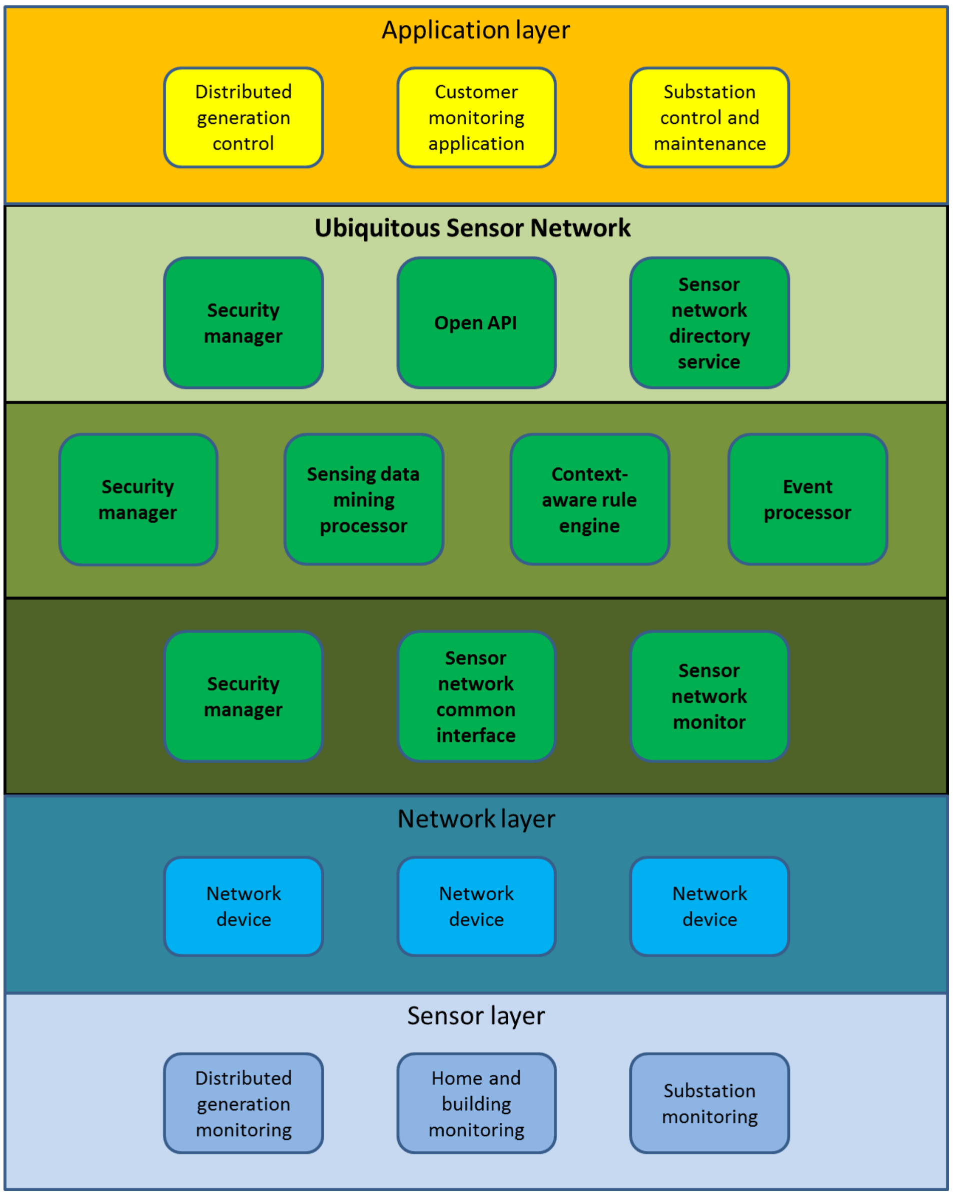
Application layer
Distributed generation control
Customer monitoring application
Substation control and maintenance
Ubiquitous Sensor Network
Security manager
Open API
Sensor network directory service
Security manager
Sensing data mining processor
Context-aware rule engine
Event processor
Security manager
Sensor network common interface
Sensor network monitor
Network layer
Network device
Network device
Network device
Sensor layer
Distributed generation monitoring
Home and building monitoring
Substation monitoring

Figure 15. USN middleware proposal, as described in [48]

Computational capabilities: while the proposal uses sensor-based networks for information transfer, little is said about what kind of devices in the Smart Grid would equip the software components it is made of. It can be presumed that they would be equipped by entities located one level higher than the network one, so virtually any computer, aggregator or TSO/DSO domain equipment would be able to have it installed.

Message coupling: the proposal mentions that the USN application level is a technology platform that can be used effectively in real-time and control functions. Besides, there is an example showing how connection and authentication procedures would be performed that implies a Client/Server communication. It seems that Client/Server requests could be addressed in real-time as a best case scenario, but there is no accurate information about it.

Middleware distribution: as it happened before, there is little information about this topic. Since a network layer is supposed to be installed below the middleware proposal, it is supposed to be either mostly centralized or a mostly decentralized proposal that is able to run in several devices simultaneously.

This proposal can be described as:

$$SGM = SA\ (3) + CC\ (0||1||2) + MC\ (3) + MD\ (1||2) \qquad (4)$$

Advantages of the proposal: the proposal makes use, or at least is compatible with, several different proved technologies used in other levels of the system, such as IEEE 802.15.4 or WiMAX. It aims at being reasonable decentralized.

Disadvantages of the proposal: the proposal tends to mix several different concepts that, due to their inherent features and locations in a system architecture, are not part of middleware (such as applications or Wireless Sensor Networks). What is more, there are no explicit mentions on what kind of equipment would be used to encase the proposal in a Smart Grid or a microgrid. Performance tests, either simulated or in actual appliances, are not shown to get a grasp on how the proposal gets needs solved, such as consumption data or data transport. Overall, the proposal feels more like an Internet of Things development that was ported to the Smart Grid rather than something that was made from the scratch for the latter application domain.

2.2.9. OSHNet (Object-Based Middleware for Smart Home Network)

Sang Oh Park et al. describe a proposal based on interoperability between home devices and Smart Grid-related ones [50]. The proposal is structured in three different layers: the *Application layer* (which offers the home users the proposal is aimed to five different Application Programming Interfaces), the *Library layer* (responsible for providing the necessary information about the present home devices) and the *Network layer* (which provides device interoperability among devices using different kinds of protocols). Thus, it expands its borders further than what is strictly located as the domain of a middleware architecture, which usually relies on data received from the network to be provided to an application; taking into account this idea, the library layer of OSHNet would be the one that is fitting an actual middleware solution, even though it is the network layer the one that is more closely aimed to the usual functionalities of middleware. Home devices characteristics are included in the Library layer as four different kinds of objects: *Control object* (applied to controlling neighbouring home appliances), *Streaming object* (focused on managing input and output data), *Status object* (used to maintain current status of the home devices included in the deployment) and *Function object* (utilized to execute functionalities in home appliances, like powering on or switching TV channels).

Service availability: the proposal can be regarded as a middleware architecture, as it is composed by several levels (referred to in the proposal as *layers*) with different services in each of them. As it was mentioned before, the highest one has the suitable API used to access the application level, whereas the library layer has the aforementioned four kinds of objects (Control, Streaming, Function and Status) used to categorize each of the devices in the home deployment, as well as three modules that are used to invoke services from the proposal: the *Object Management Module* (in order to control the functionalities that are offered by the devices where the solution is present), the *Object Discovery Module* (to gather information about other home peer devices) and the *Connection Management Module* (used to establish, maintain and terminate connections among devices). One lower level, but still within the middleware architecture, the Network layer presents the Virtual Network Adapter (VNA), which is the software entity interconnecting the different devices that are used outside the middleware proposal. Figure 16 displays the most prominent software components of the proposal.

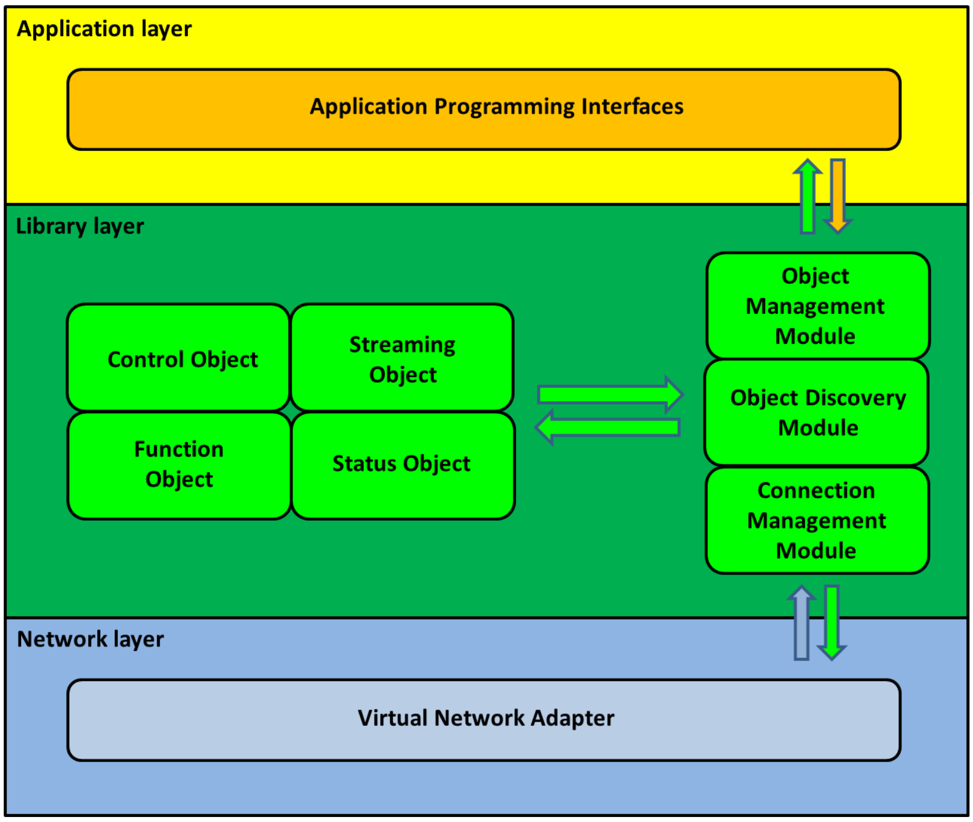

Figure 16. OSHNet architecture, as described in [50]

Computational capabilities: the proposal often refers to home systems as the devices that will equip the middleware solution. In addition to that, the proposal documentation offers screenshots of mobile devices as smartphones, so it can be assumed that End User equipment of the Smart Grid, such as Advanced Metering Infrastructure, would suffice to have the software components installed. The tests that have been carried out involve a set of virtual devices, such as a humidifier, a wind-powered generator, a smart meter, a smartphone, three laptops or an external information center.

Message coupling: since there are user interfaces shown in the proposal, it can be assumed that it is capable of handling Client/Server requests. Information is fed onto those interfaces, so at least they should be able to support some kind of polling, even though the actual way to interact with the simulated environment is not completely clear.

Middleware distribution: it is mentioned that the proposal is expected to be installed in Distributed Energy Resources, as well as in other devices of somewhat lower computational capabilities. In addition to that, there are other pieces of equipment as laptops that are expected to have higher ruling capabilities over the system, so it can be said that the proposal deals with a mostly distributed middleware solution. The devices that are included are supposed to be simulated, though.

According to the matrix that was previously defined, this proposal can be described as:

$$SGM = SA\ (3) + CC\ (0) + MC\ (2) + MD\ (2) \qquad (5)$$

Advantages of the proposal: the middleware architecture that is put forward here contains a collection of services where a group of them are specifically designed to tackle hardware interoperability. In addition to that, simulated devices and user interfaces have been created to have a realistic idea of what can be expected from the proposal.

Disadvantages of the proposal: The proposal also uses confusing names in the sense that there are several levels referred to as “layers” still within the boundaries of the middleware architecture, as the network layer or the application layer. In addition to that, while some services are added to the architecture, those are meant for the internal performance of the proposal, rather than offering a more compelling functionality for end users.

2.2.10. Meter Data Integration (MDI)

Zhao Li et al. offer in this proposal their own conception of what a middleware solution should be [51]. While the proposal is referred to in the proposal as a Meter Data Integration (MDI) solution, it is clear from the text that is supposed to integrate different kinds of AMI in a common deployment, so it is actually performing middleware functionalities to the point that is described as a middleware located between the Advanced Metering Infrastructure and the Distributed Management System (DMS). Some other elements are defined as well: a Meter Data Management System (MDMS) is used as a certain kind of data server and a Meter Data Collector gathers all the information obtained from the deployed Smart Meters. This proposal takes into account implementation details about the smart meters that are either used or manufactured by several large companies, such as Pacific Gas & Electricity or Siemens, so a more realistic approach can be taken when functionalities are designed.

Service availability: the proposal described here makes used of three different levels of information to separate and determine the specific functionalities that will be carried out. Consequently, and taking into account that each of the layers contains different services, it can be regarded as a middleware architecture. The first level contains the services used to adapt the information utilized by the DMSs to the possible transport interfaces used in the deployment. The intermediate layer is the more complex one: there is a temporal database used to verify and translate the collected smart meter data, the Loosely Couple Event (LCE) infrastructure used for message publication and subscription and a MDI monitor used to track the status of the functional components existing in the overall MDI layer. Lastly, the lower level is used to contain all the adaptors used for connectivity with the AMI network. The overall appearance of the services of this middleware architecture is shown in Figure 17.

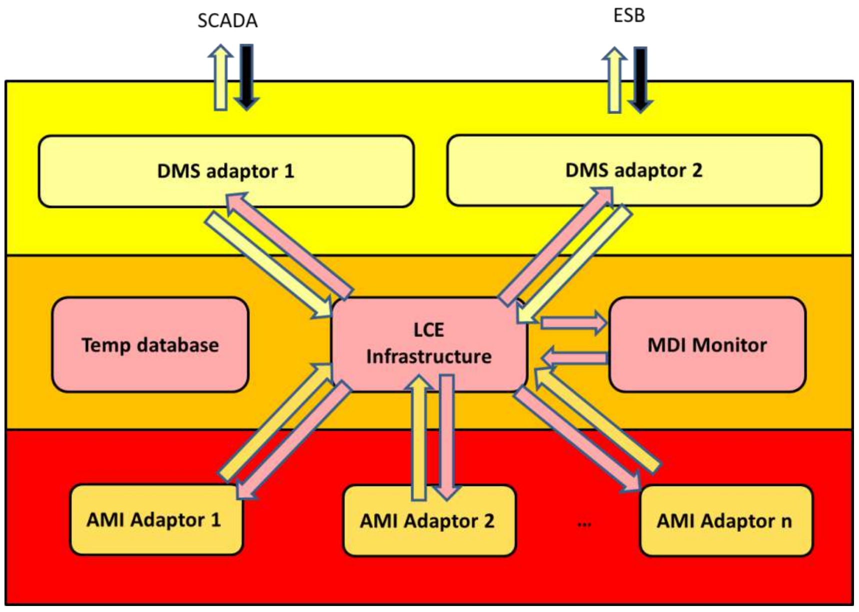

Figure 17. Meter Data Integration Proposal, as described in [51]

Computational capabilities: according to the tests that have been performed on the proposal, the devices used are two simulated smart meters (which use separated pieces of hardware and Windows Server 2003 and 2008 as the operating systems) and another one used as a server containing the MDI layer (it uses a Windows 2008 Server operating system). The proposal is mainly focused on managing Advanced Metering Infrastructure, although taking into account that the MDI components have been located in a different machine and the fact that there are data obtained from the smart meters, arguably it can be expected to be installed in aggregator-located equipment.

Message coupling: the proposal is explicitly described as using a Publish/Subscribe paradigm, as it has been designed to work publishing and subscribing the messages that are transferred by means of the LCE infrastructure. According to the information on this middleware architecture, it is loosely coupled in the sense that message senders and receivers run in different spaces and if for any reason one of them stops working, the other side of the communications will still be operational.

