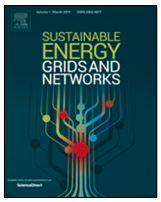
Sustainable Energy, Grids and Networks 32 (2022) 100856

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan



A comparative study on multi-agent and service-oriented microgrid automation systems from energy internet perspective

Md Tanjimuddin *, Petri Kannisto, Peyman Jafary, Mikael Filppula, Sami Repo, David Hästbacka

Faculty of Information Technology and Communication Sciences, Tampere University, P.O. Box 553, 33014, Finland

ARTICLE INFO

Article history:
Received 20 January 2022
Received in revised form 8 July 2022
Accepted 10 July 2022
Available online 16 July 2022

Keywords:
Energy internet
Multi-agent system (MAS)
Service-oriented architecture (SOA)
Microgrid
RIAPS
Arrowhead

ABSTRACT

The current advancements of energy, information, communication, and automation technologies and their integration have provided ways for the energy industry to transform into cleaner energy systems. This transition has contributed to the concept called energy internet. The recent energy technologies provide clean energy generation, storage and demand response through distributed energy resources. Information, communication, and automation technologies aim to provide supporting software tools and enabling mechanisms to automate the operation and control of those resources in a coordinated way. Thus, researchers and the software industry are developing software frameworks and platforms to support energy system automation. Commonly, most of the frameworks follow the design principles of either multi-agent systems (MAS) or service-oriented architecture (SOA). However, there are many frameworks and no straightforward criteria to select which one to implement in energy systems' automation applications to fulfill the energy internet vision. This study provides a conceptual investigation of MAS- and SOA-based software solutions by designing a use case for microgrid application automation considering its expansion for enabling energy internet. Two software frameworks, RIAPS and Arrowhead, have been selected as the candidates of MAS and SOA from the literature study. This study shows that neither MAS or SOA approach alone might not meet the requirements of microgrid automation and energy internet. Consequently, a combined approach of MAS and SOA is proposed.

1. Introduction

A transformation in the energy sector has been happening since the beginning of the 21st century, leading to concepts such as smart grid, energy internet and many others. The transformation aims to participate in the continuous effort of reducing societal side-effects of energy usage (e.g. worldwide carbon dioxide emissions) and improving the performance of power systems and markets. Aiming to reduce carbon dioxide emission, Finland is progressing towards a carbon-neutral country by 2035 [1] and the entire European Union by 2050 [2]. Energy usage in the form of electricity is a key contributor in this regard. However, the emission-free energy production and efficient usage of electricity are major challenges. To address these challenges where technology and market integration is leveraged in energy systems, the concept *smart grid* is considered as a solution to enhance the performance of a traditional power system and markets both operationally and investment-wise.

The flexible power system and market are essential since many DERs evolve and aim to integrate into the power system and participate in the market. Flexibility requires enhancing

Abbreviations: AMQP, Advanced Message Queuing Protocol; BEMS, Battery Energy Management System; BESS, Battery Energy Storage System; CIM, Common Information Model; CoAP, Constrained Application Protocol; DDS, Data Distribution Service; DHT, Distributed Hash Table; DPWS, Devices Profile for Web Services; DER, Distributed Energy Resources; DSO, Distribution System Operator; ESB, Enterprise Service Bus; EV, Electric Vehicle; FCR-N, Frequency Containment Reserve for Normal Operation; HEMS, Home Energy Management System; HTTP, Hypertext Transfer Protocol; ICT, Information and Communications Technology; IED, Intelligent Electronic Devices; IoT, Internet of Things; IIoT, Industrial Internet of Things; JADE, Java Agent Development Environment; JSON, JavaScript Object Notation; MAS, Multi-agent System; MQTT, MQ Telemetry Transport; OPC UA, Open Platform Communications Unified Architecture; PNNL, Pacific Northwest National Laboratory; PV, Photovoltaic; QoS, Quality of Service; REST, Representational State Transfer; RIAPS, Resilient Information Architecture Platform for the Smart Grid; ROS, Robot Operating System; SCADA, Supervisory Control and Data Acquisition; SSL, Secure Socket Layer; SOA, Service-oriented Architecture; SOAP, Simple Object Access Protocol; SoC, State of charge; TLS, Transport Layer Security; TSO, Transmission System Operator; WoT, Web of Things; XML, Extensible Markup Language; XMPP, Extensible Messaging and Presence Protocol

* Corresponding author.
E-mail address: md.tanjimuddin@tuni.fi (M. Tanjimuddin).

https://doi.org/10.1016/j.segan.2022.100856

the operation of variable and less controllable power production, minimizing the investments for peak power capacity, and avoiding market failures. The flexibility may also be utilized for many other purposes like grid management, customer energy cost minimization, and power system security enhancement. The management of power systems and markets based on flexibility originated from millions of small-scale DERs is fundamentally and organizationally different from the management of the traditional system, which requires more distributed decision-making than today. At the same time, DERs with distributed management enable novel functionalities, which have been expensive to realize in the traditional system concerning the enhancement of the profitability of flexibility management investments. At the local level of such an energy system, a microgrid concept has been introduced to realize the local management of DERs, enhance the utilization of DERs, and provide flexible services for external stakeholders.

Energy internet is a technological combination of energy and the internet, which is the predecessor of the smart grid concept. In other words, further development of the smart grid concept [3,4]. Energy technology consists of integration and utilization concept of DERs, power electronics technology, market mechanism, etc. On the other hand, internet technology consists of the Internet of Things (IoT), cloud computing, edge computing, Web of Things (WoT), etc. More precisely, the monitoring, control, and organization of energy technology-based power system at the physical layer with the help of internet technology at the cyber layer is an energy internet. The energy internet is a broader concept where heat, gas, electricity, etc., are considered a physical entity [3,5]. However, in this paper electricity network is considered as a physical entity and defined accordingly.

Microgrids are considered a promising concept for operating, controlling, and managing DERs in the energy internet [6–8]. One of the advantages of the microgrid is that it can operate either in an islanded or grid-connected mode. In the islanded mode, the microgrid operates independently to serve the local loads, enhancing the supply security for the customers connected to the microgrid. In grid-connected mode, the microgrid provides cost-effective energy for consumers, integrates energy storage and demand response to utilize generated electricity as efficiently as possible, and leverages grid connection capacity among DERs. In addition, the microgrid can play a grid-interactive role by providing ancillary services, such as frequency response and regulation, reactive power support, and voltage regulation in the grid-connected mode. Finally, the microgrid may participate actively in all energy and flexibility markets directly as a market participant or indirectly via a retailer or an aggregator. Therefore, the operation of the microgrid needs automation to enable automatic monitoring and activation of control functionalities and market participation. That plays a significant role in the energy internet.