Middleware distribution: the proposal has been installed in a single computer in the tests that have been carried out. On the other hand, it is commanding several devices that have been portrayed as part of an Advanced Metering Infrastructure. Due to these reasons, it has been considered a mostly centralized middleware architecture.

This proposal can be described as:

$$SGM = SA\ (3) + CC\ (0||1) + MC\ (0) + MD\ (1) \qquad (6)$$

Advantages of the proposal: the middleware architecture aims to provide interoperability of different devices that are used as real smart meters. In addition to that, it is clearly justified why the proposal is conceived as one using a publisher/subscriber and the computational resources that are suitable to run it with good enough performance.

Disadvantages of the proposal: as it happened with several previous proposals, there is very little information about how the services can be implemented or any detail about their performance on different programming languages. What is more, services provide not so many functionalities beyond interoperability and interconnectivity; security or semantics, for example, seem not to be supported by the proposal. In addition to that, the information about the tests that have been carried out is referred to as a small group of devices that are simulating actual ones; having more devices that were located in an application domain that involved a microgrid or something alike would have been welcomed, especially when actual smart meters have been studied for the proposal.

2.2.11. IEC 61850 and DPWS Integration

Stjepan Sucic et al. combine in their proposal the usage of DPWS (Device Profile for Web Services, a standard put forward to promote interoperability among constrained implementations of Web services, along with higher interoperability with more flexible client implementations [52]) and an electric standard (IEC 61850, a communication model used for functionalities like setting requirements for device models or the description of the language for communication among substations [53]) typical of the power grid needs that the authors present [54]. Their main idea is that, since the principles of IEC 61850 are defined under a non-Platform Specific or software specific point of view, and in fact, they use a description known as Abstract Communication Service Interface (ACSI) that is not linked to any specific middleware solution, Web services can be used to complement one effective middleware mapping enabling IEC 61850 communications. This mapping enabling is known as Manufacturing Message Specification or MMS, and is also used for distributed transmission of power control [55]. Rather than replacing this technology, the proposal seeks to improve some of its weaknesses; the authors cite its complexity and involving a full OSI stack that makes integration of any distributed solution more expensive.

Service availability: there are several levels that are defined in this middleware proposal as an architecture. One important fact is that, despite their different features, they are all focused on providing facilities to the Web services that are present in it. DPWS is used here, as it is claimed by the authors that it is a technology aimed to provide interoperability for heterogeneous and constrained devices that may have limited and low capability resources. The three different levels that are present are located as it is usual in middleware solutions, that is to say, between the application layer and the network one. The uppermost of them is used for Web service discovery (which provides an addressing mechanism for Web services), Web service Metadata Exchange (defines operations to retrieve device and service metadata, along with datatypes) and Web service eventing (for management of the publish/subscribe mechanism used in the proposal. These capabilities are linked to the intention of using the functionalities that a Service Oriented Architecture (SOA) could use in a distributed system. The middle layer contains all the functionalities related to security (specification on how confidentiality and several security token formats should be used), Web service policy (describes constrains and capabilities of available policies) and addressing (addressing mechanisms for the Web services). Finally, the lower layer contains Simple Object Access Protocol (SOAP) functionalities and metadata facilities like eXtensible Markup Language (XML) schemas or Web Services Description Language (WSDL)-formatted information. The authors claim that the proposal can deal with the requirements imposed by both ACSI (real-time, distributed applications, integration at the device level etc.) and other new control models (Virtual Power Plants or VPPs). These features have been depicted in Figure 18.

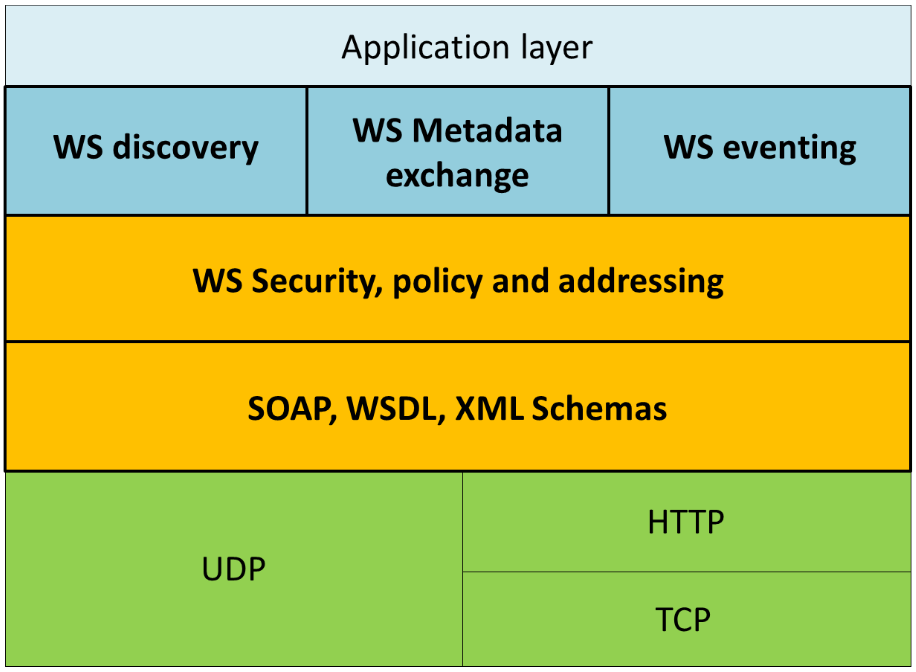

Figure 18. Protocol stack of DPWS, as described in [54]

Computational capabilities: this proposal covers a wide range of Smart Grid devices, as the electric protocol that is enabled is usually implemented in substations, and more recently, in wind power plants or in Distributed Energy Resources. Consequently, it can be said that the proposal covers the area of TSO/DSO domain, as well as the power plant one. Two more aspects of the proposal have to be taken into account too: on the one hand, the Web services can be installed in a myriad of different devices, so hardware to be used can be very flexible. On the other hand, the proposal makes use of the concepts present in DPWS about the devices holding the services (*hosting devices*), along with the devices they are keeping (*hosted services*). It is mentioned in the proposal that it can be used for integration of Virtual Power Plants.

Message coupling: the proposal is mentioned to work with a Publish/Subscribe paradigm in several cases, being the most prominent one any interaction that involves the Web service Eventing component. For instance, a subscriber can establish a link with an event source, and if an event is triggered, an event sink will be used to publish a message containing data about it.

Additionally, it is mentioned in the proposal that ACSI runs with a Report Control Block that require a Publish/Subscribe model for its correct performance.

Middleware distribution: while still making use of the hierarchy of the actors involved in the Smart Grid (the infrastructure that the proposal is most aimed at are power plants, either physical or virtual), the fact that it can be used for the integration of the distributed resources of a Virtual Power Plant provides an idea of how distributed the proposal has been conceived. Therefore, it can be argued that, as far as middleware distribution is concerned, it is a mostly distributed one.

This proposal for middleware in the Smart Grid can be characterized as:

$$SGM = SA\ (3) + CC\ (2\,||\,3) + MC\ (0) + MD\ (2) \qquad (7)$$

Advantages of the proposal: unlike many of the other proposals that have been studied, this one openly mentions how semantic resources are used for interoperability among devices, and provides information about the technologies used to offer the application layer an homogeneous set of services. What is more, wind power plants and DERs available in Virtual Power Plants are considered in a level that other proposals hardly ever do. Lastly, the proposal presents services genuinely useful for the whole deployment such as security.

Disadvantages of the proposal: the middleware architecture presented here lacks any relevant information about tests that may have been carried out, so it is hard to get an idea of the actual performance of the proposal. Furthermore, the proposal is lacking information about how to abstract hardware heterogeneity, as DPWS is more focused on higher levels an ACSI mechanisms are not detailed.

2.2.12. Intelligent Agents Platform

Álvaro Paricio et al. offer a middleware solution that tackles the heterogeneity existing among devices linked to both the Smart Grid (AMI) and other areas as Home Area Network devices [56]. The proposal refers to an Intelligent Agents Platform (IAP) used as the framework where message interchange is done. The devices present in a Smart Grid where this proposal is deployed (smart meters, sensors, etc.) are controlled by IAP Mediation Devices. The required management among the elements that are using the Smart grid framework is guaranteed by Integrated Network Management (INMS) functions, which rely on several functional blocks participating of actions like fault handling, integration capabilities or configuration management. The proposal makes use of Enterprise Service Buses for internal data transfer among its distributed components, so it will be regarded as an architecture.

Service availability: the proposal consists of three differentiated elements: two management layers that use internal buses for data transfer (the *Network Mediation Layer* and the *Management Application Layer*), and an intermediate one connecting the other two (*Middleware Communication Services*) that may or may not appear according to the specific needs of operational models. The Network Mediation Layer processes in a bi-directional fashion the usually large amount of information transferred through the deployment. There are pieces of equipment called IAP Mediation Device (MD) that use this layer for control purposes. Any management traffic type is controlled with these MDs, regardless of its nature (status monitoring, security, configuration, operation flows, etc.). The Management Applications Layer is responsible for using a collection of functional application blocks that are closely tied to an end user functionality as a reporting engine, alarm handling or a task scheduler. Lastly, the middleware communication services basically interconnect one layer with the immediately other one in case data have to be transported from the applications backend to the mediation system. All these features have been represented in Figure 19.

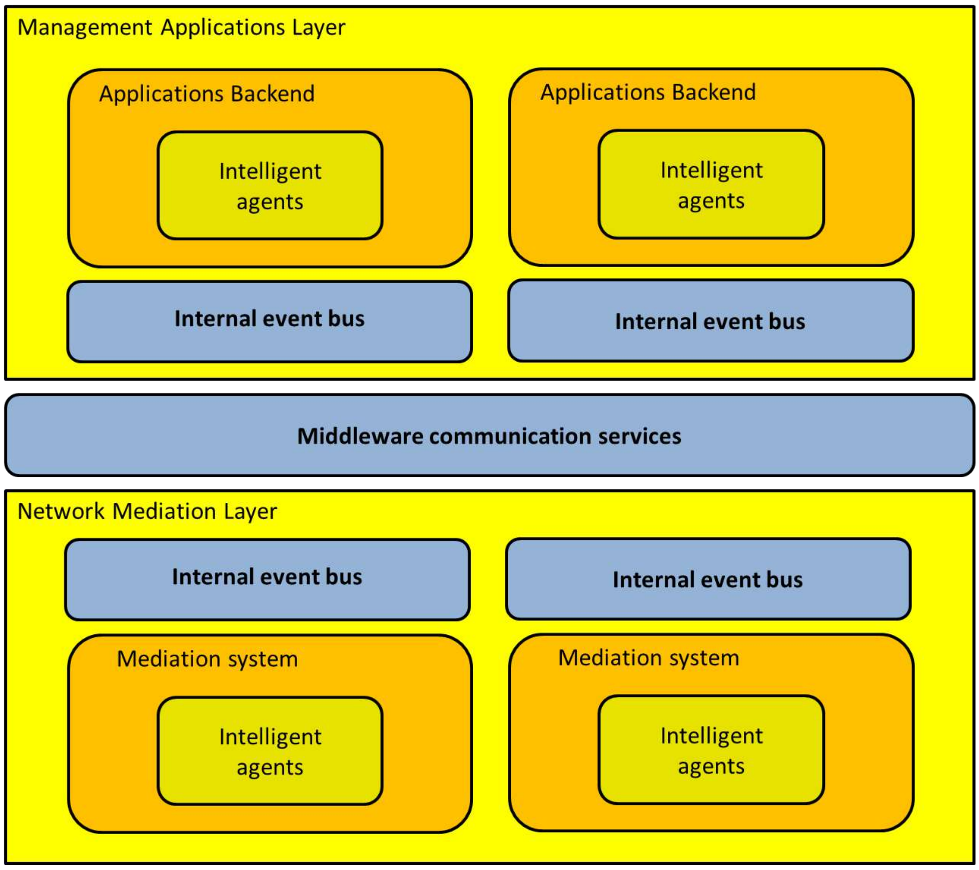

Figure 19. IAP-INMS appearance, as described in [56]

Computational capabilities: it is mentioned that the proposal has been tested repeatedly in different application domains. Furthermore, Customer Premises Equipment has been used as a way to employ a Smart Grid-like deployment where information could be transmitted via IP network. Unfortunately, there are scarce data of the equipment that was used for the text, or even if it was simulated or not (it is mentioned, for instance, that each Mediation Device will control one hundred concentrators, so a total of ten thousand AMIs will be managed).

Message coupling: there are two different kinds of paradigms that are used in this proposal. The first on is real-time for event collection, with no message decoupling whatsoever. On the other hand, a Publish/Subscribe mechanism is also utilized for the Intelligent Agents Platform used for the implementation and testing activities. There are even other kinds of paradigms, such as agents using peer-to-peer communications, or polling done on the concentrators used for testing activities.

Middleware distribution: as it happened before, not many data are available regarding what kind of distribution is expected to use the proposal. Since Mediation Devices, as well as the IAP platform are mentioned in the tests, it can be argued that the equipment will not be located as part of the power plant premises. Also, since the AMI seems to be controlled by the software components of the proposal, it can be said that this is a mostly decentralized proposal.

This proposal can be described as follows:

$$SGM = SA\ (3) + CC\ (0||1||2||3) + MC\ (0) + MD\ (2) \qquad (8)$$

Advantages of the proposal: it is implied by the information provided that this middleware architecture is capable of handling data of very different characteristics. The proposal also makes use of an Enterprise Service Bus, which is considered to be a very suitable option for distributed and Cyber Physical Systems with the features of the Smart Grid. Behavioral rules provide some degree of a semantics level. Some tests have been carried out in order to assess how the proposal would work in a distributed system.

Disadvantages of the proposal: the amount of services offered by the proposal is lower than in the other ones, despite using an ESB (which could be filled with software packages that will contain the services that can be provided). What is more, information about the performance results obtained in the tests, or what the devices used for them looked like, is missing. Finally, even though the middleware architecture can transfer many different kinds of data, it is not explained where services of major importance, like security or hardware abstraction, can be obtained.

2.2.13. Self-Organizing Smart Grid Services

Abdalkarim Awad and Reinhard German offer their own concepts for Smart Grid services capable of be aware of the context where they are deployed upon [57]. The authors introduce several metrics, referred to as *degrees*, to be used in order to define what features a development should have and to what extent. These degrees are: *degree of scalability* (among other facts, it assesses whether information can be created using just local messages), *robustness* (which evaluates the adaptability of the self-organizing services), *target orientation* (how any node creates its own data from an initial state), *emergence* (a phenomenon that can be witnessed at a macro level), *flexibility* (providing redundancy so that there will not be single points of failure in the deployment), *reliability* (ability of the self-organizing service to find alternative solutions in case any issue appears, like route unavailability) and *parallelism* (ability of the service to join or leave the system from different sides simultaneously).

Service availability: this proposal focuses on the services that can be provided in a Smart Grid and treats the middleware as a software component located in one of the two levels that are shown in Figure 20 (a). Here, it can be seen that the two levels of the proposal include a decision level (where the data is received and design, supervision or control are performed) and an infrastructure level that provides feedback to take decisions. Those two levels mimic the solution structure that is provided in Figure 20 (b), where decision support is done at the decision level and data communication combined with physical devices roughly matches the infrastructure level. It is expected from middleware that it will cope with several processes: data routing, aggregation, filtering and replication are mentioned by the authors. Middleware as a software participant in the system is less relevant here than in other proposals, as it is used in the infrastructure side of the communications that take place throughout the system. Therefore, it can be regarded as an abstraction middleware.

Computational capabilities: the proposal makes no explicit mention about the devices that are expected to be used to install the proposal. The physical level seems to correspond with Advanced Metering Infrastructure, whereas the decision support centers could be located in the aggregator side, as they are controlling the devices present on the very end of the deployment and they are able to send commands.

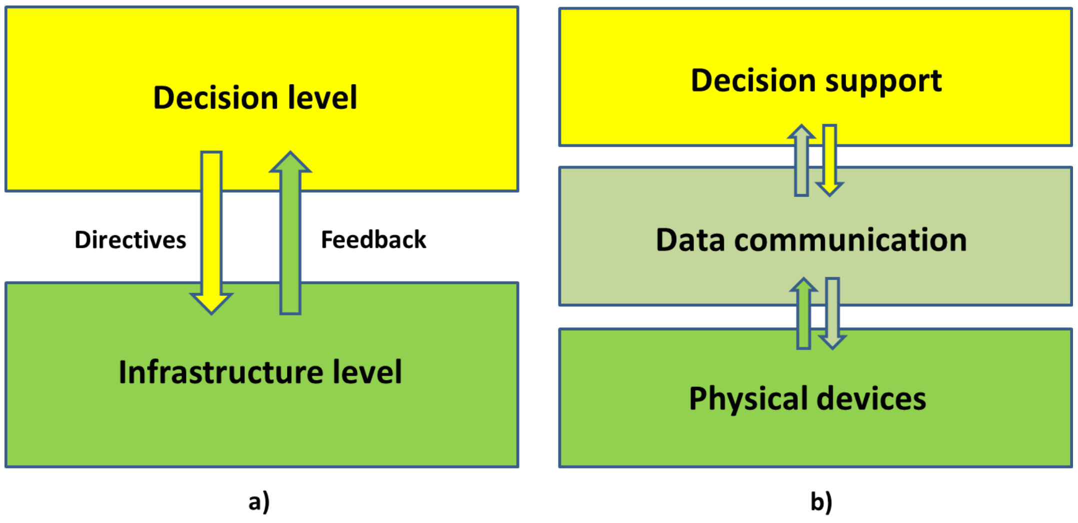

Figure 20. Solution structure (a) and main levels of the proposal (b), as described in [57]

Message coupling: real-time communications are mentioned the most in the proposal as one of the most critical ways to interchange data among the elements of the Smart Grid. Aside from that, not many details are providing any other form of message coupling (publish/subscribe, polling or Client/Server communications are not mentioned).

Middleware distribution: the proposal openly criticizes centralized middleware solutions, and it is also mentioned that all nodes have the same importance here. However, while it can be inferred from the proposal that there is some degree of distribution (which could mean that the proposal is a mostly decentralized one), it is not clear where the solution is expected to be installed. Peer-to-peer is mentioned in the proposal as the most preferred used for networks in the proposal.

Considering all these facts, the proposal can be described as:

$$SGM = SA\ (0) + CC\ (0||1) + MC\ (3) + MD\ (3) \qquad (9)$$

Advantages of the proposal: the middleware architecture put forward here offers a distributed way to interchange data among several software entities.

Disadvantages of the proposal: there is very few information on how the proposal can be used in an actual deployment. Besides, not much information is provided on what tests have been carried out to validate the middleware part. Additionally, not prominent services are offered as part of the middleware (semantic features, security). The overall impression of this work is that middleware is a secondary element that is required to have the self-organizing services working fine.

2.2.14. Secure Decentralized Data-Centric Information Infrastructure

The proposal presented here by Young-Jin Kim et al. puts forward how a framework for a decentralized and distributed system can be applied to the Smart Grid [58]. The authors claim that their proposal tackles challenges related to security, latency, real-time events and distributed data sources. The Information and Communication technologies infrastructure makes use of the Internet Protocol as the underlying way to transmit information at the network level. In addition to that, it offers securitization for the services that have been included in the proposal. It also makes use of the Common Information Model as a way to interchange data between Energy and Distributed Management Systems (EMSs, DMSs) in a seamless way. It is assumed that each of the devices present in the grid is able to handle symmetric-key operations to establish secure channels, as public-key operations are way more costly in terms of time and performance.

Service availability: taking into account the layered approach that has been taken in this solution, it can be said that this is an example of a middleware architecture. There are several layers in the whole proposal to take into account: the highest level one deals with what the authors refer to as *power applications*, which is above the proper middleware and has been conceived as the applications that can be accessed by the end users. The *Middleware Application Programming Interface* is offered just one level below; it offers a description of how middleware is accessed from the application layer and the services it can provide to it. More importantly, it provides a way to access to the services provided for event management services (*Non-time critical* and *time critical data/event*) and *control commands*. At the same time, these services rely on other software components based on networking and distributed information transfers, such as the *network cache* for information storage (used for pulled data) and a *publish/subscribe dissemination* (responsible for pushed data). A *Secured grid overlay network* is used for the data transmitted in unicast, multicast and broadcast communications. Lastly, *low-latency transport protocols* have also been included in the proposal. The proposal has been portrayed in Figure 21.

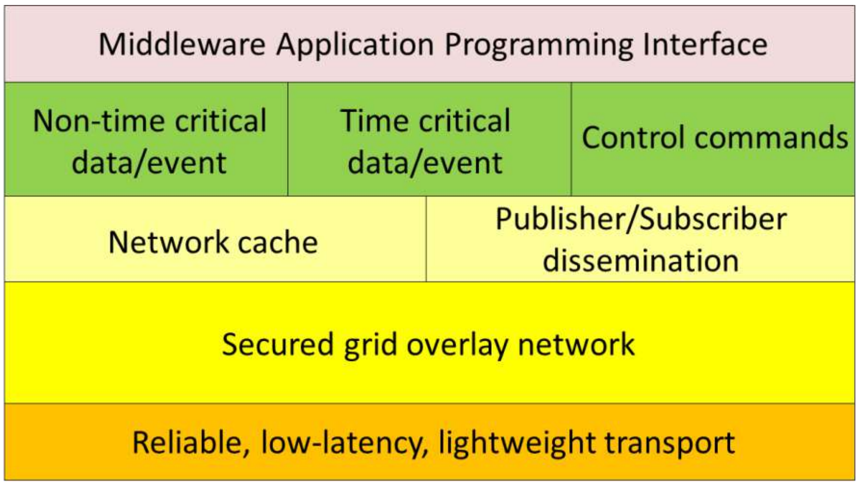
Middleware Application Programming Interface
Non-time critical data/event
Time critical data/event
Control commands
Network cache
Publisher/Subscriber dissemination
Secured grid overlay network
Reliable, low-latency, lightweight transport

Figure 21. Data-Centric Information Infrastructure, as described in [58]

Computational capabilities: there are few data on what kind of equipment has been conceived to have the middleware components installed. The authors stress the idea that their proposal is data-centric, as opposed to host-centric, so hardware is expected just to have the components installed without worrying too much about the nature of the hardware and its location.

Message coupling: there is a component in the proposal that is stated as responsible for publisher and subscriber dissemination, so the messages are expected to at least follow this pattern when they have to be transmitted. Real-time communications are also taken into account here: the software components that manage the events do so taking into account their real-time nature; it is also mentioned in the proposal as something that can be provided by means of a Real Time Protocol (RTP).

Middleware distribution: it can be inferred from the proposal that this is a mostly decentralized one, since it relies on technologies like IP or security for a whole network. Still, an application layer has been conceived to be accessed by just a group of machines, so there will be some centralization left, even if it is held by several pieces of equipment.

Therefore, the proposal can be described in the following manner:

$$SGM = SA\ (3) + CC\ (0||1||2||3) + MC\ (0||3) + MD\ (2) \qquad (10)$$

Advantages of the proposal: this middleware architecture is strongly based on the capabilities of any other distributed system, which makes the proposal easily portable to any other environment where CPSs are used. The proposal makes use of common technologies for networking and securitization, so its implementation and deployment should be easy enough. Alas, by having an API it is possible to access the middleware in a very accurate manner and have a good grasp on what it is capable of providing. Security is a concept of major importance in this proposal, as it is clearly stated by the presence of a layer used for securitization as its main role.

Disadvantages of the proposal: there are some major services as the ones used to provide semantic capabilities that are missing in this middleware architecture. The proposal has also included a level used for low-latency transport layer that, while a welcome addition, falls out of the traditional scope of middleware and may represent an extra work to have it working with the most usual transport layer protocols like UDP or TCP. Finally, the implementation works that have been carried out are conceived as something extra done for the benefit of the IP network, rather than to create a separate distributed software layer for hardware interoperability.

2.2.15. A cloud optimization perspective

Xi Fang et al. put forward their own proposal for a Smart Grid middleware-like architecture that uses cloud computing as a way to provide services in a distributed manner [59]. In this context, it can be said that cloud computing is a large-scale computational paradigm where the resources required to provide a service (storage, network utilities, etc.) are done in distributed infrastructures and/or platforms that make use of tools as the Internet or web services [60]. Cloud computing platforms are used in a widespread manner; companies such as Amazon or Microsoft offer their own solutions for commercial and research purposes, like Amazon Web Services (AWS, [61]) or Microsoft Azure [62]. According to the authors, cloud computing is a paradigm that should be used as a way to offer services in the application domain of the Smart Grid due to four reasons: a) it fits the high information processing requirements of the Smart grid, where changing prices during short periods of time demand high computation capabilities, b) cloud computing can improve information integration by avoiding having what they call “islands of information”, c) it can be used to outsource some of the tasks related to information management, thus having a less complex system and d) it eases the duties of Distributed Energy Generation parties willing to enter the electricity marketplace, who are usually of smaller scale that traditional participants.

Service availability: the proposal relies on several domains that encircle the core functionalities that have been conceived by the authors, namely, the *Smart Grid domain* (composed of seven subdomains that tackle different functionalities related to the application domain of this thesis: service providers, operations, markets, bulk generation, transmission, distribution and the customers), the *cloud domain* (used to offer storage or computing services), the *broker domain* (a mediation party between the cloud domain and the smart grid domain that collects the needs of the main actors in the Smart grid and finding the best cloud services available to satisfy them) and the *network domain* (which provides the network and the communications infrastructure). According to the subdomains that are offered as part of the Smart Grid domain, the proposal can be regarded as a middleware architecture, as it has a collection of functionalities that are matching a collection of services offered in a distributed manner. Figure 22 displays the location of the main components that have been included in the proposal. Interestingly enough, middleware as such is not mentioned in the proposal, even though it is clearly used to interconnect heterogeneous entities belonging to different domains by means of a broker.

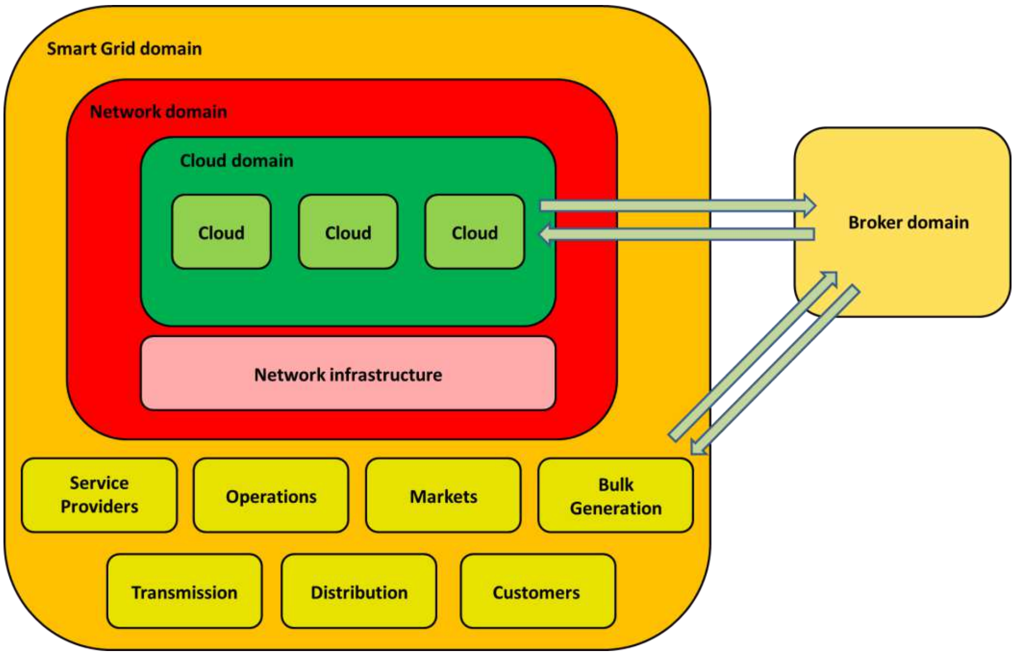

Figure 22. Main components of the cloud optimisation service, as shown in [59]

Computational capabilities: these are strongly linked to the paradigm of cloud computing. Since the underlying idea of the proposal is that the cloud will separate the ICT-related functionalities of the Smart Grid, any kind of device capable of running the algorithms and tools required for optimization (CPLEX Studio from IBM is mentioned as one of them) will be able to hold the middleware components. By “middleware components” it has to be assumed that they will be the main software subsystems associated to the subdomains present in the Smart Grid. The clouds that belong to the cloud domain might be regarded as part of the middleware as well, but due to the fact that they do not show what software components include, it can only be estimated what kind of facilities could be used. Overall, regular equipment (servers, Personal Computer-like appliances) falling within the facilities of the Aggregator or the TSO and DSO can be considered as the optimal choice to install the required software.

Message coupling: it is not possible to know what the main way to interchange messages in the proposal is. It can be assumed that a cloud computing-based infrastructure should be able to, at least, interchange data both in real-time (as mentioned in the related works that are described by the authors) or under a Publish/Subscribe paradigm (where data obtained from the Smart Grid domain can be stored until a subscriber interested in those data is registered).

Middleware distribution: since a cloud computing infrastructure is strongly implied in the proposal, it can be regarded as a mostly distributed one where even though most of the elements are distributed, there are also some more centralized ones (such as a the broker domain described by the authors) also included.

Therefore, the proposal can be described as:

$$SGM = SA\ (3) + CC\ (0||1||2||3) + MC\ (0||3) + MD\ (2) \qquad (11)$$

Advantages of the proposal: the proposal is crystal clear in describing how it is a distributed one (by providing cloud computing services, any proposal becomes distributed almost by default). In addition to that, the idea of distinguishing the Information and Communication infrastructure from the power grid one is appealing in the sense that it makes easier to work in parallel in both areas, provided that detailed interfaces are provided to cooperate among them. Services with a major importance as security have also been included in the proposal, as well as some tests to assess its likely performance.

Disadvantages of the proposal: it would have been welcomed if the authors had provided some kind of API to reflect how interactions are done between the cloud, the network and the Smart Grid domain. While it can be assumed that commercial solutions have been used (Amazon Simple Storage Service, known for its ability to work with other cloud platforms [63], is mentioned in the proposal, as well as its pricing options) there is no more specific information. Finally, next to no information is given on how messages are interchanged among the software elements present in the proposal.

2.2.16. KT Smart Grid Architecture and Open Platform

Jisun Lee et al. have also described the approach that has been taken by KT (former Korea Telecom) to offer a commercial solution based on an energy management platform [64]. It is claimed to be based on the functionalities offered by a SOA, along with intelligent agents and business process management. The new functionalities that have to be dealt with in order to integrate the new facilities that become part of the Smart Grid (Demand Response, Distributed Energy Resources, Electric Vehicles, grid performance optimization or energy management for end users) are taken into account. One remarkable fact of this proposal is that despite being offered by a private company, it is regarded as an open service platform, so the scalability that it can offer a far as service availability is concerned should be guaranteed.

Service availability: while middleware as such is mentioned in the proposal, there are three different levels , as it usually is the case among middleware architectures. The first, uppermost level is the *Customer Energy Management Systems* (CEMS), which integrates, among others, the management capabilities required for home dwellers (Home Energy Management System, HEMS). The second one consists of a database that contains information about customers, energy usage or even metadata obtained from the system. The third level, referred to as KOC Platform, is used for management of functionalities as demand response (Demand Response Management System, DRMS), renewable energies (Renewable Energy Management System, REMS), smart metering information (Metering Data Management System) and for business-related operations (Business Supporting System, BSS). Apart from these levels, there is a low level interface used to connect hardware devices typical of the Smart Grid (Supervisory Control And Data Acquisition systems or SCADAs, AMI, power panels, etc.). The service platform that has been depicted in Figure 23 also enables the participation of third party providers and customers that will use the platform by means of service interfaces.