To enable the development of the automatic microgrid operation and control functionalities in a coordinated way, an appropriate software framework for integrations is required. Two architectural concepts dominate the software solutions for the dynamic and heterogeneous manufacturing environment [9]. One is called multi-agent systems (MAS), whereas another is service-oriented architecture (SOA). Similarly, in smart grid applications, the usage of software solutions based on MAS and SOA are promising. Many MAS- and SOA-based software frameworks are available and emerging rapidly. Therefore, selecting a proper software framework for the application can be difficult from the range of the frameworks available. In literature, both approaches are investigated separately in different smart grid automation applications. Most of the investigation and implementation are conducted through a pre-selected software framework based on MAS or SOA. In other words, the literature aims to find out the applicability of software architecture for a particular application in the smart grid domain. Thus, the literature lacks a theoretical, systematic comparative study to support the selection of the software framework for microgrid automation application considering energy internet.

This paper presents a systematic comparative study on MAS- and SOA-based software frameworks based on one automatic power supply restoration functionality of microgrid defined in Section 3.1. However, this study does not confine to analyzing that functionality alone. Other aspects like DSO interaction and inter-microgrid communication are also considered. The contribution of this study lies in finding the benefits and shortcomings of the different architectural paradigms implemented by two candidate frameworks examined in this study. The MAS-based one is Resilient Information Architecture Platform for the Smart Grid (RIAPS [10]), whereas the SOA-based one is Arrowhead framework [11]. RIAPS was selected because it solves multiple shortcomings of earlier platforms, whereas Arrowhead excels at features for dynamic changes at run-time and a system consisting of subsystems. This study formulates a comparative analysis of the handling ability of advanced automation and control functionalities by the frameworks. The functionalities are defined from the use case selected for the study. The analysis focuses explicitly on the guarantee of solving scalability issues in terms of dynamic addition of entities, interoperability in terms of information exchange and communication mechanisms, security in terms of the level of encryption, the secure communication and integrity of the information, real-time communication support, as well as resiliency in terms of fault management in the physical device, software services level, and application level.

The rest of the paper is organized as follows. A brief overview of the concept of energy internet, MAS and SOA, related comparative studies about MAS and SOA, and a review of MAS and SOA based frameworks are described in Section 2. Section 3 presents the use case definition and its functional, information, communication, and cybersecurity requirements for the implementation. Section 4 presents a brief presentation of RIAPS and Arrowhead and their utilization in the defined use case. The criteria for comparing both software solutions is formulated in Section 5. Later, analysis and comparison are made in Section 6, followed by the proposed combination approach of MAS and SOA in Section 7. Finally, the discussion and conclusion are in Sections 8 and 9 respectively.

2. Background and related work

2.1. Conceptual foundation: Energy internet

The concept of energy internet is relatively new and emerging gradually. Consequently, several definitions and interpretations of the energy internet exist in the scientific community. However, most of the research work found in the literature highlights the concept of merging energy and internet, information, and web technologies to form a new ecosystem [5,12–14]. This ecosystem consists of emerging information technologies such as IoT, Big data analytics, cloud computing, etc., and energy technologies such as the integration concept of large-scale distributed generation, the intelligent device for distributed generation to enable interfacing, energy storage technologies, etc. In other words, the energy internet is a complex cyber–physical system where the physical layer consists of energy technologies, and the cyber layer consists of the internet, information, communication, and web technologies.

Though the detailed discussion of the energy internet is out of the scope of this article, the main characteristics of the energy internet are described here briefly to provide a foundation.

- Openness [13,15]: The idea of openness is to provide accessibility. Accessibility means operating energy resources globally or locally, which opens new business possibilities and makes the energy system more sustainable, reliable, and efficient. Therefore, information and physical interface should be open.
- Plug and play [4,14,16,17]: Plug and play is the core idea to develop energy internet. This feature enables electrical devices to connect anywhere to the energy system anytime with less human intervention. Thus, electrical terminals in the energy internet should have interoperable energy, communication, and information interfaces.
- Intelligence [15,18]: Renewable energy resources are the main foundation in the energy internet, and most of them are intermittent. To operate them in a safe, secure, reliable, and organized way, the energy internet focuses on providing enough intelligence on the edge devices to realize self-diagnosis, self-healing, distributed, and adaptive control.
- Business [12,19]: Energy internet is evolved to explore new business opportunities in the energy industry. The application of big data and managing the distributed renewable resources within the concept of microgrid, virtual power plant, energy communities, and prosumers open new business possibilities.

2.2. Conceptual foundation: MAS and SOA

In MAS, the core concept is agent, an entity capable of autonomous decision-making and communication. The agent is a computer system with sufficient software and communication support attached to the end devices to perform an intelligent task. When several agents in an environment are connected to perform application-specific tasks, formed a MAS. According to [20] an agent can exhibit autonomous, social, reactive, and proactive properties based on design objectives. The autonomous behavior shows the self-decisive capability of agents based on the situation. Social behavior includes interaction and coordination between the other agents to achieve design objectives. Reactive property shows the dynamical fitting to the environmental changes on time. Finally, performing each task to reach the final goal by an agent shows the agent's proactivity.

SOA is an architectural pattern used in distributed software systems, including industrial automation. The basic building block of this architecture is service. A service comprises a group of functionalities that appear as a contract. According to Erl [21], the design of a service follows following design principles.

- Abstract: Hiding the implementation logic and only exposing the details necessary to make the service usable.
- Autonomy: Services have control over their own implementation logic and do not bind other services to use specific dependency in order to utilize the service.
- Reusable: The design of a service enables reuse in various applications.
- Loose coupling: Services should be independent form one another. If one service needs to modify, it will not break other services.
- Composability: The embedding ability of one service into another makes the services composable.
- Discoverability: The published services should be discoverable by clients.
- Statelessness: Services are designed not to maintain state information to maximize resource utilization and scalability.

Web technologies such as SOAP (Simple object access protocol) and REST (Representational State Transfer) are the most common, enabling service-oriented design. SOAP utilizes extensible markup language (XML) for data format and Hypertext Transfer Protocol for messaging. On the other hand, REST is currently the most used architectural style for making web services. Representing services as resources is the main idea of web service applications in REST style. The most common data serialization format and communication protocol utilized in REST are JSON (JavaScript Object Notation) and HTTP (Hypertext Transfer Protocol). However, other implementations are possible, such as XML as serialization or CoAP (Constrained Application Protocol) as communication protocol.

2.3. Comparative studies about MAS and SOA

The comparative study on MAS- and SOA-based frameworks or platforms are found in very few pieces of literature. This section provides a review of related research focusing on comparative studies on MAS- or SOA-based software platforms and frameworks. In [22], a comparison of frameworks and a specific use case of the smart grid has been conducted. However, the evaluation is only limited to MAS-based software. [23] compared four multi-agent platforms by selecting evaluation criteria based on the platform development stages. However, this study only focuses on multi-agent platforms, and the evaluation criteria are not based on the features required for the smart grid automation application. In [24], the authors presented a comparative review of 24 available agent platforms. Although the authors selected 28 evaluation criteria, the selection of the criteria is universal, not domain-specific. In [25], the authors set up the software and interface requirements for two industrial use cases, and then flexibility and the dynamic configuration of services requirement is solved using the pre-selected Arrowhead framework. However, the study lacks evidence that Arrowhead were the superior candidate for such a requirement.