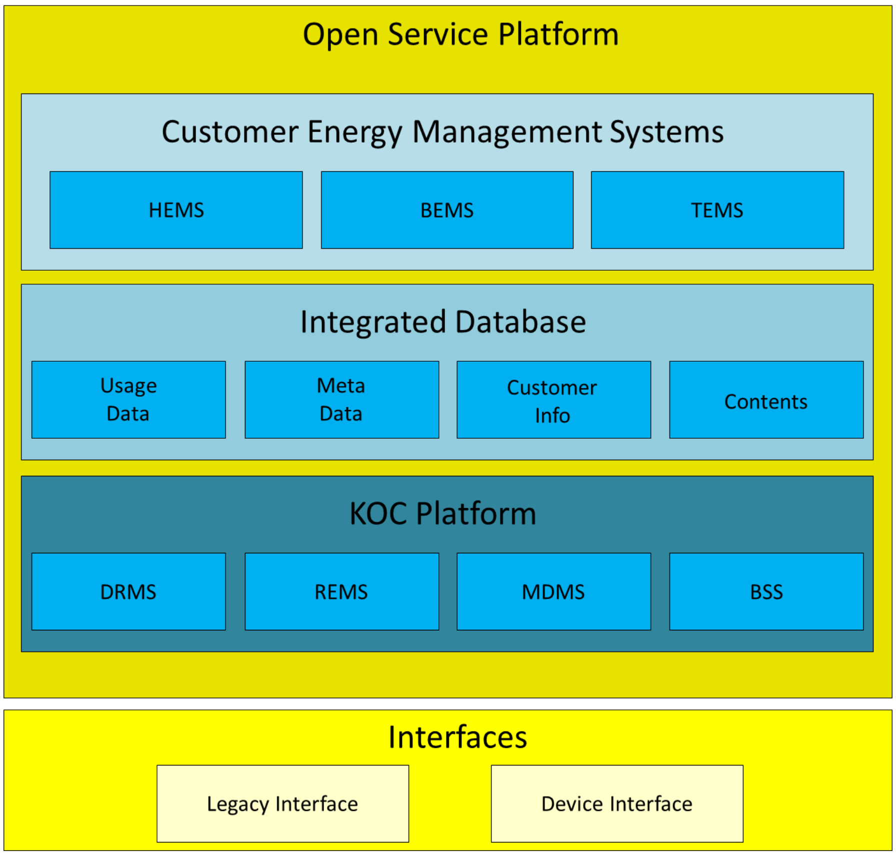

Figure 23. Main components of the Open Service Platform, as described in [64]

Computational capabilities: despite being clear that the proposal is accessing a plethora of devices, it is harder to tell where it is expected to be installed. Arguably, since information is going to be collected from AMIs or SCADAs, it could be located in a device that is outside those, so it would belong to either the aggregator or other participants like the TSO or the DSO. In that case, hardware as servers or mainframes should be powerful enough to have the components installed.

Message coupling: it is explicitly mentioned in the proposal that the data is analyzed and displayed in a real-time fashion. Apart from that, few data are provided on how the information is transferred throughout the architecture.

Middleware distribution: it is implied in the proposal that the platform will be installed throughout all the locations where its services are required. Yet there are several specific devices (specifically, the ones installed in the end users dwells) that are expected to feed data to the system, so it can be said that it will be a mostly decentralized proposal.

This middleware for the Smart Grid can be also described as:

$$SGM = SA\ (3) + CC\ (1||2||3) + MC\ (0) + MD\ (2) \qquad (12)$$

Advantages of the proposal: the proposal has also been tested in an actual scenario where information about energy consumption or energy flow is offered to end users. Perhaps as a consequence of having a realistic deployment done, concerns about electricity usage and information about the users of the platform is widely available. The idea of having the platform as an open development is positive in the sense that it allows the improvement of the architecture according to the needs of the end users that may become participants of it.

Disadvantages of the proposal: there are some concepts and terms of the proposal are not correctly explained, such as how data is sent from one side of the architecture to the opposite one. Likewise, end users are regarded just as consumers, instead of taking into account that they could also be in the position of injecting electricity into the power grid. Furthermore, there are several aspects of the deployment, like securitization of services, that are not mentioned and it is not known how they would be implemented, or even if they would be. Finally, information about how to access the application layer or what kind of Application Programming Interfaces are offered is not available.

2.2.17. Smart microgrid monitoring with DDS

The authors of this proposal [65] rely on an already developed middleware solution in the form of Data Distributed Service (DDS) to offer interoperability among the devices present in a microgrid. DDS is a standard defined by the Object Management Group (OMG) that defines a middleware protocol and an Application Programming Interface for interconnectivity in a data-centric fashion [66]. It describes how to interchange information at the data level (as opposed to bits, frames, packages or frames) by means of a Publish/Subscribe system, among other functionalities offered. The DDS specification has been conceived as a Data-Centric Publish/Subscribe (DCPS) model that describes all the features required to understand the procedures associated to DDS, to the point that a Platform Independent and a Platform Specific Model (PIM and PSM) are thoroughly depicted [67]. The DCPS model also makes use of a lower level, real-time oriented architecture referred to as Real Time Publish Subscriber (RTPS) [68]. This architecture is used to interconnect different DCPS-based, upper level developments. In order to understand how DDS works, there are several terms that have to be known. A *topic*, for example, is a definition for an association of participants in a communication specified and distinguished from others by several features (the topic name, the topic type and the topic domain ID). A *domain*, on the other hand, is a data space that comprehends a logical network of participations [69, 70] where several entities referred to as *domain participants* publish data and become subscribed to other pieces of information. Lastly, the Publish/Subscribe paradigm is pivotal in the development of a DDS solution, as it is the most prominent communication paradigm used whenever data are going to be transferred by means of this solution. In an environment like this, the publisher will implement a data writer, whereas the subscriber will have a data reader to collect the data that was published by the other entity of the communication. Figure 24 shows the main participants of a DDS communication where data is published and transferred to the subscribers.

Since the OMG only provides a description of the standard, it is left to companies, universities and research institutions the actual implementation of DDS solutions. There are solutions that have been developed under the idea of providing an open source solution to a greater or a lower extent (OpenDDS [71] or the OpenSpliceDDS developed by PrismTech [72]) whereas other solutions utilize a more commercial approach (eProsima [73] or the CoreDX solution developed by Twin Oaks [74]). Commonly, commercial solutions require licensing agreements for their usage in any kind of middleware development; open source-based solutions hardly ever require the same, even though the kind of license they are using must be taken into account so as it will not backfire and creates issues when exploiting the developed software.

As for the proposal itself, the authors take advantage of the Quality of Service features provided by DDS as a way to enhance the power grid. The RTI distribution of DDS is explicitly mentioned as being capable of providing the required efficiency and reliability for data communications. They stress the importance of having a data-centric model due to the fact that they are able to avoid having to worry about other lower level tasks (network locations, etc.). As far as the implementation of the proposal is concerned, it uses the Interface Description Language (IDL) as a way to define the data that are going to be transferred (that is to say), since IDL files are the ones used by DDS implementations to include the data that is relevant for a deployment. IDL is oblivious to the programming language that is used, so it enables the implementation of a DDS-based solution regardless of how it is programmed. A microgrid monitoring system is also offered in the proposal, where a DDS-enabled data acquisition system is used for data publishing.

Service availability: this proposal solely conceives middleware to transfer messages that will collect information from microgrid devices while offering this information via DCPS (as it offers an API that can be used by applications to access the data). Therefore, it can be regarded as a Message-Oriented Middleware with no further services (such as security or semantics) encased in it.

Computational capabilities: little is said about what kind of devices is expected to have the DDS components installed. Since the middleware is collecting data from devices as wind turbines or IP cameras, it can be inferred that it will be installed in other appliances that receive those data. Therefore, the middleware proposal will be installed in pieces of equipment that would correspond to the aggregator or the TDO/DSO infrastructure. They might even be installed in power plants, since it can be used to track real-time events that force the electricity production to change in one way or another.

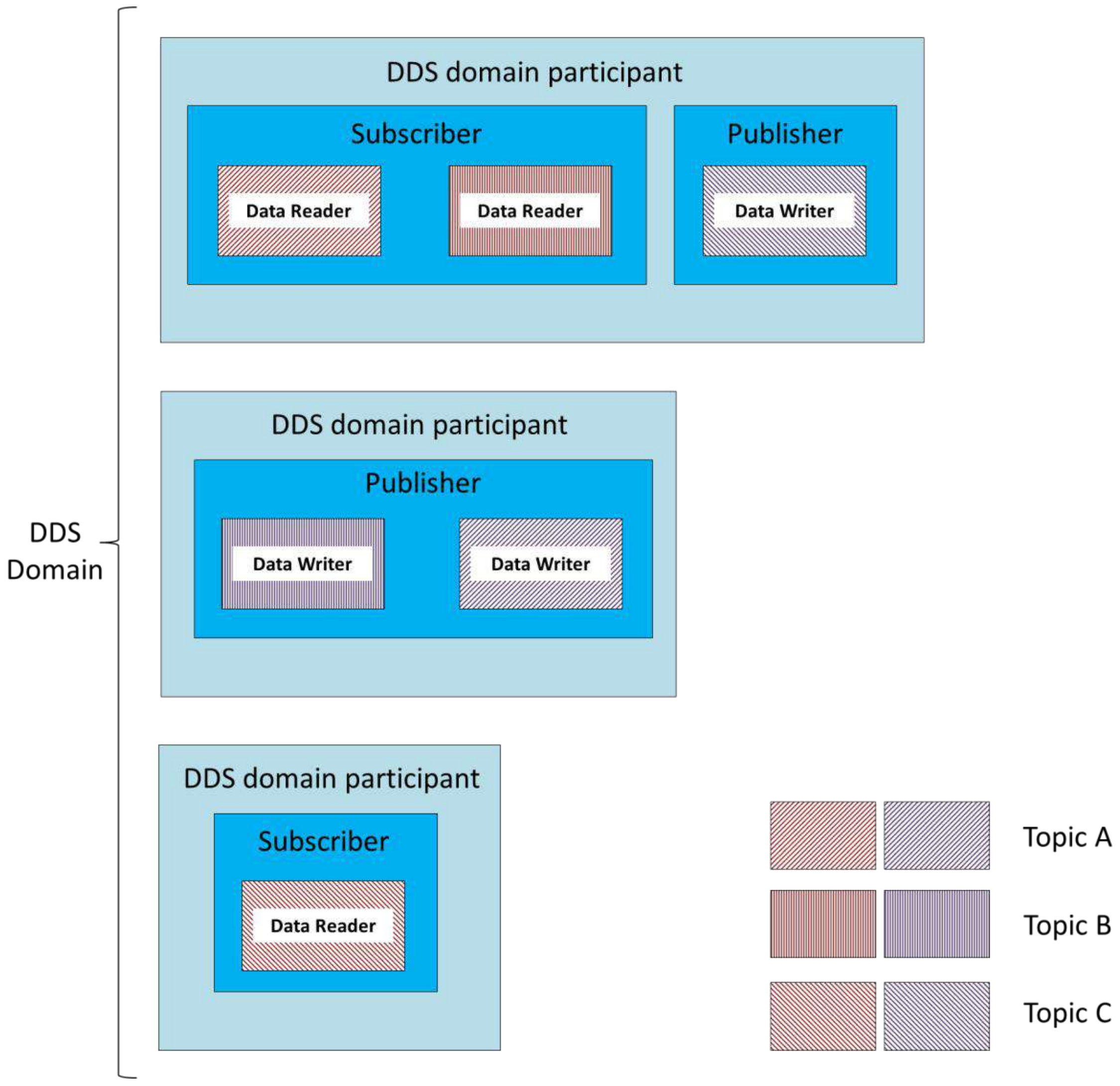

Figure 24. Main features of a DDS deployment, as described in [65]

Message coupling: the proposal is clearly oriented to two ways to transfer data, namely, Publish/Subscribe and real-time paradigms. On the one hand, DDS is heavily reliant on a Publish/Subscribe model for communications. On the other hand, a database has been conceived in the proposal where real-time data generated from the devices and the non-end user equipment. These data end up stored in the historical database for data playback, in case it is needed for information analysis or system performance.

Middleware distribution: DDS is consistent with the idea of having a distributed system sharing data among different elements. Consequently, it is a mostly distributed architecture that still has some elements more likely to have a higher level of computational capabilities (for example, a DB server will usually be more powerful than a mobile device) and hence have more functionalities installed.

The middleware solution that has been presented here should be regarded as:

$$SGM = SA\ (2) + CC\ (1 || 2 || 3) + MC\ (0 || 3) + MD\ (2) \qquad (13)$$

Advantages of the proposal: DDS is a portable middleware solution that has already been proven as efficient in several application domains. It is flexible enough to support different kinds of communications (Publish/Subscribe, real-time) and is able to provide an API that can be used to the advantage of the application developers. In addition to that, its distributed nature makes possible the addition of an ever-increasing number of infrastructures used in Distributed Energy Resources.

Disadvantages of the proposal: While DDS results convincing in many ways as a middleware solution for the Smart Grid, it presents trickier challenges related to Intellectual Property Rights and licensing: OpenSpliceDDS is licensed under the third version of the GNU Lesser General Public License (LGPLv3) for its usage that, unlike the ordinary GPL, does not automatically imply that the developed software will also be of free nature [75]. Depending on the kind of development that is going to be carried out, this might be desirable or counterproductive. On the other hand and in case of CoreDX, a license might be provided for research and runtime or for commercial exploitation, which could imply a payment to Twin Oaks. In the end, many of the advantages and disadvantages are associated to the fact that a solution already developed by a third party is going to be ported to a microgrid, which may result in the need of getting commercial agreements on the libraries and the content that can be used. Regarding the aspects of the proposal that are not related to DDS, it can be said that, for the good and the bad, it merely ports the middleware standard to a microgrid, offering a reliable solution easy to port to other environments but no additional services that would enhance it are included. Besides, there is little information on what kind of devices is expected to have the middleware.

2.2.18. ETSI M2M

Guang Lu et al. put forward a proposal [76] where it is proven how a set of standards for Machine-to-Machine communications designed by European Telecommunications Standards Institute (ETSI) [77] can be ported to the Smart Grid. ETSI is an institution known for its efforts in Information and Communication Technologies in general, with significant contributions in the Internet of Things. It is stated by the authors of the proposal that the application domain involving the Smart Grid presents a series of challenges that are common to the ones that have to be tackled by middleware solutions, like interoperability (conceived in the proposal as the capacity of two or more systems to exchange information based on a common set of communication specifications) and scalability. Some other features such as security (key management, authentication, cryptography) and device management (configuration, performance, firmware/software, etc.) are considered in the proposal, at the same time as service specific of the Smart Grid (demand response). The architecture design that, according to the authors, is offered by ETSI M2M, offers the idea of Service Capabilities (SCs) as a way to provide the functionalities to be shared by the applications located in the upper layer. Some of the most prominent SCs are: Application Enablement, Remote Entity Management, Telco Operator Exposure or Interworking Proxy.

Service availability: the information provided by the authors of the proposal mentions how the Service Capabilities in the ETSI M2M architecture can be installed in a quite distributed manner. The proposal is also claimed not to require a specific underlying access technology. Furthermore, an open API is mentioned as a way for the applications to obtain access to the middleware. Taking into account that security and device management are also implemented in the proposal, it can be mentioned that this is a middleware architecture. When applied to the Smart Grid, there will be a collection of control centres that will gather as particular domains of a M2M Core that communicates via network with a group of Smart Grid Systems that include the components that deal with the main facilities of a deployment (generation, transmission, distribution). Its overall appearance has been displayed in Figure 25.