To conclude, no previous study has been found which provides a comparative study addressing the following combination:

- The selection of MAS candidate and SOA candidate for the comparison through a literature study.
- Application domain is energy internet.
- The comparison is between MAS and SOA frameworks or platforms.

Fulfilling this research gap is the outcome of this study.

2.4. Review of MAS- and SOA-based software frameworks

The application of MAS in power systems has previously appeared in multiple studies. Within last ten years, the multi-agent research in power systems has based on obtaining the smart features of smart grid. In [26], the authors presented several use cases where MAS is utilized as the conceptual basis of an automation platform. In [27,28], MAS enables the automatic fault detection and isolation in power systems, and in [28] system interaction follows a decentralized architecture. The use of a multi-agent system in microgrid control is reviewed in [20], which identifies a few microgrid-related application areas where the MAS-based architecture can be utilized. Furthermore, the work mentions a few MAS-based application development platforms, including Java Agent Development Environment (JADE), ZEUS, and VOLTRON. JADE is an open-source software framework for MAS-based applications. It has been applied in several microgrid applications in a distributed environment [29–32]. However, in [22] argue that, JADE might not be a well suited platform over RIAPS and VOLTRON, in terms of platform performance

and response time for the use case of under frequency load shedding scheme. ZEUS is another platform for MAS-based systems but no longer provides any support [20]. VOLTRON is a MAS-based platform originated from Pacific Northwest National Laboratory (PNNL), targeting at power system applications with a MAS-oriented approach [20]. Additionally, it is a message-oriented, modular, open-source platform suitable for the IoT environment [33]. The data-bus-oriented, MAS-based open-source platform called OpenFMB aims to solve interoperability issues in the power system. Along with those mentioned above, some other MAS-based frameworks, such as Robot Operating System (ROS) and Agent factory, lack real-time synchronization features for time-critical applications, variety in communication patterns, and a selection facility for the leader agent based on the need [34]. However, the shortcomings of these platforms have been solved in RIAPS. RIAPS is an open-source, MAS-based software platform originally designed for managing the two-way flow of information and power in a distributed, interoperable manner with real-time support [35].

The application of SOA has been studied in several cases in smart grid automation. The evolution of Common Information Model (CIM) standard and its usage in power systems support software design in a service-oriented way. In [36], a coordinated voltage control use case is implemented with CIM for information exchange. The service integration is done using Enterprise Service Bus (ESB) to realize a distribution grid management system. The ESB lacks easy scaling in a system of system application as it needs to implement an adapter for delivering data in the bus cost-effectively. However, integrating services using CIM data format provides improvements to utilize ESB [37]. In addition to that, the representation of data types in CIM is a power system resource that facilitates the RESTful design of services [38]. A conceptual framework for information integration in power systems is proposed in [39]. The integration is done through a web service infrastructure. A web service-based Supervisory Control and Data Acquisition (SCADA) system is analyzed using this integration infrastructure to discover different substation and control center services. However, this study fails to provide an implemented framework from that conceptual framework. In [40], various services and their interface are designed using a RESTful architecture to facilitate microgrid applications, but this lacks a service registry and allocation mechanisms. In [41], Open Platform Communications Unified Architecture (OPC UA, a service-oriented communication framework) is analyzed in smart grid applications deriving the data model mappings to CIM and IEC 61850. The conventional OPC UA is based on client–server communication, which can hamper scalability if the server becomes a bottleneck [42]. Thus, OPC UA has been extended to support the publish–subscribe communication pattern, which also needs support from a message broker technology for the routing of messages [43]. The advantages of the publish–subscribe-based architectures have been discussed in [44].

The availability of a software framework for developing the service-oriented application in microgrid automation is essential. The framework enables interconnection between the services, systems, and devices in a microgrid in an interoperable manner, providing security and flexibility. In industrial automation, several frameworks have been developed in recent years. For example, Arrowhead [11], FIWARE [45], Eclipse BaSyx [46], AUTOSAR [47], and IoTivity [48] implement a service-oriented architecture. Furthermore, since the microgrid is a subsystem of the smart grid and the concept of energy internet, the framework should have the following properties.

- Security by separating an independent subsystem from other subsystems.
- Interoperability to facilitate information exchange between different subsystems.
- Capability to dynamically allocate services for the service consumers and enable distributed control.

Among SOA frameworks, Arrowhead is the most promising for microgrid automation. It enables automation in separate, independent subsystems called the local cloud. Furthermore, its orchestration service can provide dynamic configuration of services at run-time and enable the application to perform in a distributed way. Finally, the gatekeeper service in arrowhead allows inter-subsystem service exchange utilizing interoperable information exchange. Other frameworks lack all the combination of properties. For example, FIWARE, Eclipse BaSyx, AUTOSAR, and IoTivity do not follow subsystems' separation principles and enable distributed automation due to their centralized implementation of the message bus. Therefore, Arrowhead is selected as the candidate framework to compare and analyze with the MAS-based framework.

3. System overview

The microgrid is an operational concept where intelligent prosumers, consumers, or resources operate in a coordinated way to meet energy demand inside the group. The microgrid can be realized at, for example:

- An industrial site, shopping center, or remote island where the distribution grid is not present.
- An industrial site, shopping center, or small household where the distribution grid is present.

In both cases, the microgrid owns the grid to accumulate the power generation and serve the loads. However, in the latter case, the microgrid can operate in a grid-connected mode. Whenever needed, they can provide ancillary services to the distribution grid. Since the microgrid concept relies on its own established grid inside, in that sense, it has operational independence. In the use case of this paper, the microgrid operates in a shopping center that has a power network of its own, consisting of a battery energy storage system (BESS), photovoltaic (PV) generations installed on the rooftop, and several controllable or non-controllable loads as well as charging possibilities for electric vehicles (EV). In addition to that, the microgrid has a point to connect with the local distribution system when necessary, for its own need or serving the requests of the local distribution system. The network topology of the microgrid is illustrated in Fig. 1. When the microgrid operates in the grid-connected mode, there should be intelligence to detect faults at the upstream network and act in a coordinated way to aim at optimal economic operation and protect the microgrid from endangering maintenance workers, instability (i.e., the unregulated voltage and frequency of distributed generation), and protection malfunctions. The blackout detection and secure separation of microgrid and power supply restoration needs automation through software solutions. The power supply restoration use case in microgrid is introduced to utilize an agent-based software platform (RIAPS) and a service-oriented software framework (Arrowhead).

3.1. Functional requirements

The MAS and SOA-based solutions in a microgrid are assessed by defining a generic use case. The use case is defined here to examine how our selected software solutions perform microgrid control functionalities. The functionalities are defined for microgrid blackstarting, i.e., power supply restoration when a full blackout has occurred.