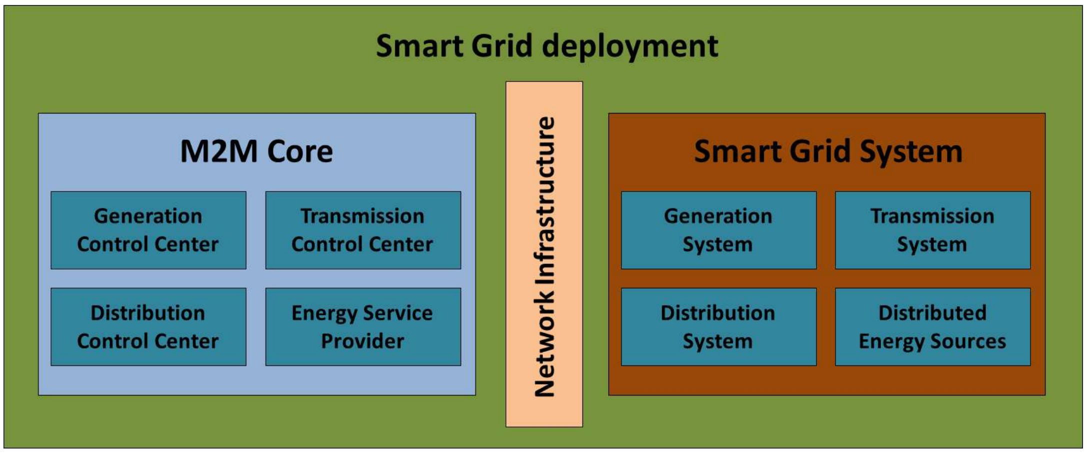

Figure 25. Structure of the ETSI M2M proposal, as described in [76]

Computational capabilities: it is clearly stated in the proposal that Service Capabilities are installed in devices as gateways and Machine-to-Machine communication cores, which can be estimated as regular PC-like or server-like hardware components in a deployment. In addition to that, since ETSI M2M is based on the Internet of Things, it has been conceived to be installed in almost any location. Therefore, the proposal could be installed almost anywhere in a deployment, such as the Advanced Metering Infrastructure of a home dweller, Aggregator, TSO or DSO equipment or even monitoring the facilities of a power plant.

Message coupling: there is little data about how messages are expected to be transferred. While it is mentioned once that a real-time automated response is to be expected, the most predominant idea that has been considered is that there are appliances like servers for information on power resources or how M2M servers have been included in the proposal, so it can be said that a Client/Server architecture is the one that is coming closer to how messages are interchanged. No information available about the existence of a broker is present either, so it can be assumed that a publish/subscribe paradigm is unlikely to be used in this case.

Middleware distribution: as it usually is the case with IoT-like proposals, this is a mostly decentralized architecture that is able to run in devices of very different nature, even those that have limited computational resources. In addition to that, a peer-to-peer paradigm could be implemented as well, as three are several machines that will communicate with each other (which can be deduced from the preeminence of the concept of M2M in the proposal) without requiring the intervention of any end user or application with a Graphical User Interface (GUI).

This proposal can be represented by:

$$SGM = SA\ (3) + CC\ (0||1||2||3) + MC\ (2) + MD\ (2) \qquad (14)$$

Advantages of the proposal: this middleware architecture covers most of the prominent issues that have to be tackled in a middleware for the Smart Grid, such as service and API availability, security or interoperability. Details about how prototyping is performed and the kind of tools that have been used are profuse and focused on each of the characteristics that have been deemed as important by the authors (interoperability and scalability, security, device management, demand response). The fact that this proposal is ported from an IoT application domain also points out the fact that CPSs share multiple challenges that can be addressed with a similar kind of solution.

Disadvantages of the proposal: information offered about the challenges and the activities that have to be carried out to port the set of ETSI M2M standards from the IoT to the Smart Grid is somehow confusing. While it is mentioned a wide plethora or services that can be added, sometimes it is not clear how they are implemented, if either a set of protocols cooperating with each other or as software components that are executed in different pieces of hardware. Finally, how data messages are transferred from one part of the deployment to the other is described with too scarce information.

2.2.19. Smart Middleware Device for Smart Grid Integration

Oliveira et al. [78] show in their work regarding middleware how it can be included as another software component in a single hardware device tailored for Smart Grid integration. One of the main issues that according to them is yet to be fully dealt with is the integration between devices and protocols such as Modbus (considered as an standard for the industry application domain), which would require special gateways in a deployment. This proposal puts forward a design and implementation for a gateway of this sort, which has been named Smart Middleware Device (DMI in the Portuguese acronym used by the authors). The appliance is capable of managing the software components to be used for protocol translations, along with data flow characteristics. This device is expected to be used as a connector between the power grid and the ICT domains, as it will be linked to other power stations present in the grid, as well as to routers included in the network infrastructure.

Service availability: DMI has been conceived as a middleware architecture because it explicitly mentions a plethora of services that have been developed. According to the authors, though, the middleware solution will be included just as a part of the device that has been implemented for that purpose, so it will be present only in one device. As for the services, they are part of two different groups: the translators and the core components. Translators will be used as a way to transfer information from the protocols deemed as standards to be considered by the proposal (Modbus, IEC 61850 and Distributed Network Protocol, version 3; it is hinted that other protocols might be integrated too). At the same time, the core components are used for the usual requirements faced by information transferring in the Smart Grid: queries done between its different elements and units, a repository to store those answers and a management component to control operations as queries (or queries pushed to a SCADA) or answers that need translations. The appearance of the internal components of the proposal and how they interact with external elements has been summarized in Figure 26.

Computational capabilities: rather than considering what kind of device could have the proposal installed, the middleware is expected to run in a specific one. According to the specifications offered by the authors of the proposal, a DMI has been assembled and runs as a service under a Linux, Berkeley Software Distribution (BSD)-like operating system. Considering those features, it could be included as part of the pieces of equipment installed in almost any location, even though the fact that it is used for translating protocols used in power substations points out that it could be located as part of the TSO or the DSO infrastructure.

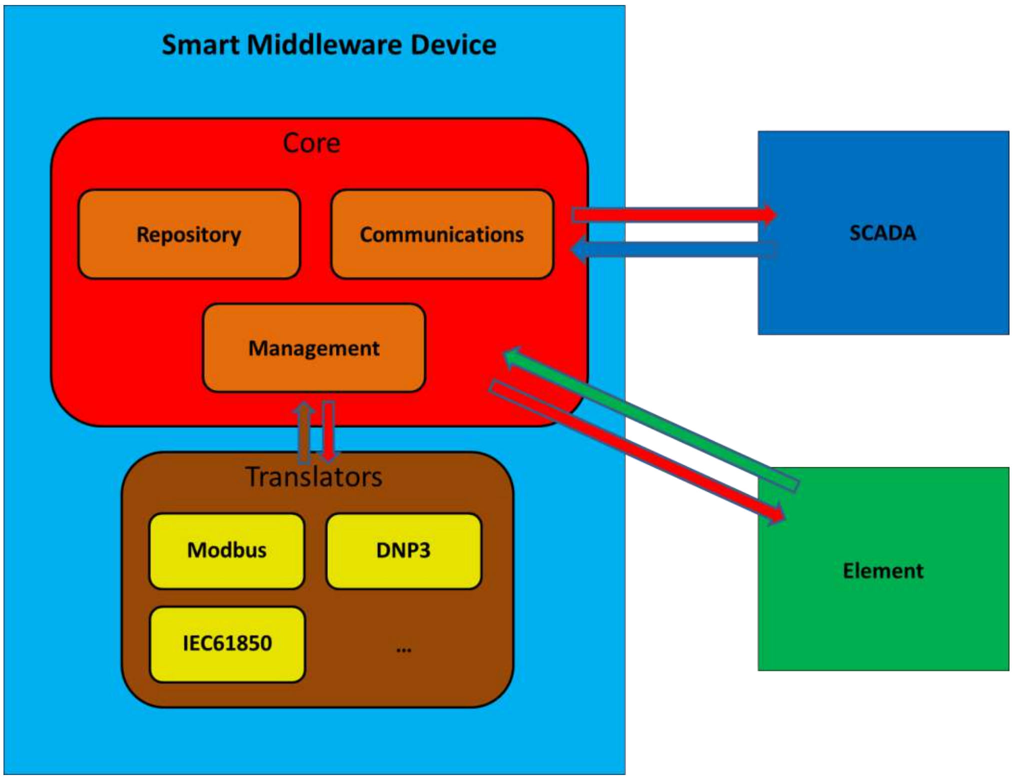

Figure 26. Smart Middleware Device, as described in [78]

Message coupling: it is mentioned by the authors that when tests were carried out with the device, it was able to receive real-time data from a power grid, which was used to configure Quality of Service indicators that would trigger alarms or actions to be taken. In addition to that, it is also explicitly stated that it has been defined within a client / server architecture.

Middleware distribution: this proposal is fully centralized in one single device (the Smart Middleware Device), which is the one used to include all the software components required for data transmission and protocol translation.

According to the matrix that was defined before, this proposal can be defined as follows:

$$SGM = SA\ (3) + CC\ (2||3) + MC\ (3) + MD\ (0) \qquad (15)$$

Advantages of the proposal: the authors have included information about actual tests that have been carried out by integrating the DMI with a Smart Meter, and a General Packet Radio Service (GPRS)-enabled device used by the device of the proposal to send commands and receive readings. Integration of different kinds of protocols by the same kind of entity is also addressed by the proposal; translating data representations from one format to the other is also to be expected considering how middleware is used.

Disadvantages of the proposal: one of the most important decisions taken by the authors of the proposal is confining the middleware in a single device, which will require having that device whenever middleware is required, so middleware is not actually providing hardware abstraction in this case. Overall, there are not many data on how the put forward solutions are going to be implemented for a distributed environment, and the fact that the middleware will be located in one single device makes it more vulnerable for the deployment, as middleware may completely fail in case there is a malfunction on the hardware that has it installed. What is more, the idea of having a device embodying the concept of middleware can be disorienting, as it looks as inconsistent with the intention of having middleware as a software layer installed in several devices and appliances that will negate the differences between them to offer a homogeneous set of facilities to an upper layer. Lastly, there are some major features like security, semantics or an API for the application layer, which are not mentioned in the proposal.

2.2.20. WAMPAC-based Smart Grid communications

Aditya Ashok et al. stress the importance of security in CPSs in their work [79]. Specifically, they address a Wide-Area Monitoring, Protection and Control subsystem that is included in a Smart Grid. It is claimed in the proposal that PMUs are the main components of the power grid that take advantage of the existence of a WAMPAC subsystem, as it leverages the PMUs to obtain real-time knowledge from the operations that are being carried out in the power grid where it is installed. According to the authors, WAMPAC can be divided into several different subdomains: Wide-Area Monitoring Systems (WAMS), Wide-Area Protection Systems (WAP) and Wide-Area Control (WAC). As it happened with former proposals, the authors rely on a SCADA system to collect information from outside the middleware. They also provide a classification on cyber security attacks (timing attacks, data integrity attacks, replay attacks) that can be performed onto a system. According to the proposal, deploying the different WAMPAC subsystems can have several challenges: for example, WAMS must be able to provide high availability, integrity and some degree of confidentiality in utility data. Furthermore, WAPAC schemes, based to an extent on power-related protocols such as IEC 61850, must guarantee that messages that are transferred ensuring that they have been authenticated to separate malicious information or commands. Finally, the idea of a real-time WAC using information from a PMU is planned to be further researched. The proposal makes use of a game theoretic framework in order to model features from both cyber and physical points of view.

Service availability: the authors´ proposal is an architecture built on the three subdomains that WAMPAC is made of. WAP, for example, relies on large amounts of information collected from the whole deployed system, which will be used to enable decisions that will counter disturbances across the system. Furthermore, WAMS is claimed to be responsible for distributing information in a reliable way (which involves an infrastructure that must offer integrity and high availability regarding the PMU data) and needing a high-speed networking infrastructure. Lastly, WAC is mentioned as a potential way to offer applications as secondary voltage control, static control or inter-area oscillation damping. The location of the WAMPAC controller in a deployment is shown in Figure 27. Note that this WAMPAC component that has been included is regarded as part of a wider Smart Grid deployment bent on solving security issues, so it is not conceived to be a whole middleware solution.

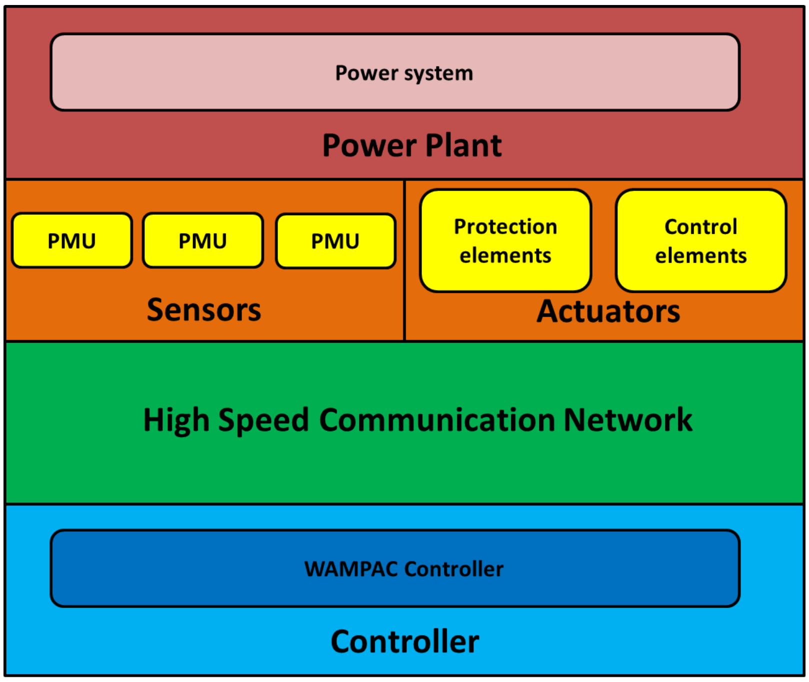

Figure 27. WAMPAC controller location, as described in [79]

Computational capabilities: even though not very much information is provided about this feature, it can be said that the WAMPC controller involves networking, security and data management solutions, so it is supposed to be part of the facilities used for communications and data interchange between the end users´ equipment and the infrastructure used to generate electricity, that is to say, the TSO and DSO pieces of equipment.

Message coupling: as it happened with the former case, there are few data to get a grasp about how messages are transferred, except for the idea that securitization must be provided. Nevertheless, real-time communications are mentioned as the most usual ones that take place in the proposal scenario. Furthermore, the approach taken by the North American Synchro-Phasor Initiative (NASPI) regarding secure synchronized data measurement infrastructure

(NASPInet, [80]), where a Publisher/Subscribe component has been implemented, is mentioned in the proposal as a known solution.

Middleware distribution: while the proposal makes use of a distributed domain for information interchange, and it is included in a distributed system as the Smart Grid is, it is not made clear how devices are expected to be deployed where the proposal can be installed. Considering the kind of infrastructure mentioned in the proposal, it can be claimed that it will be a mostly decentralized solution, as the WAMS component requires a network for data delivery.

This proposal can be further described as:

$$SGM = SA\ (3) + CC\ (2\,||\,3) + MC\ (0\,||\,3) + MD\ (2) \qquad (16)$$

Advantages of the proposal: security is strongly tackled in the proposal and it is made clear how the Smart Grid can be securitized against the most damaging attacks. Also, the concept of using a game theory framework as a tool for security in Cyber-Physical Systems is interesting and offers a different point of view on how to address it. The authors of the proposal have also provided a justification of how a testbed can be implemented to test security in a Smart Grid-like deployment when game formulations are involved.

Disadvantages of the proposal: while the focus of the proposal on security is welcomed, as it is a functionality of major importance in a distributed system, there is less information about all the other services expected to be part of a middleware architecture (semantics, API, etc.). Clearly, the main idea that the authors want to transmit is that they have created a system capable to provide secure communications, rather than a middleware solution with several services encased.