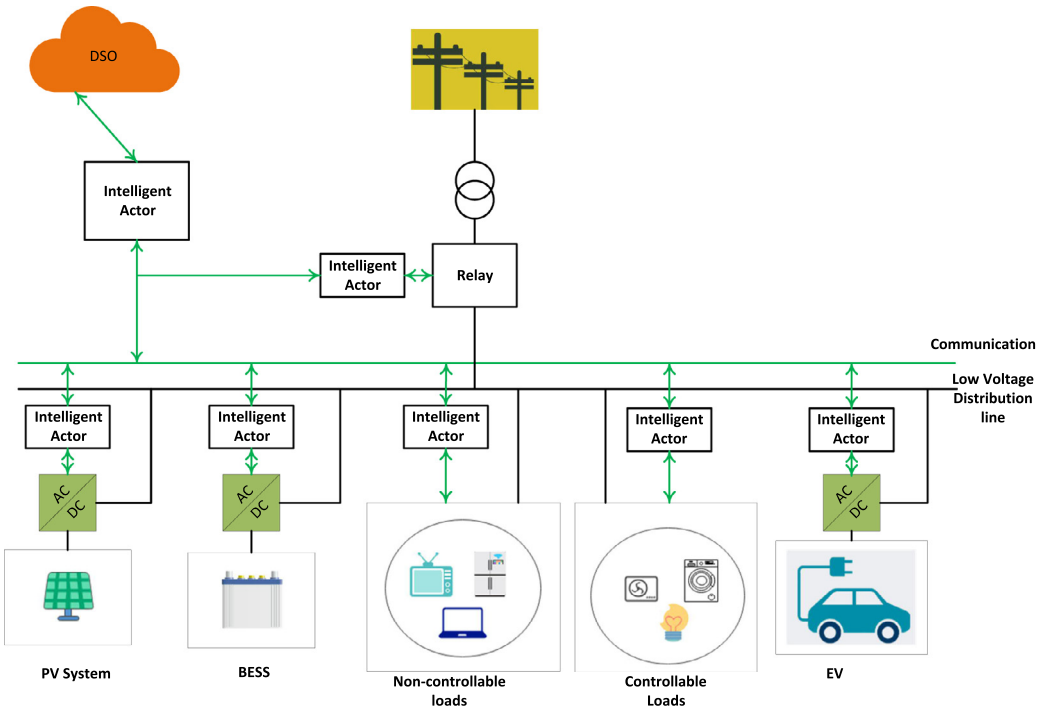


Fig. 1. Microgrid system overview.

The blackstarting of the microgrid is one option to restore the power supply. When a blackout happens in the main grid, the microgrid might fail to island and continue its operation smoothly. Blackstarting may happen locally in a sequential manner to restore power supply in the microgrid. Conventionally, a sequential blackstarting is performed with the help of a centralized microgrid controller and its coordination functionality. The centralized controller, consisting of software modules, is mainly responsible for setting up control rules and monitoring the condition of the blackstarting process.

Because centralized control introduces a single point of failure, this use case specifies a distributed approach for blackstarting. However, even though the approach is distributed, it is impossible to remove certain centralized functionalities that are necessary for protection and stability. Therefore, the blackstarting agent and its monitoring functionalities utilized in the use case are somewhat analogous to the centralized microgrid controller.

Fig. 2 illustrates the participants and their interaction as a sequence, showing main success scenario of distributed blackstarting described below. The primary grid forming resource is the responsible to form the grid which is selected collectively by exchanging information between all available candidates. After the grid formation, all the remaining candidates can be participated in blackstarting as grid followers.

1. The blackstarting agent receives the status of the circuit breaker through the isolation actor.
2. The blackstarting agent detects the blackout situation. The preconditions for blackstarting are the status of the circuit breaker "open" and the microgrid does not have voltage. If these preconditions are met, the blackstarting process may continue. If the preconditions are not met, the blackstarting agent continues monitoring the situation.
3. The blackstarting agent publishes blackstart initialization requests to all the actors inside the microgrid.
4. The actors start their initialization processes:
   - The actor changes the parameters of control mode, internal controllers, and protection settings for predefined settings, capable of blackstarting and island operation.
   - The actor starts blackstart/island operation monitoring functionality (publishes its state measurements: voltage, frequency, current, active power, reactive power, state of charge (SoC), etc.).
   - The actor stores the pre-blackout measurements as an estimate of its state.
   - The actor disconnects itself from the microgrid.
   - The actor publishes an initialization acknowledgment.
5. The blackstarting agent receives the initialization acknowledgments published by the actors involved in the microgrid. The acknowledgment contains the following information:
   - Status: actor disconnected, connected, not connected (e.g. EV unplugged), etc.
   - Role: grid forming, grid following, active load, passive load, etc.
   - Capabilities: available control services (e.g., primary frequency control, primary voltage control, inertia emulation, secondary frequency control, secondary voltage control, etc.), load connection steps (e.g., ventilation load consists of variable-speed motor load (frequency converter) + induction motor load), etc.
6. The blackstarting agent waits until all actors are disconnected and prepared for blackstarting. If the actor has not replied within the waiting time, it may apply force disconnection to the specific actor.
7. The blackstarting agent defines if the blackstart of a microgrid is technically able to realize based on available resources and capabilities.
8. If all actors are disconnected, and technical capability for a blackstart exists, then the blackstarting agent publishes a blackstart "start" message.

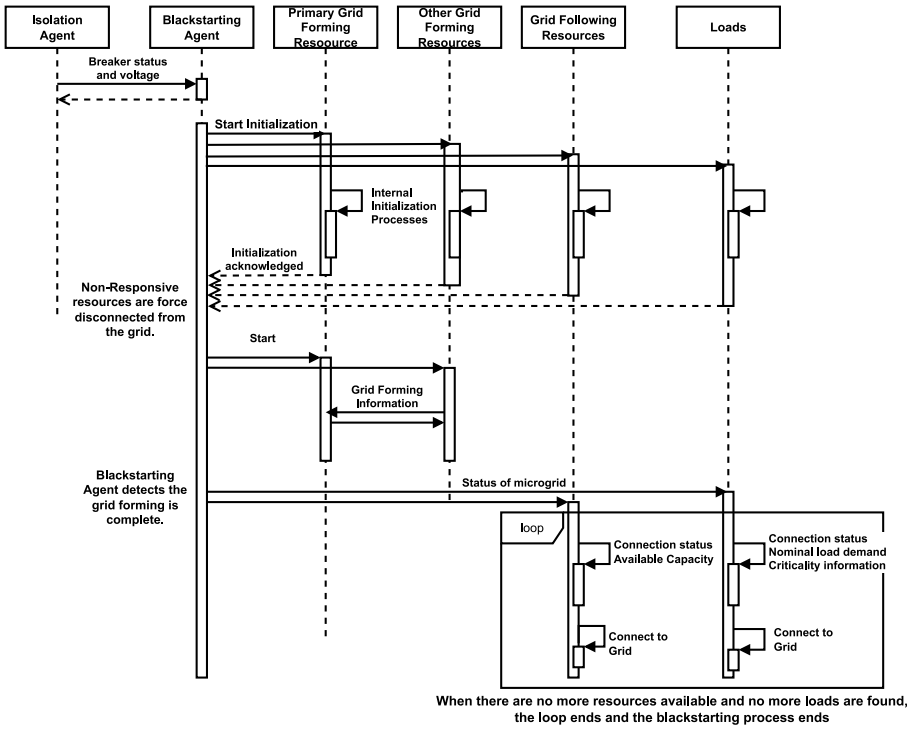


Fig. 2. Sequence of interaction and exchanged information among the participants.

9. Grid forming resources, those who are interested in forming the grid receive the blackstart "start" message to configure their state measurement messaging accordingly (e.g., SoC or available generation capacity, voltage, frequency, etc.). The messaging is continuous, and it has already started on step 4 to exchange the state measurements of the actors. The configuration will define which state measurement messages the grid forming resources should receive from the other grid forming resources available. The grid forming resource with the highest amount of SoC will be selected as the primary blackstarting unit. The selection is done by themselves by comparing their own value with others.
10. The primary blackstarting unit will initiate the blackstarting, i.e., it forms voltage to the microgrid in a no-load condition.
11. The blackstarting agent checks whether the grid is formed or not (Measuring microgrid voltage and frequency).
12. If the microgrid is formed, then the blackstarting agent publishes the status of the microgrid as "started".
13. Grid following resources and loads interested in connecting to microgrid receive the status of the microgrid as "started" message to configure their messaging. Every actor needs the following messages from all other actors available:
    - The state measurement (or estimation of actor state at the beginning of the process) of all actors.
    - Connection status and the capacity available (the maximum capacity limit minus output and output minus the minimum capacity limit) of all connected grid following resources.
    - Connection status, nominal load demand, and criticality of all connected loads.
14. The load to be connected is defined collectively based on the criticality information of load actors and the technical capability of the microgrid.
15. The load to be connected checks the state of the microgrid (usually by measuring their connection point values: voltage, frequency, current, etc.), if the state is acceptable for the load. The load is connected if the state is acceptable. Otherwise, nothing is done.
16. The grid following resource to be connected is defined collectively based on the capacity need and technical capability of the microgrid. These are calculated based on the information published by the grid following resources and loads already connected to the microgrid.
17. The grid following resource to be connected checks the state of the microgrid (usually by measuring their connection point values: voltage, frequency, current, etc.), if the state is acceptable for the resource. The resource is connected if the state is acceptable, otherwise nothing is done.
18. When there are no more resources available and no more loads are found, the loop ends and the blackstarting process ends.

3.2. Information requirements

From the functional requirements, multiple requirements arise regarding the information to be communicated. Table 1 summarizes these requirements explained in the previous section and sequence of interaction including exchanging information is presented in Fig. 2. The table columns indicate the source of information, the information to be communicated, and the destination or recipient of the information. In general, the information consists of real-time electrical measurements and control commands.

To contribute to interoperability within the energy domain, the information models should preferably comply to a standard. This should cover both the logical information structures and the data model, that is, the concrete serialization. The potential standards include at least IEC 61850, which specifies information structures for the interfaces of substation automation systems.

Table 1
Information exchange in the microgrid system.

Source	Information object	Information content	Destination
Isolation actor	Status (breaker)	Status: open or closed	Blackstarting agent
— " —	Voltage	Voltage value	— " —
Blackstarting agent	Start initialization	"Start initialization"	Grid forming, grid following, and load
Grid forming, grid following, and load	State measurement	Voltage, frequency, active and reactive power, and state of charge	— " —
— " —	Initialization acknowledgment	Roles: grid forming or grid following, active or passive load, status (disconnected, connected, or not connected), capabilities	Blackstarting agent
Blackstarting agent	Start	"Start"	Grid forming resource
— " —	Status of the microgrid	"Started"	Grid following, load
Grid following resources	Connection status	Status: (disconnected, connected, or not connected)	Grid following resources
— " —	Available capacity	Maximum limit output and minimum limit output	— " —
Load	Connection status	Status: (disconnected, connected, or not connected)	Loads
— " —	Nominal load demand	Power	— " —
— " —	Criticality information	Connection priority represented as a number	— " —

CIM, specified in IEC 61968 and 61970, is another alternative as it provides information structures for energy management systems. Due to the complexity of information models, this article leaves them out of scope. Still, it is acknowledged that the microgrid system should enable standard-based information models applicable in interfaces within system.

3.3. Communication requirements

In the blackstarting use case, the microgrid aims to restore power supply when a blackout happens inside the islanded microgrid, which necessitates communication. The communication facilities should enable interaction between all the actors in a distributed manner. According to [49] the design of the communication architecture needs to consider following:

- Supporting the communication protocol and any other communication-related standards of the equipment involved in the system.
- The structure of the control system can be centralized, decentralized, or distributed.
- The physical location of the DERs, as it depends on the location which type of communication link is available.
- The level of control in the control hierarchy of the microgrid and the tolerable latency.
- The restrictions set by existing technologies, e.g., data packets and the size of the measurement data and control commands.

The system should enable both the native protocols of the equipment and internet-capable publish–subscribe protocols, at least allowing adaptation between these when required. To meet the functional requirements of the microgrid, the required devices are inverter and intelligent devices. These devices can support Modbus or IEC 61850 for communication. Since the control architecture is distributed, it is evident that each actor needs to exchange its information asynchronously over internet. This requirement can be met with a messaging platform that implements the publish–subscribe communication pattern. The suitable communication protocols include MQTT (MQ Telemetry Transport), AMQP (Advanced Message Queueing Protocol) or XMPP (Extensible Messaging and Presence Protocol), among others.

Most of the information exchanged between the actors are switch status messages, measurements, and control commands. This must consider the limitations of the technologies in use. For example, if IEC 61850 is applied, the information signals are converted to data packets within the size of a maximum of 27 bits [50]. Moreover, additional bits are required to utilize those data to communicate via the network in a secure way. Depending on the communication protocol and its security mechanism, the total size of the message is large and can vary from one protocol to another. For instance, the maximum size of the data packets that exist in the microgrid communication network is 200 bytes when messages are encoded to the IEC 61850 data format [50].

The functional requirement is to restore the power supply in a shopping center after a blackout, assuming that all the process level types of equipment are inside the shopping center and networked in a way to form a local area network. The most time-critical communication feature is blackout detection and the submission of initialization messages to all the actors involved in the system. Additionally, the control functionalities involved in blackstarting are secondary control that requires fast communication, allowing a maximum of two-second delays [50]. Regarding connectivity, it depends on the location of the field devices and message bus. Depending on location WiFi, or Ethernet communication can be used.

3.4. Cybersecurity requirements

In a microgrid, communication must be secured to ensure the correct and authentic operation of the microgrid automation applications such as blackstarting. In modern automation solutions, components communicate by exchanging standard-based messages through an open networking infrastructure in order to provide interoperability. Although this enables efficient integration of DER and facilitates energy internet objectives, it also exposes the microgrid's data to cybersecurity threats. Therefore, cybersecurity measures must be applied to ensure data authenticity and consequently reliable operation of microgrid in energy internet architecture.