2.2.21. C-DAX

Michael Hoefling et al. [81] provide their own vision of what a secure middleware should be according to what has been designed in the research project called *Cyber-secure Data and Control Cloud for power grids* (C-DAX, [82]). As it was presented in other proposals, their solution is focused on solving security issues found in data exchanges for the Smart Grid (the authors refer to Active Distribution Networks or ADNs). Facilities that are likely to make use of this proposal according to the criteria of the authors are: Phasor Measurement Units (PMUs), the ones related to Real-Time State Estimation (RTSE) and Phasor Data Concentrators (PDCs). RTSE applications are described as receiving aggregated information from the PDCs and feeding it in a mathematical model used to estimate the current conditions of the grid. PDCs are used for the reception of data that have been timestamped (refreshed at a frequency of 50 Hz), time-aligned and aggregated from different PMUs. Information is sent throughout a deployment with this solution according to topics established to separate one kind of content from the others. These topics are defined as representations of abstract nature used for unidirectional information channels with a certain storage capacity. PMU measurement is made by using NASPInet (North American Synchro-Phasor Initiative Network) as the protocol.

Service availability: the middleware that has been prepared for this project can be regarded as a Message-Oriented Middleware, considering that the main efforts of the proposal are centred in creating a secure procedure for the interchange of messages in the deployment, instead of offering a collection of services that will be contained in the software infrastructure chosen for the middleware. Two planes of information have been defined as a way to organize the data: control plane and data plane; the latter is utilized in combination with Designated Nodes (DNs) that grant access to the cloud used to implement the middleware proposal for the C-DAX project. Data Brokers (DBs) are used as a way to either store or send forward information. Finally, there is a Monitoring and Management System devoted to monitor and control the deployed devices. The overall appearance of the proposal has been displayed in Figure 28.

Computational capabilities: the proposal is expected to use pieces of equipment that are to be expected according to a regular Information and Communication Technologies deployment. It is also mentioned in the proposal that in order to perform the tests, a 100 Mbit/s link has been used. Therefore, and taking into account that the proposal deals mostly with data transmission and securitization, it can be inferred that it is expected that TSO or DSO infrastructure will be the main beneficiaries of the security measures that are described by the authors. In addition to that, pieces of equipment present in power stations may also welcome having the proposal installed, as it will add security to information of critical importance for their performance, as it will depend on the feedback obtained from a Smart Grid.

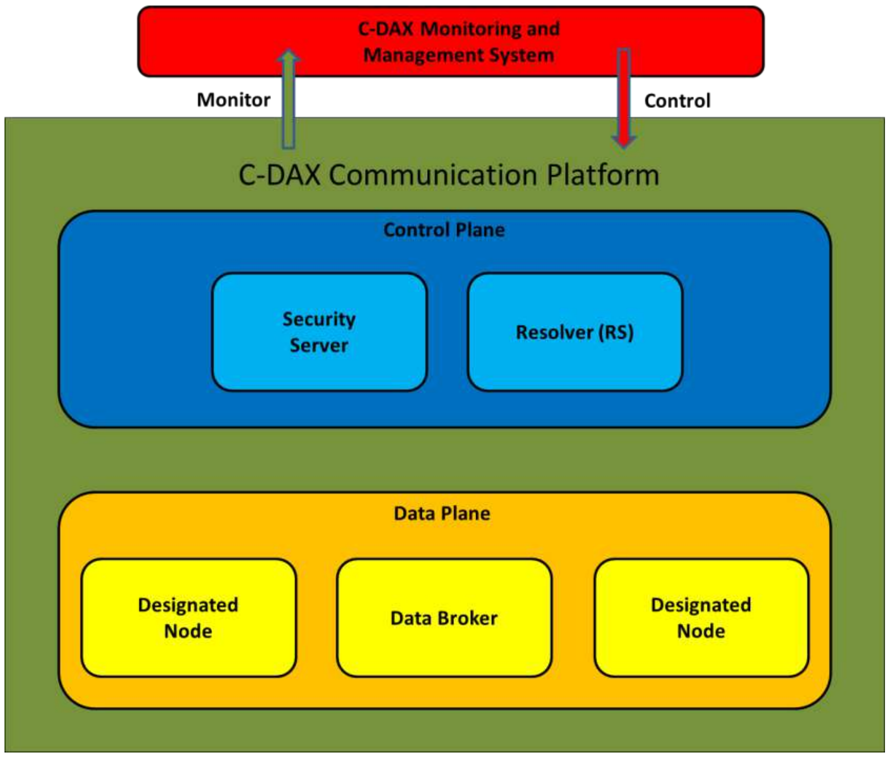

Figure 28. C-DAX middleware appearance, as described in [81]

Message coupling: the proposal makes possible several different ways to establish communications. The authors distinguish three of them: *streaming communication* mode (where subscribers receive topic data after becoming part of one of them in a continuous, non-interrupted way), *query communication* mode (subscribers send explicit data queries) and *point-to-point* communications (without using Designated Nodes or Data Brokers). However, it can be said that there are two different paradigms in the proposal that are heavily mentioned and used. On the one hand, real-time data are required to be taken into account, as they usually imply information obtained from the smart meters deployed in the Smart Grid deployment in an almost instantaneous fashion. On the other hand, information is published throughout the data plane (and by means of the designated nodes that are at both ends) using a Publish/Subscribe paradigm, where application data is published in one end of the communication and is consumed by the subscribed located at the other end. It is explicitly stated by the authors that one of the objectives of the project is to adapt the Publish/Subscribe paradigm to the needs of power grids. Finally, Client/Server communications are discouraged by the authors of the proposal, claiming that they demand servers to be configured properly and clients have to be re-configured when they have to establish connections with backup servers.

Middleware distribution: the proposal is a mostly decentralized middleware solution due to the fact that it is distributed in several different pieces of equipment that are part of the ICT infrastructure of the Smart Grid (subscribers, publishers, databases) and a communication infrastructure has been used in order to perform tests on the proposal. Since there are elements that manage data reception and delivery (such as the Data Brokers) there will be still some more prominence of several software elements over others, thus proving that the solution is not entirely peer-to-peer, with similar features in each of the nodes where the proposal has been installed.

This proposal can be described as:

$$SGM = SA\ (2) + CC\ (2||3) + MC\ (0) + MD\ (2) \qquad (17)$$

Advantages of the proposal: the authors offer a compelling proposal that has been tested in a realistic scenario (a laboratory belonging to one of the project partners). There are several scenarios that have been consider for testing activities, being failing DBs one of them.

Disadvantages of the proposal: The number of services included in the proposal is more modest than in other ones, due to its condition of being a Message-Oriented Middleware. No information is provided regarding an API to be offered to the higher levels as a specific way to interact with the middleware shown here.

2.2.22. Building as a Service (BaaS)

The main concept that has been implemented in the proposal led by Susana Martin et al. is the usefulness of the Smart Grid for energy efficiency in buildings [83]. According to the authors, buildings can be used to obtain services from them (hence the term Building as a Service) by developing a middleware solution used to link a collection of entities of a CPS that stresses the importance of energy consumption. The Open Services Gateway initiative (OSGi, [84]), which is claimed to offer openness, interoperability and transparency, offers interfaces in the implementation works required for the solution. Several other ideas common to this kind of application domain (Event-Driven Architecture, Service-Oriented Architecture) have also been included in the proposal. Interoperability among several interlinked entities is provided by integrating Building Information Models (BIMs), Data Warehouses (so as to store information as a whole) and Building Management Systems (BMSs) and the legacy ICT facilities as a single system. Furthermore, services based on optimal control, prediction and assessment have been included too.

Service availability: The proposal is considered to be a middleware architecture due to the fact that it has included several devices that are distributed in different layers. However, the only software component regarded as middleware by the authors of the proposal is the Communication Logic Layer or CLL. It is encased with two more levels as part of the intermediation architecture defined for this proposal, that is to say, the Application Layer and the Data Layer. The Data Layer has included information dealing with the services related to the building used for the BaaS concept (for example, the ICT infrastructure for weather and access control), the Data Warehouse infrastructure, the Building Information Modelling Server and the external services responsible for collection information. The Communication Logic Layer is composed by two sub levels: the Core Communication and the Data Access Object sublayer. The first one consists of the Domain Controllers or DCs and the Data Acquisition Control Manager or DACM. The second one adds the Data Access Objects for the DC and the DACM. Figure 29 shows the overall appearance of the proposal.

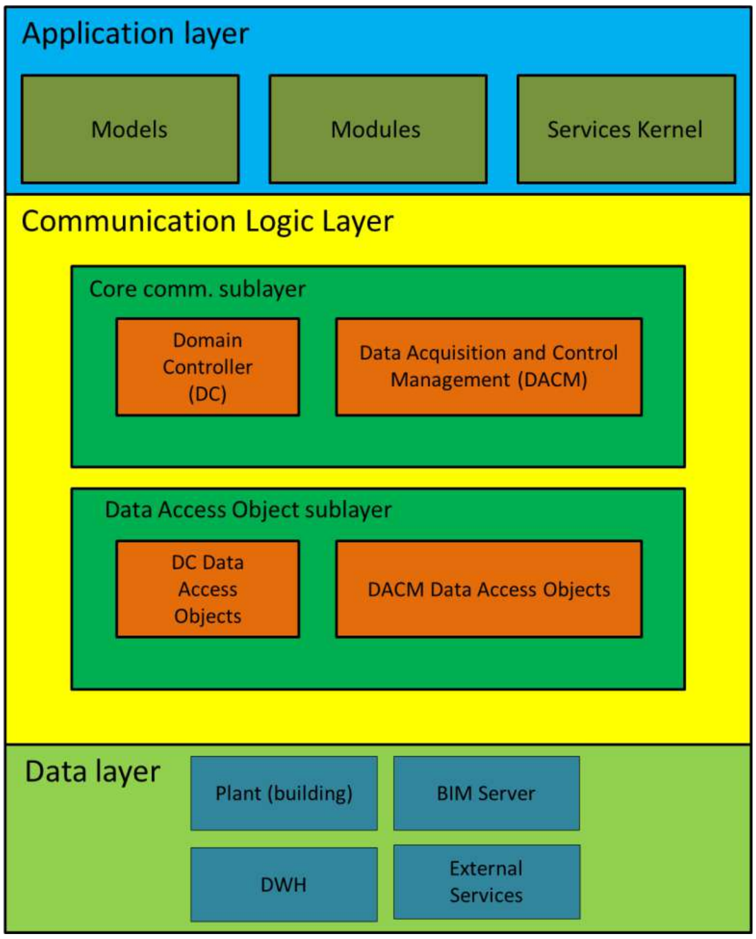

Figure 29. High level architecture of the BaaS system part of the proposal, as described in [83]

Computational capabilities: the software components that the middleware architecture is made of are implemented as OSGi-bundles that are usually on the kilobyte or the megabyte scale, so it is easy to install the software in any kind of piece of hardware with normal and even low capabilities. Considering that data is being obtained from sensors it can be argued that the middleware solution can be installed in any intermediate piece of equipment belonging to the Aggregator, the TSO or the DSO.

Message coupling: it is explicitly said in the proposal that this middleware has been conceived for applications that follow a Client/Server architecture. Additionally, it is mentioned that the proposal has been designed taking SOA and EDA concepts into account and a modular event-driven design has been created. On the other hand, it is added that components can be subscribed to events in an EDA architecture, so it is subtly implied that there could some kind of Publish/Subscribe system. However, this point is never mentioned explicitly, so a Client/Server approach should still be considered as the one that is used here.

Middleware distribution: the proposal is a mostly decentralized one due to the fact that while it is distributed in different elements that are independent from each other (the buildings) there are still some elements that have a higher degree of responsibility than others, such as the Data Acquisition and Control Manager.

Considering all these features, this middleware proposal can be described like this:

$$SGM = SA\ (3) + CC\ (1||2) + MC\ (2) + MD\ (2) \qquad (18)$$

Advantages of the proposal: the proposal makes use of an API for data interchange that makes it easier for application developers to access the middleware solution, and how levels are interfaced with each other is made clear. Moreover, the usage of OSGi technologies is consistent with the idea of having open source software technologies for the middleware, which is a desirable feature for further scalability and improvement of the implementation works done. Finally, the proposal makes use of a collection of standardized technologies that ease the replication and implementation of software solutions. The proposal makes use of software technologies of widespread usage that have proven to be useful, such as Java Database Connectivity (JDBC) in the Data Warehouse component, JavaScript Object Notation (JSON) for the Building Information Model or Simple Object Access Protocol (SOAP) in the Building Management System.

Disadvantages of the proposal: Services of major importance to be encased by a middleware solution that are related to security or the semantic enhanced of information are not present in this related work. Furthermore, it is very centered on the idea of using it only for energy consumption in buildings, which may make challenging to port the middleware architecture to other applications related to the Smart Grid, or even other Cyber-Physical System.

2.2.23. Middleware-based management for the Smart Grid

Villa et al. put forward in their proposal [85] how a hardware platform can be used to manage, control and integrate electrical installations that are large scale and heterogeneous.

Specifically, they refer to it as Embedded Metering Devices (EMDs), capable of controlling pieces of equipment that require no parameters to be changed. Among the features that are claimed to be enabled by the authors, adaptability (EMDs for devices that require no changes), scalability, availability (segments of the network can continue working in an autonomous way in case of failure) and hierarchical design are the most prominent ones. Other advantages related to the hardware and its components are: its small size and cost, low energy consumption, flexible access, access transparency and conventional tools. The proposal aims to have a generic platform capable of develop power management services for any kind of environment. It should be noted that the main interest of the authors is regarding the device as Advanced Metering Infrastructure, rather than software components to be installed in one piece of hardware in a microgrid.

Service availability: while there is very scarce information about the services that can be provided, it can be said that the software used for device interconnectivity is solely employed for that purpose. Consequently, it is a hardware abstraction kind of middleware where no other services are offered as part of the deployed hardware. What is more, there are no mentions on how the middleware is accessed by any other higher layer (in case it is a possibility), so the existence of an API is unlikely. The main hardware elements that are to be mounted on a node that is using the device (Bluetooth interfaces, the microcontroller or the very Embedded Metering Device) are present in Figure 30.

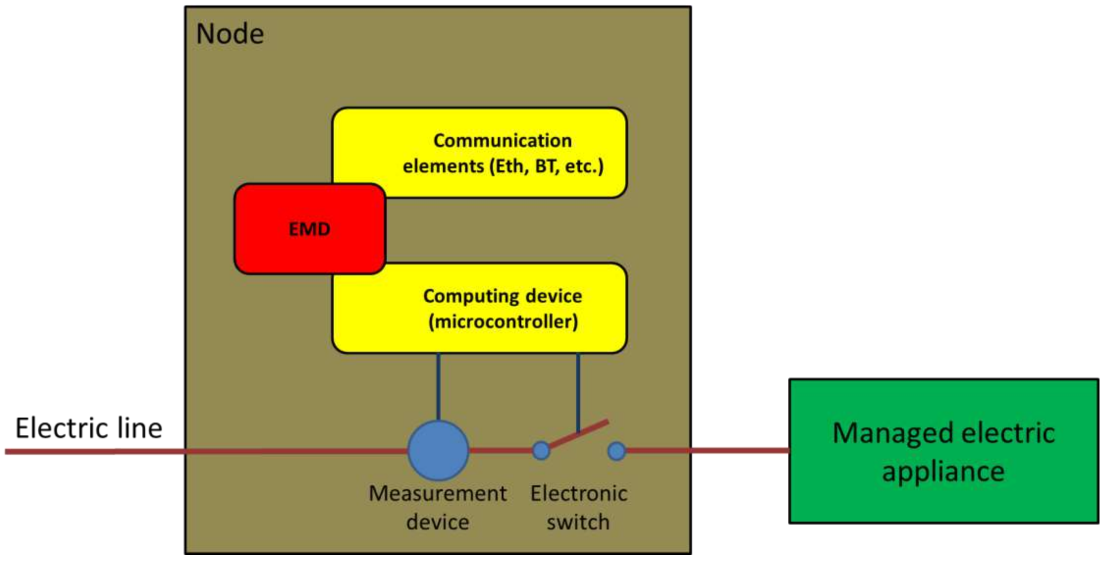
Node
Communication elements (Eth, BT, etc.)
EMD
Computing device (microcontroller)
Electric line
Measurement device
Electronic switch
Managed electric appliance

Figure 30. Block diagram of the EMD structure, as described in [85]

Computational capabilities: the proposal aims very specifically to be a scalable piece of Advanced Metering Infrastructure, so it has been aimed to be part of the AMI infrastructure that an end consumer/prosumer would have in their dwell or store. No other location of a Smart Grid deployment (TSO, DSO, Aggregator or power plant) is expected to be used to have the software components related to the middleware installed.