The cybersecurity measures aim to provide data Confidentiality, Integrity, and Availability which are defined as the high-level security requirements for the smart grid information networks. Confidentiality protects sensitive data from unauthorized access. For example, the power consumption status of a consumer is exchanged in microgrid power supply restoration process, which is a consumer's private information and must be confidential. Integrity prevents unauthorized modification of data. In microgrid

power supply restoration processes like blackstart, the controller reference point, voltage, and power setpoint are exchanged. Integrity is crucial in microgrid information exchange because invalid reference points, unauthorized setpoint changes, or wrong state measurements may lead to instability or serious protection issues. Finally, the availability attribute of security ensures access to data when needed. The availability is of utmost importance for the microgrid automation that is responsible for controlling and monitoring physical processes. The unavailability of data might create unnecessary system shutdowns or endanger human safety.

In Fig. 1, communication can be categorized into local and remote, which determines the availability of security-related communication tools. The local communication is accomplished in customer premises and related to primary controllers (e.g., inverter and DER). The local communication can be secured by using a secured version of the device-level communication protocols. For example in case of IEC61850-based communication, secured messages can be created by implementing security mechanisms defined in IEC 62351 standard. The remote communication occurs between microgrid components as well as between the microgrid and Distribution System Operator (DSO) when the microgrid operates in grid-connected mode. This remote communication can realize secondary and tertiary control [51] applications in which data communication is needed among distributed components, i.e., intelligent actors, and DSO actor in Fig. 1. These components are equipped with IoT functionality and implemented based on MAS or SOA frameworks. The IoT devices exchange messages over communication network where cybersecurity (particularly data integrity and confidentiality) becomes important.

The remote communication can be protected by creating secured messages or a secured communication path depending on several factors, such as the resource constraints of endpoint devices, IoT protocol, and IoT application design. The well-known security protocols like Transport Layer Security (TLS) and IPsec can be used for securing the remote communication to provide data integrity and confidentiality. If the IoT devices have not enough computational power for running TLS and IPsec, they can use other security frameworks and approaches such as a ticket-based method [52] to satisfy the cybersecurity requirements. Alternatively, a security gateway can be used as an intermediary device in communication between IoT devices. This gateway has more processing power and adds security mechanisms to the messages exchanged between IoT devices. Additionally, in order to provide data availability, IoT devices should be protected against security attacks (e.g., Denial of Service) that endangers availability.

The important aspect in cybersecurity of microgrid is to consider resiliency as well. Microgrid automation systems are critical cyber–physical systems. In these systems, cyber-attacks to communication may not only modify critical data but also lead to undesired switching of microgrid mode or damage to the electrical network environment. Therefore, cybersecurity should be understood widely [53] in this context and the automation system should not only be cyber secured but also resilient. Cybersecurity and resiliency are two distinct but complementary areas in which resilient system aims to maintain its functionality even during or after detection of cyber-attacks.

4. RIAPS, Arrowhead and their utilization

This section introduces the agent-based RIAPS integration platform and the service-oriented Arrowhead framework. A presentation of the features and characteristics that could benefit microgrid information exchange and control are covered and utilized in the designed use case accordingly.

4.1. Resilient Information Architecture Platform for the Smart grid (RIAPS)

RIAPS is an open-source software platform aiming to support distributed control and computing applications in the smart grid through a software foundation. The software foundation is composed of a component building framework and platform management services. A set of libraries has been developed to develop the application components, secure interaction between the components, scheduling, life cycle management, and data logging and persistence. In addition, the component framework provides fault management services in order to detect and mitigate faults in components. On the other hand, the platform management services include discovery, device interface, time synchronization, distributed computing, application management and deployment, security, and resource management. The details of RIAPS platform can be found from [54].

4.1.1. RIAPS utilization in blackstarting use case

The designed operational use case of microgrid blackstart in a distributed way can be implemented using RIAPS. Fig. 3 shows the application architecture utilizing RIAPS platform framework and services. Other than platform services, needed actors made of components to realize microgrid automation are also illustrated and described as follows.

- Device interface: The duty of the device interface component is to communicate with the device using device-level communication protocols such as Modbus, C37, etc. This component is needed in every actor connected with real power system devices.
- Outer interface: The responsibility of the outer interface is to receive initializing requests from the blackstarting agent and send the acknowledgment to the blackstarting agent with the help of device interface component. In addition, it is responsible for publishing state measurements and receiving state measurements from the other actors if required. The outer interface component is used in grid following, grid forming and load connection actor. However, the outer interface for the blackstarting actor is responsible for sending blackstart initializing requests and receiving initializing acknowledgment from other actors and relay data and status message from the relay actor. In addition, it also sends a "started" message when the grid is formed and realized by the monitoring component of blackstarting agent.
- Monitoring: The responsibility of the monitoring component in blackstarting actor is to monitor the blackout situation and blackstarting capability of the microgrid based on information collected by its outer interface component.
- Relay data and status: The duty of this component is to collect relay status and measurements (frequency, voltage, and phase) of the microgrid through device interface components and send them to the blackstarting agent.
- Grid Forming: The responsibility of this component is to decide whether it is participated in forming the grid or not. The decision is taken from the collecting state measurements from other grid forming candidates and leader election services provided by the RIAPS. The presence of other grid forming candidates is known from the discovery service. The grid forming component can regulate voltage and frequency through the device interface.
- Grid following: The grid following component decides to connect to the grid based on information received by its outer interface. The influential information for this component is the microgrid's capacity need and technical capability. The capacity need and technical capability are calculated

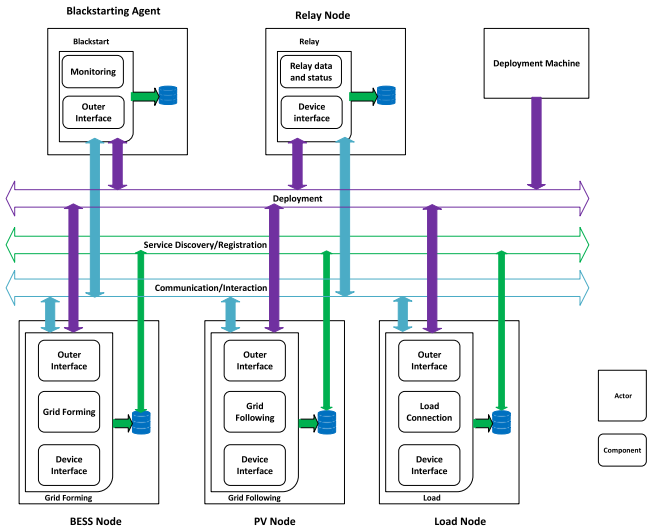


Fig. 3. Blackstarting application architecture utilizing RIAPS..

from the state measurements of other grid following actors and load actors. In addition, when needed, it can regulate its active and reactive power setpoint through the device interface.

- Load connection: load connection component is utilized here to make the connection decision from several loads existing in the microgrid. Connection decision is based on microgrid technical capability information and criticality information of load. Technical capability is calculated based on received information of connected other actors.

All the related components are grouped to form an actor and included them in an RIAPS application. The actors are separated from each other by providing the IP address of each node participating in an application. Later, application image can be deployed to the RIAPS node form the deployment machine. The raspberry pi can be used as a RIAPS node and located close to the real devices. Blackstarting agent is a software module containing blackstarting actor and its components running in raspberry pi but is not attached to power system devices.

4.2. Arrowhead framework

4.2.1. Arrowhead framework architecture

The Arrowhead is an open-source, service-oriented framework developed for industrial automation to automate the Industrial Internet of Things (IIoT). The framework consists of mandatory core services and auxiliary services depending on the application needs [25,55]. The core services include service registry, orchestrator and authorization:

- Service Registry: the responsibility of the service registry is registering the available services and keeping track of them. The service providers register their services when available and unregister when unavailable.
- Orchestration: The orchestrator enables the consumers to connect to suitable service instances. The service registry cannot do this because it has the only task to keep the track of available services. Orchestrator comes in this place to provide information about right services to the appropriate consumers.
- Authorization: Verifying the consumer to provide information about the service provider is needed to avoid unauthorized service usage. The authorization service from the core system generates a token-based permission for the consumers. The orchestrator asks about legitimate consumers from the authorization service and provides service provider information accordingly.

Besides the core service, Arrowhead provides supporting services to make an application system functional, such as gatekeeper service, event handler service, quality of service, etc. The gatekeeper service is usually utilized to enable inter-cloud information exchange [56]. Therefore, gatekeeper service can be used for inter-microgrid interaction or interaction with other system specified by use case.

4.2.2. Arrowhead utilization in blackstarting use case

The Arrowhead-based blackstarting system is illustrated in Fig. 4. In Microgrid, the resources such as grid-forming, grid-following, and loads are located locally, forming a closed boundary with sufficient communication and power network. These local resources aim to operate as a single entity without external support. Therefore, according to Arrowhead, a microgrid can be considered a local cloud and utilized as such. The blackstarting agent participating in this system is responsible for producing and receiving the events originated by the available actors or services. The events are classified in the following way:

- Blackstart initializing event: When the blackstarting agent realized blackout situation and relay opening status from the isolation actor, initializing event is produced by sending a “start” message.
- Grid forming event:The blackstarting agent realized the grid is formed provided voltage and frequency information by the isolation actor and sends a “started” message.
- Initializing acknowledgment event: Event produced by the grid forming, grid following, and load service in response to blackstart initializing event.
- A continuous measurement and connection status publishing event: is produced and consumed by grid forming, following, and load service. It is required to make sure all the service actors can get the available information on other actors.

As illustrated in Fig. 4, the blackstart use case can be implemented utilizing arrowhead core services and message-oriented middleware (message bus). In addition, the functionality of resources needs to be implemented as a service to make it compatible with the arrowhead core services. Blackstart initializing event is produced when it receives information about the blackout situation and the relay opening status from the isolation actor. Grid forming, grid following, and load service have their own dedicated interfaces to receive events from the blackstarting agent and deliver initializing acknowledgment messages to the blackstarting agent through the implemented message bus. All the available service needs to be registered in the service registry at the initial condition. A grid forming service is necessary in the blackstart use case to energize the microgrid. This service aims to form the grid by regulating the frequency and voltage of battery energy storage. It has another interface to receive the start message as an event from the blackstarting agent to prepare its service. After receiving the start message, the available grid forming services decide whether to start its service based on the other grid forming actor’s information or not. During the decision process, the presence of other grid forming services is obtained from the orchestrator. Orchestrator checks the available and authorized grid forming services from the service registry and authorization service. Then, the orchestrator provides information about them to all available grid forming services. The grid forming candidates then subscribe to other grid forming

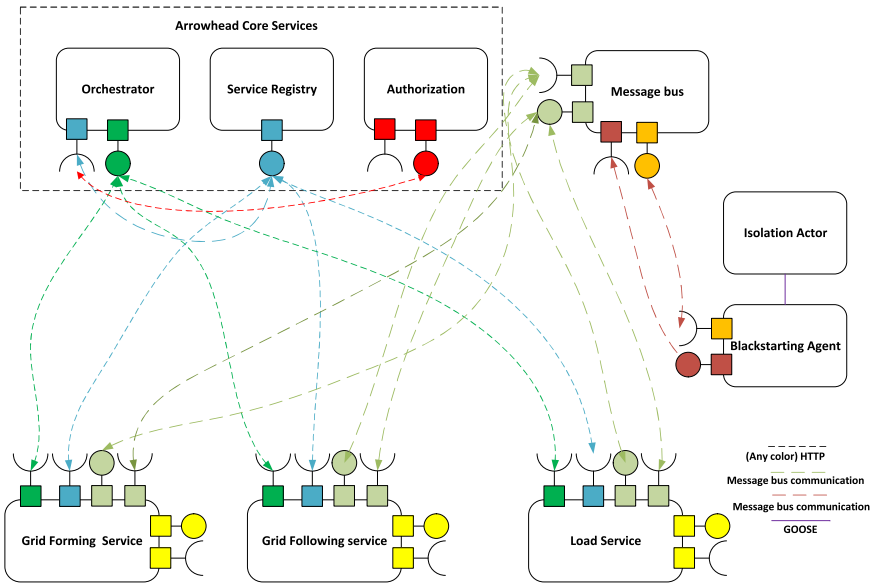


Fig. 4. Blackstarting application architecture utilizing Arrowhead.

candidate's information, and it receives from continuous measurement and connection status publishing events. Later, the grid forming candidates decide their participation in a grid forming by themselves. For example, suppose the SoC is the determinator for selecting a grid forming service. In that case, every candidate subscribes to the SoC information of others and compares them with their own SoC value. The highest SoC one will be selected to provide a grid forming service. The selected grid forming service starts forming the grid. When the grid is formed and blackstarting agent realizes it, a new event message "started" is produced. The grid following service and load service receives that new event and provide services by connecting to the grid, similar to the grid forming service.

5. Evaluation criteria formulation

This section formulates the criteria for evaluating RIAPS and Arrowhead. The criteria consider microgrid automation and energy internet characteristics. In addition, an explanation of each criterion is provided.

Interoperability of information exchange. In a microgrid automation system, the information flow happens vertically from enterprise-level components to field-level components typically applied for market integration of DERs or from centralized microgrid controller to field-level components applied for microgrid's internal control. Moreover, in the distributed approach, information also flows horizontally among the field devices (e.g. Intelligent Electronic Devices (IEDs) of DERs) and systems (e.g. Home or Battery Energy Management System, HEMS/BEMS) of the customers within the microgrid). Both flows of information need to encode and decode different data formats depending on the communication protocol used. The criteria for evaluating the framework for interoperability is to find out the data handling ability of the selected framework. Can the platform or framework translate the semantically and syntactically correct forms of information or services for the destination sources in a heterogeneous environment?

Interoperability of interfaces and communication pattern, and network. In a microgrid automation application design, the software components or services need to interact with each other. Therefore, the component or service needs to provide an interface to facilitate either synchronous or asynchronous communication. Moreover, components or services of one application under one software framework need to interact with other applications of different frameworks locating different networks during the system of system integration. Thus, the interoperability of interfaces and communication patterns evaluates the ability to support different communication patterns during component interaction, and their interfaces provide accessibility supporting inter-network communication.