Message coupling: it is stated by the authors that their prototypes were working under CORBA [86] and the Internet Communications Engine (Ice, which provides a Remote Procedure Call protocol using standards from the transport layer [87]), so it essentially works under a Client/Server paradigm.

Middleware distribution: since the middleware solution is strongly linked to the hardware where it has been installed, it has been considered as a mostly centralized proposal that is running in a single kind of element (AMI) that is likely to be present in the facilities of many end users.

Bearing all these features in mind, the proposal can be described as:

$$SGM = SA\ (0) + CC\ (0) + MC\ (2) + MD\ (1) \qquad (19)$$

Advantages of the proposal: the authors have made a great effort in creating a realistic way to have a piece or AMI capable of establishing connections with other parts of any possible deployed infrastructure of this sort by using middleware in it. Besides, they take into account aspects like the cost or the size that are usually not considered for prototyping purposes in other pieces of work.

Disadvantages of the proposal: the proposal is too focused on the hardware components that have been mounted in order to have EMDs that can be considered as a scalable AMI, so the limitations of the proposal come from that fact. Moreover, the EMD shown here basically makes use of CORBA and Ice as the actual middleware implemented within it, so it is not possible to use any other standard or a more novel solution that what is being provided here. Therefore, in order to port the software elements to other part of a Smart Grid, a significant recoding work is likely to be needed. In addition to that, software-related features such as security, semantics for information enhancement, or offering a specific API to access the services that may be provided by the device, are not stated in the proposal.

2.2.24. OpenNode Smart Grid architecture

Florent Leménager et al. offer their own novel concepts for a middleware solution focused on the Smart Grid [88]. They show the implementation works that have been made as part of the OpenNode project [89]. In the proposal, it is shown that there is a collection of key concepts (modularity, extensibility, distribution of intelligence, an open common reference architecture, usage of open standards and cost effectiveness) of major importance by the authors. The efforts carried out in this proposal have been directed to creating an open development for Secondary Substation Nodes (SSNs) located within secondary substations of the power grid, and a middleware solution used to interconnected the SSN with the utility systems related to the electrical facilities and infrastructures. It is mentioned by the authors that the middleware, always regarded as something separated from the SSNs and running above them, is used to deal with the stakeholder diversification and the flexibility needed to interoperate among the power grid, the network and the management methods.

Service availability: judging from the information available in the proposal, the middleware is solely focused on hardware abstraction, as there is no information about the kind of services that could be encased in the middleware proposal or in any message format interchanged between the data-based levels and the lower ones. There are two separated roles for the new software components developed both at the middleware and the SSN level: the Secondary Substation Node is used for interacting with any smart meter or local Intelligent Electronic Device (IED), whereas the middleware is conceived as a way to interchange metering data, along with grid automation information. Thus, it can be argued that some of the SSN functionalities resemble the ones that are expected from middleware from the point of view of hardware interoperability. The open nature of the content that is provided by the project is of major importance for the project partners, as it is claimed that the project aims to provide a reference architecture publicly available to the community for the integration of Distributed Energy Resources. At the same time, middleware transfers information to an Enterprise Service Bus in case it might come in handy to be transfer to other utility management systems. The locations of the middleware and the SSN in the project have been displayed in Figure 31.

Computational capabilities: according to the tests that were carried out, SSN prototypes were developed by using industrial PCs and embedded Linux CPUs. Smart Meters from five different manufacturers are also used in the testbed. Since the middleware is going to be interacting with software resources of the Secondary Substations deployed in the power grid, it is expected that it will be deployed in the aggregator, TSO or DSO facilities (in fact, it is explicitly mentioned in the proposal that the time constrains based on operational DSOs have been defined in the testbed).

Message coupling: no information is provided about how data is expected to be transferred from one side to other one in the proposal. It can be expected that, due to its location, real-time information will be exchanged in the system.

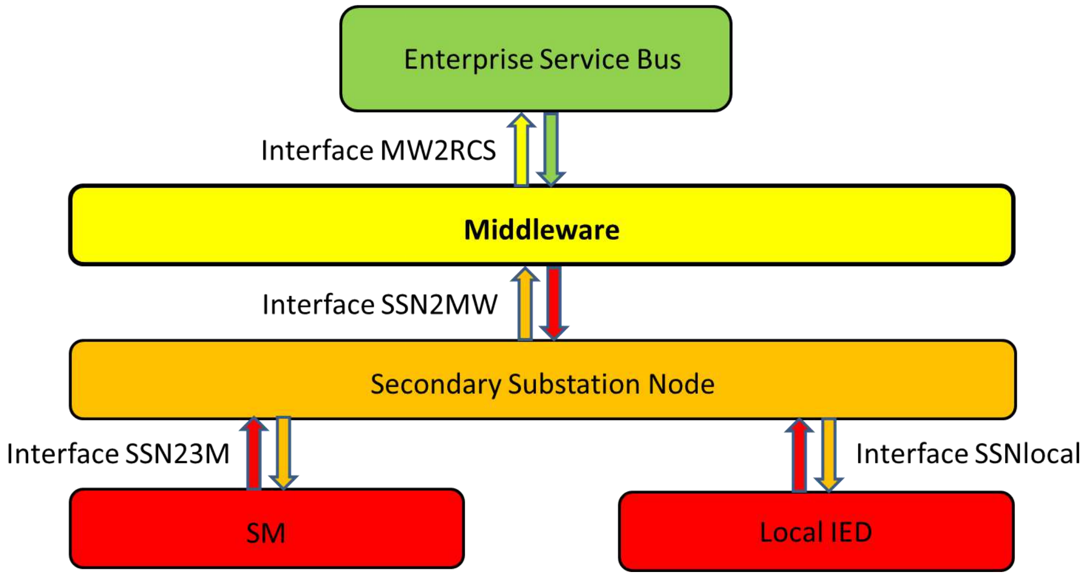

Figure 31. Middleware high level location in the proposal, as described in [88]

Middleware distribution: the proposal is a fully decentralized one, as there is not a prominent middleware location that will rule over other entities with middleware components installed and it is supposed to be interacting with SSNs scattered in a Smart Grid.

Consequently, this proposal can also be described like:

$$SGM = SA\ (0) + CC\ (1||2) + MC\ (3) + MD\ (3) \qquad (20)$$

Advantages of the proposal: the authors ensure that what has been developed will have a high degree of applicability to the power grid in order to enhance its performance. Furthermore, the proposal has been tested in several realistic scenarios where two physical and one virtual tests were carried out.

Disadvantages of the proposal: the proposal relies in middleware only to transfer information from the Secondary Substation Node to the Enterprise Service Bus that is used to interchange information with other utility deployments. Besides, data about major characteristics of the middleware are absent, such as how data messages are transferred from one area to the other or how to enable semantic capabilities or security as components of the middleware solution.

2.2.25. DIRECTOR

James Wilcox and Mahesh Sooriyabandara present what they refer to as a distributed communication transport manager for the Smart Grid [90]. It is expected from DIRECTOR that it will be able to manage the requirements of application communications throughout a Smart Grid. It has been placed by the authors between the application layer and the socket Application Programming Interfaces located right below the DIRECTOR middleware layer. It is claimed to instruct those sockets to generate an optimised interface that supports application needs to its best possible. Messages in the middleware domain contain the work payload, a list of destinations and the priority of the message.

Service availability: among what the proposal is capable of providing, hardware abstraction is the most prominent and obvious service. However, message orientation is also strong, stating that the socket configuration can be changed when sending them and different levels of bandwidth efficiency are required, so it has been regarded as an intermediation middleware. According to the authors, the DIRECTOR middleware is made up by several functionalities: an *application interface* (conceived as an inter-process communication transport socket), *network health* (offered by monitoring data exchanges over an Ethernet bridge), a *custom transport layer* (generated after considering the inputs provided both by the application interface and the network health information) and a software component used for *custom socket* (generated with negotiated transport features). The location of the middleware layer has been displayed in Figure 32.

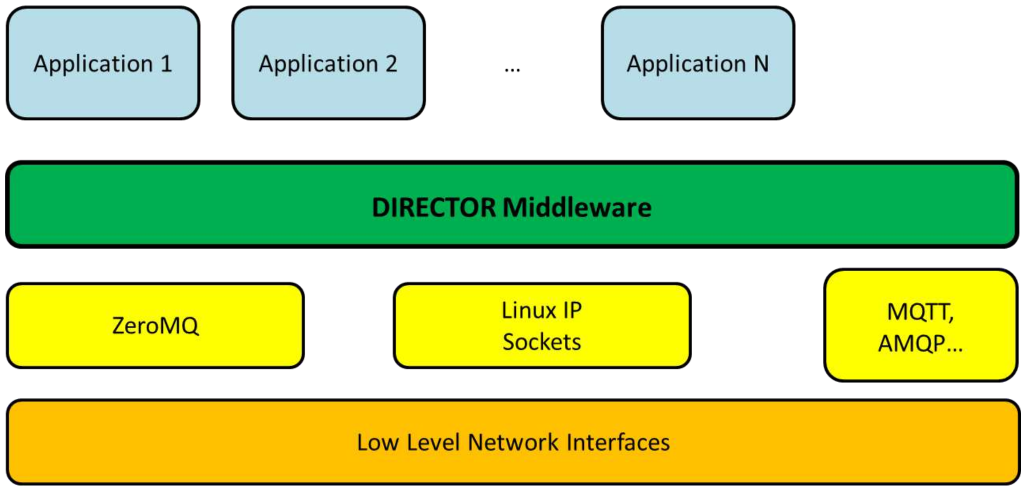

Figure 32. Location of the DIRECTOR proposal in a Smart Grid deployment, as stated in [90]

Computational capabilities: the proposal can be installed in almost any piece of equipment present in the Smart Grid, as tests were carried out in five Raspberry Pi platforms that, according to the authors and the specifications of the Raspberry Pi model B [91] have a 700 MHz ARM processor and 512 MB of Random Access Memory (RAM). Therefore, and considering that a smart meter can be implemented out of a Raspberry Pi, even hardware located at the end user´s facilities could handle the computational requirements of the middleware solution, let alone any other piece of equipment in the aggregator, TSO, DSO or power plant.

Message coupling: the proposal mentions that it has been conceived for distributed and real-time embedded systems, so it must be capable of transferring that kind of information. Furthermore, a virtualized Demand Response Automation Server has been used to simulate an actual Demand Response scenario.

Middleware distribution: it can be regarded as a fully decentralized middleware architecture that works under a peer-to-peer level, due to the fact that there is not a single component that adds an extra prominence when compared to all the others. However, this is due to the fact that no Publish/Subscribe (and thus, centralizing elements such as brokers) or any other message interchange paradigm is ever used.

Therefore, this architecture can be described as follows:

$$SGM = SA\ (1) + CC\ (0||1||2||3) + MC\ (3) + MD\ (3) \qquad (21)$$

Advantages of the proposal: DIRECTOR has been tested under a deployment using hardware easy to find in a more realistic scenario (such as a microgrid, a power plant, etc.). The solution makes use of a data model to exemplify how applications would work. Lastly, the location of the proposal in a distributed system is made very clear, as it is aimed to be placed between the application layer and the sockets that are used for transport layer communications.

Disadvantages of the proposal: there are no mentions about how security or semantic capabilities can be added to the proposal, or any service that may go beyond interoperability among different devices. Besides, the proposal seems to be just involved in the messages that are sent through the transport layer but shows less information about how the information is transferred to the application layer, or the kind of APIs that are available for that level. As far as message coupling is concerned, it is not easy to figure out the kind of criteria that has been used in the proposal to use one paradigm or another.

2.2.26. DDS interoperability for the Smart Grid

As it happened with a former one, the proposal of Tarek A. Youssef et al. makes use of Data Distribution Service as a way to provide middleware to the Smart Grid [92]. DDS is utilized combined with data structures and standard interfaces in order to have a scalable Smart Grid infrastructure that can be used in the testbed utilized to assess the feasibility of the hypotheses formulated. It is claimed that it makes possible integration with other deployments, experimentation, data gathering and algorithm testing. At the same time, DDS enables the system to establish a global Data Space where data structures are used to generate information that can be interchanged under a Publish/Subscribe paradigm. Interoperability is guaranteed by means of Real Time Publish Subscribe Protocol (RTPS), the DDS lower level layer used for this purpose, whereas an API is provided for the higher levels to access the middleware services.

Service availability: the proposal is bent on providing connectivity between the devices present in the testbed and the applications that would be requested by the end users. The authors focus on the idea of providing a gateway for each of the applications that are estimated to be required in a Smart Grid deployment. These gateways will be used as a way to provide an Application Programming Interface for the topmost level of the architecture to access the middleware facilities. Therefore, it can be said that this is an example of Message-Oriented Middleware rather than an architecture with several distinguished devices. The most foreseeable services to be included have been depicted in Figure 33.

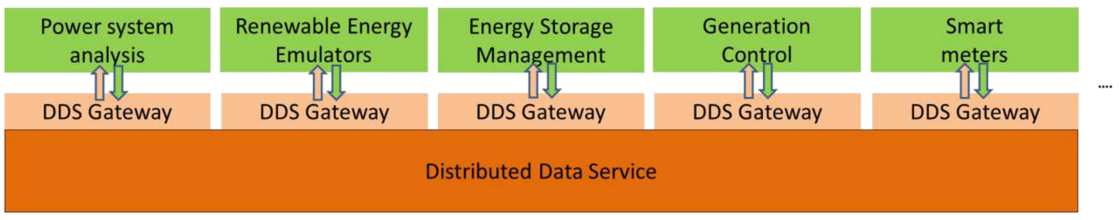

Figure 33. Testbed for the DDS infrastructure, as described in [92]

Computational capabilities: the proposal is expected to be installed in devices capable of handling DDS, so a regular computer should be able to handle it. Regarding the pieces of equipment expected to be included here, PC- or server-like appliances belonging to the aggregator and the TSO/DSO are likely to encase the software components of the proposal.

Message coupling: DDS strongly relies on a Publish/Subscribe paradigm to deliver data; consequently, it is used in this proposal as well, in order to deliver information related to the applications typical of the Smart Grid. In addition to that, real-time data is cited to be added as one kind of information that can be included in the deployment.

Middleware distribution: although there is not a clear mention about it, if the testbed that has been used by the authors is taken into account, the proposal can be expected to include and transfer data from end user devices. Thus, it can be located in any facility that is handling them, such as the aggregator infrastructure or the pieces of equipment installed as part of the TSO and the DSO.

This proposal can be further displayed as:

$$SGM = SA\ (2) + CC\ (1 || 2) + MC\ (0 || 3) + MD\ (1) \qquad (22)$$

Advantages of the proposal: as it happened before, many of the advantages and disadvantages of the proposal are linked to the idea of using DDS as the provider of the middleware architecture. DDS has been proven to be effective in several environments using CPSs and so it demonstrates that it can seamless interconnect the devices and applications of the deployment put forward by the authors of the proposal. In addition to that, a compelling testbed is provided where experiments can be run to test the feasibility of the ideas related to the proposal. In addition to that, an Application Programming Interface is available for any DDS development compliant with the standard.

Disadvantages of the proposal: the main focus provided by the authors is solely on interconnectivity of high and low level capabilities, so additional services to be included in the middleware (semantic capabilities, security) are not considered. Furthermore, due to licensing in the used proposals, it might be a problem to exploit the developed works done under a DDS implementation.

All the information that has been collected from the proposals is shown in a more summarized manner in Table 1. It is here where the advantages and disadvantages offered by the state of the art are immediately present, and an accurate idea of the developments done can be conceived.

Table 1. Main characteristics of the studied middleware solutions

Proposal name	Advantages	Disadvantages
GridStat	Detailed framework. Implementation works. Requires low computational capabilities.	No remarkable services available. No details regarding security implementations. No semantics.
Service-Oriented Middleware for Smart Grid	Collection of services. Performance tests. Security solutions conceived.	No remarkable services available. Scarce information about distribution. Vague boundaries defined.
Ubiquitous Sensor Network Middleware	Decentralization is enabled. Compatibility with a plethora of technologies.	Vague boundaries defined. Scarce information about the pieces of equipment where the proposal would be

Proposal name	Advantages	Disadvantages
		included. No performance tests are included.
OHSNet	Significant collection of services for hardware interoperability.	Vague boundaries defined. Services do not provide value for end users.
MDI	Precise and abundant information about devices to have the proposal installed and computational resources.	Very few data regarding implementation. Security or semantics have not been enabled.
IEC 61850 and DPWS	Information about semantic capabilities and implementation is provided. Security is provided.	No information about tests or hardware abstraction carried out.
IAP-INMS	Data heterogeneity is taken care of. An ESB is used. Tests have been made on the proposal.	No data about hardware abstraction or security.
Self-Organizing Smart Grid Services	High degree of distribution.	No data about how to use the proposal in a deployment. Scarce information about tests. No prominent services offered.
Secure Decentralized Data-Centric Information Infrastructure	Easily portable proposal. Networking and securitization capabilities. API is provided. Security is a major service in the proposal.	No semantic capabilities provided. The proposal is too much focused on transport and network layers.
A cloud optimization perspective	High degree of distribution. Parallel works can be done. Security can be included.	No API is provided. No data is offered about software elements of the proposal.
KT's Smart Grid Architecture and Open Platform	Platform as an open development. Realistic deployment done.	End users are regarded as just consumers in the proposal. Service securitization information is missing. No data about the API.
Smart microgrid monitoring with DDS	DDS is very suitable for middleware solutions. Computational capabilities are suitable.	Licensing can be a challenge depending on the purpose of the deployment. No additional, Smart Grid-related services are offered.
ETSI M2M	API available. Profuse information about prototyping. Successful port of another distributed systems solution.	Confusing exposition of the implemented services. Not many data about message transmission.
Smart Middleware Device for Smart Grid Integration	Tests carried out on Smart meter and devices with real technologies.	Main focus on encasing one piece of hardware with specific characteristics. No information about how data

Proposal name	Advantages	Disadvantages
		becomes distributed in the system is given. Security, API or semantics are missing.
WAMPAC-based Smart Grid communications	Security is described very accurately. Information about how to build a testbed is present.	No description provided about other services unrelated to security (semantics). No API is described.
C-DAX	Tests carried out in a real scenario regarding components and use cases. Security is enabled.	Other services aside from security are not described.
Building as a Service	An API and popular software technologies (JSON, SOAP, JDBC) are used.	No information about data semantics or securitization. A too specific scenario (energy consumption for buildings) may be challenging if the solution is going to be ported.
Middleware-based management for the Smart Grid	Great effort in improving Advanced Metering Infrastructure by means of middleware.	Solution extremely dependent on a single kind of hardware. Solution only proved with a small set of specific technologies (CORBA, Ice).
OpenNode Smart Grid architecture	High applicability to the power grid. Tests carried out in realistic scenarios.	Middleware designed for very specific functionalities. Information missing about important middleware services.
DIRECTOR	Tests done in realistic pieces of equipment. Clearly described distributed features.	Important services (semantics, securitization) are not mentioned. No API is available. It is difficult to know what kind of message coupling is used.
DDS interoperability for the Smart Grid	DDS is a suitable solution for middleware. Compelling testbed provided.	Only focused on interconnectivity of high and low level capabilities.

A further analysis on open source ESB solutions will be provided, as they have been found to be the most adequate solution to use in middleware for the Smart Grid.

2.3. State of the art in Enterprise Service Bus solutions

Among the several ways possible to find a way to embody middleware in an actual implementation, using an Enterprise Service Bus (ESB) platform is a suitable solution for the challenges that may be found in the Internet of Things. Typically, an ESB will contain several features to guarantee that interoperability can be provided:

1. Rather than having a massive number of point-to-point mesh-based communications between the integrated applications, which would be impractical and a source of delays, a bus is used to connect them all and transfer the data. This enables the good usage of the software entity used for data transfer.
2. An ESB hides away the heterogeneity and complexity of the services and applications that is interconnecting, so it fits in accurately with the purpose of middleware.
3. An ESB allows the usage of software packages (bundles) that make possible their port from one deployment to a different one, thus saving time and resources in development works.

The position of an ESB (and of middleware when it is embedded in an ESB) in a CPS-based deployment (with similar features that can be found in a Smart Grid from the software point of view) has been depicted in Figure 34. Note that apart from the services that it is interconnecting, the ESB is capable of containing some more if required (device registration, securitization), so it effectively becomes a way to implement a middleware architecture.

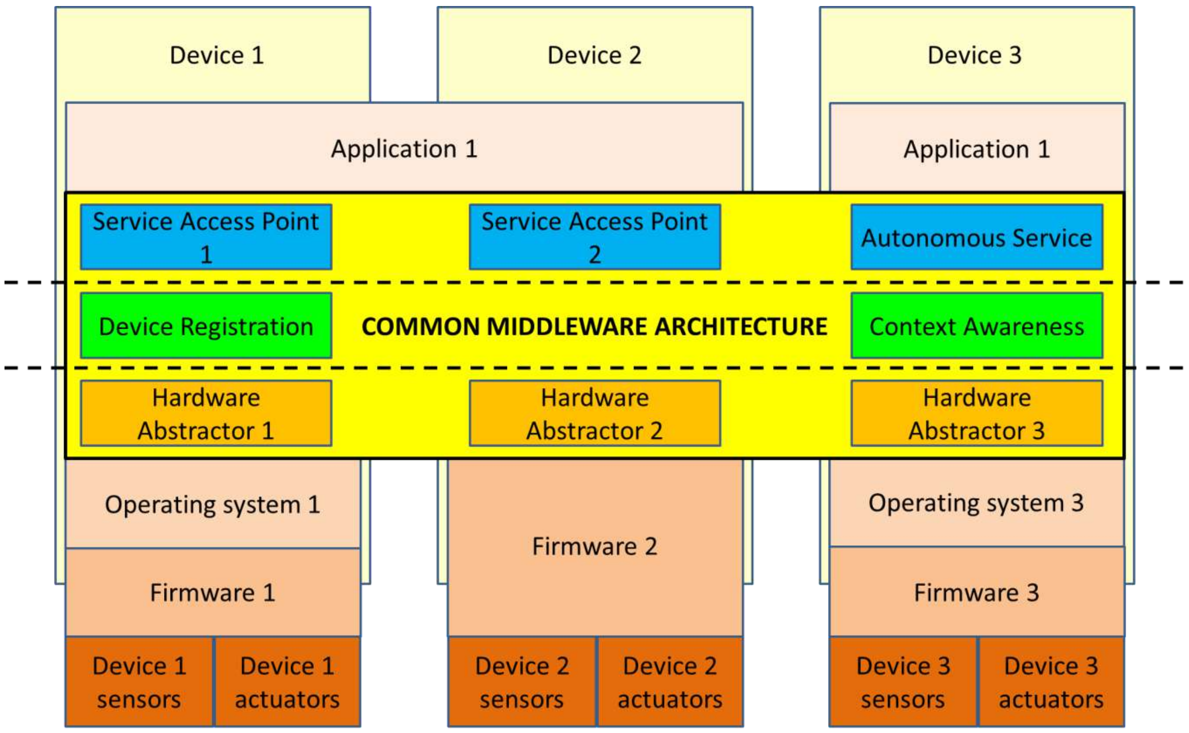

Figure 34. ESB location and most usual services in a Cyber-Physical System

The study that has been done about the state of the art in Enterprise Service Bus solutions focuses only on the ones that are offered under open source agreements. The motivation to do it like this is because two reasons:

1. From a purely economic point of view, it is much more convenient to use an open source solution, as they usually offer a number of services and interfaces enough for the idea of providing a middleware architecture for the Smart Grid. While open source

solutions usually offer a weaker support than commercial ones, the development and integration issues that may be found are usually likely to have taken place before, so there is usually a way to solve them. However, having a strong community behind an ESB solution is more important here than in other solutions were support is a paid service.

2. As far as the research projects where implementation works were done are concerned, a non-open source ESB was never considered as an eligible solution due to the European Commission policies, which favour the usage of open source solutions, and the utilization of inexpensive tools that will not challenge budget requests.

The most interesting options that have been found matching the previous requirements have been listed in the following subsections.

2.3.1. OpenESB

According to their developers, OpenESB is an open-source ESB written in Java and expected to be used as a platform with a dual purpose: enterprise application integration (the original purpose that ESBs were designed for) and a service-oriented architecture (much closer to what a middleware architecture represents), as mentioned in [93]. It is claimed by its creators that "*OpenESB is the unique open source ESB relying on standards that provides you with Simplicity, Efficiency, Long term durability, save on your present and future investments with a very low TCO (Total Cost of Ownership)*" [94]. OpenESB makes use of Java Business Integration (JBI) as a way to provide an architecture that will encase components for service production and consumption [95]. It is maintained by the OpenESB community that was created after Sun Microsystems was purchased by Oracle. This ESB has five different parts organized as follows:

1. Framework: it is an implementation of JBI claimed to be lightweight [96]. Even though it works in close cooperation with the used container, the framework remains container-agnostic, so one can be chosen among a plethora of them. The framework is encased in a Java Virtual Machine (JVM) that makes use of binding components to communicate with other frameworks. In addition to that, it is claimed to be fully manageable by Java Management Extensions (JMX)-related tools. Lastly, the framework is the part that contains the virtual bus (named Normalised Message Router) to be utilized in this ESB.
2. Container: the container can be regarded as the application server used to store the facilities that might be needed by the applications. OpenESB is claimed to be capable of processing more than 10 million complete messages per day [10] that use this container to be transferred. The current version of OpenESB has six different containers available: GlassFish Version 2 [97] (the one that is included by default), JBoss [98] Version 4, Version 5 and Version 7 and the vey used JVM.
3. Components: these are the software entities able to deliver the ESB the facilities used to transfer information. There are two different kinds of them: one is the Services Engine and the other one is the Binding Component. The former are used for receiving and sending messages to the bus, without dealing with any messages that have to be

sent outside it, whereas the latter interacts with messages outside the defined bus, but can generate either messages that will be sent inside the bus or messages transferred within the bus that leave it afterwards.

4. Integrated Development Environment and development plugins: it refers to the facility that is used for programming, debugging and testing the code that is produced. They usually include any plugging that is required for code generation. XML Schema Definition (XSD), WSDL, Business Process Execution Language (BPEL), Data Mapper and Composite Application editors are available to build, deploy, un-deploy, run, test and perform debugging tasks.

It is also noteworthy to mention that an OpenESB instance is able to work with both legacy solutions, which are dealt with by Domains and Node Agents working together with a Glassfish application server, and standalone stacks (that rely solely on an OpenESB instance making use of a Java Virtual Machine) [99]. A comparison of what is required for each of the iterations can be seen in Figure 35.

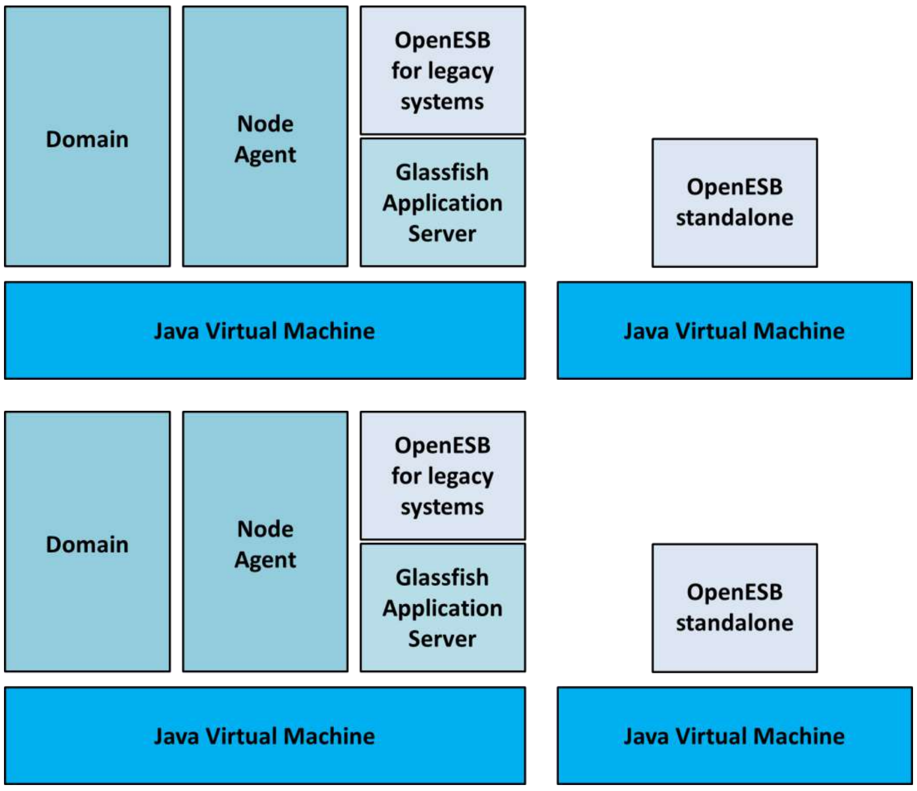

Figure 35. Legacy and standalone required software components, as described in [99]

As it can be inferred, OpenESB works in a way that implies that messages are sent from and to the applications accessing the bus where the legacy and the standalone stack are used. These messages will be sent via Binding Components and Services engines when they are required to travel inside the bus. All these components will be included in a selected container provided by a framework, and they will be worked upon by means of the Integrated Development Environment and the development plugins added to the distribution. One key feature of OpenESB is its openness; it is provided under a Common Development and Distribution License (CDDL) that makes possible providing a world-wide, royalty-free, non-exclusive license that enables the owner of the licensed software to use, reproduce, modify, display, perform, sublicense and distribute the original software [100].

OpenESB provides a good framework for services and delivers messages to the applications that require them. It has a foreseeable behaviour as it uses a bus to transport all the information required to interconnect the applications that have been distributed. However, the fact that is focused on one single programming language might be an issue for developers that are not proficient with that programming language.

2.3.2. WSO2 ESB

As it happened with the previous proposal, this iteration of the Enterprise Service Bus is claimed to be lightweight. In addition to that, it also has an open source development that explicitly regards itself as middleware conceptually relying on a SOA approach. Among the main features that are supported [101] the most important ones are:

1. WSO2 ESB makes use of WSO2 Carbon as the core of the platform, which provides a componentized architecture able to adapt business activity automatically in response to market events.
2. This proposal is expected to offer seamless adaptation among public, private and hybrid clouds, along with on premise solutions.
3. The developers of this proposal say WSO2 is optimized for the Internet of Things. The architecture offers a server-side and a cloud-based Reference Architecture so that the developers working on the IoT can use it as a starting point.
4. Flexibility is also a stressed feature: all the products linked to the proposal are said to be completely based on open source solutions and open standards. Legacy and package apps are integrated in WSO2, and the developers are able to extend the platform and customize it if required.
5. The possibility of having an Original Equipment Manufacturer (OEM) is also offered in order to offer subsystems for an end product provided by a third party.

Figure 36 shows the most important services that the WSO2 architecture is capable of providing; security for the deployed components, facilities for portals and stores (namely, asset storage), device management focused on mobile devices and the IoT, app development and management (working closely with what can be offered to portals and stores), API management (interweaved with the integration functionalities and the facilitates offered to portals and stores, as they require the services of an API to provide their own functionalities), integration by means of connectors with legacy systems and other facilities (along with service orchestration and composition) and analytics services. Note that several of them make use of other functionalities in order to offer their own, whereas there are some other services delivered in all the involved levels of the communication (such as security or analytics).