Cybersecurity. In microgrid automation, information and communication technology monitors and controls the physical processes through computation-enabled embedded devices makes a cyber–physical system. Therefore, the microgrid automation system needs special attention to cybersecurity aspects. In this evaluation criteria, security measures provided by the selected software framework or platform are conceptually examined. Significantly, confidentiality, integrity, and availability attributes are focused on the evaluation.

Authentication and authorization are the essential aspects of designing a platform or framework for microgrid automation. In a microgrid automation application, the participation of nodes or actors can be dynamic. Therefore, the nodes or actors providing or consuming services need to be registered and authorized as legitimate service providers or consumers. Consequently, the platform or framework should have an authorization mechanism to protect the application from unauthorized service usage and participation.

Scalability. In a heterogeneous microgrid and energy internet environment, the number of physical actors in one system and the number of systems that need to interact with the microgrid is not limited. Thus, the software framework needs to integrate the increasing number of physical actors in one system and system of systems. Adding more physical actors into the system locally in the local network is called horizontal scaling. On the other hand, vertical scaling means adding higher-level systems to the local system. In addition, components, user requests, data volume, storage capacity may increase in microgrid automation applications. The scalability criteria define the power of scaling capacity of a software framework or platform depending on application needs.

Table 2
Summary of evaluation.

Properties		RIAPS	Arrowhead
Interoperability of information exchange	Device-level to the framework or platform-oriented data format	✔	✖
	Template to convert device-level data format to data format chosen by the developer/used in another framework	✖	✖
	Possibility of syntax to semantic conversion	✖	✔
Interoperability of interfaces, communication pattern, and network	Availability of message bus or library for message passing	✔	✖
	Message bus integration possibility	✖	✔
	Synchronous and asynchronous communication patterns	✔	✔
	Open API specification	✖	✔
	Internetwork communication	✖	✔
Cybersecurity	Authentication, authorization, and access control	✔	✔
	Log monitoring for accountability	✔	✔
	Resource monitoring	✔	✖
	Trust chain management	✖	✔
	Application-level security	✔	✖
Scalability	Horizontal scaling	✔	✔
	Vertical scaling	✖	✔
Evolvability	Loose coupling	✖	✔
	Configurability	✔	✔
	Ease of maintenance	✖	✔
	Enhancement plan	✔	✔
Real-time support	Hard real-time	✔	✖
	Soft real-time	✔	✔
	Time synchronization	✔	✖
Fault-tolerant	Active fault handling	✖	✖
	Passive fault handling	✔	✖

Evolvability. In a distributed automation system, the software involved needs to support the run-time changes of hardware and software components. There should be a mechanism to add new abilities to components and fix potential errors without degrading the performance of the software. These evaluation criteria are set here to check the availability of such kind of mechanism in the software framework.

Real-time support. Several time-sensitive tasks need to be performed on time in the microgrid automation system. The framework should have a mechanism to support real-time communication and action to complete the task with maximum timing accuracy. This criterion checks the technology or tool available in a platform or framework for applying real-time features in an application

Fault tolerance. Faults can happen anywhere in the system. They can be categorized as application, physical, or framework fault. A framework should have a mechanism to detect anomalies in the framework level fault and mitigation abilities to run the application smoothly. This criterion evaluates or checks what kind of mechanism is provided by the selected framework or platform.

6. Analysis and comparison of RIAPS and Arrowhead framework

The detailed analysis of the MAS-based RIAPS platform and the service-oriented Arrowhead framework is presented in this section. Table 2 summarizes the results of the analysis, whereas the following paragraphs explain these in more detail.

Interoperability of information exchange. The interoperability of information exchange in a platform or framework depends on the mechanism available to convert different data formats originating from various sources to an understandable format for the destination sources.

RIAPS uses a device abstraction mechanism to mimic the power system device to a RIAPS component interacting with other components. The message format during the interaction is based on Cap'n Proto serialization. This has been designed for fast transmission and a small message size, but the format less commonly used compared to alternatives, such as XML, and JSON. This is likely to make the platform challenging when interacting with another platform. In addition, the platform does not provide any mapping tools, such templates, to transform messages into another format. Secondly, the platform is unable to translate the semantic meaning of the data exchanging between the components, which is primarily required in the WoT concept. Thus, there are restrictions regarding the umbrella of the energy internet concept.

On the other hand, Arrowhead, a framework that utilizes service-oriented architecture, does not provide any mechanism to communicate with low-level devices like RIAPS. Instead, Arrowhead does not restrict developers to developing an adapter to translate device-level protocol message format to any format chosen by the developers. Secondly, Arrowhead introduced a concept of a translator service [57]. The translator services consist of converting abilities of different communication protocols and message formats. It is called when a service provider or consumers need to communicate using different communication protocols and message formats. Finally, the Arrowhead is open to extend syntactic data model to semantic information model implementing a translator. However, it does not provide enough documentation on how to implement translation services.

Interoperability of interfaces, communication pattern, and network. The MAS-based RIAPS has limitations in openness, providing interoperability in communication interface, pattern, and network. For example, the RIAPS utilizes ZeroMQ messaging system for component interactions, ZeroMQ is used for message bus implementation. The brokerless architecture of ZeroMQ makes the application distributed with low latency and high throughput. In addition, ZeroMQ supports both the publish–subscribe and request–response communication pattern, which makes the platform interoperable with different patterns. However, ZeroMQ

lacks any tools for reliable message delivery to the destination sources. In microgrid automation, reliable message delivery is critical. Some other protocols, for example, DDS (Data Distribution Service), MQTT, or AMQP, provide tools for reliable messaging. Therefore, RIAPS necessitates another message-bus-like middleware in the use cases where reliable data delivery is necessary. In ZeroMQ, message publishing is done by broadcasting or multicasting to the local area network which prevents scalability to wider networks. In reality, the resources can be situated in different networks.

The SOA-based Arrowhead provides more freedom for the developer to implement a use case. Thus, the interoperability of protocol, middleware-based message bus or other framework integration is easy compared to RIAPS. For instance, Arrowhead is a framework for local cloud automation that does not have message bus implementation. Instead, it does not restrict implementing a message bus required for the use case and completely depends on the developer's choice. In addition, the core services provided by the framework have their endpoints description to utilize them in the application. Arrowhead framework enables publish–subscribe and request–response communication patterns. The current release, Eclipse Arrowhead version 4.3 [58], has shown the integration mechanism with another framework. In that release, FIWARE IoT platform is integrated with Arrowhead. Moreover, Arrowhead provides a gateway mechanism called gatekeeper system to connect with another local automation cloud. This mechanism solves the network interoperability problems.

Cybersecurity. Concerning cybersecurity solutions in Arrowhead, it relies on Secure Sockets Layer (SSL) chain trust to secure and encrypt connections between the application system and the core services. Secondly, it utilizes token-based authentication and authorization for the service requester and service provider. This mechanism is built-in in the authorization service in the Arrowhead framework. Finally, inter-cloud communication utilizes SSL security and a chain of trust to authorize and encrypt communication between two application systems running on different clouds. However, the application systems under the local Arrowhead cloud have the autonomy to interact with its service components. Arrowhead does not bother how the service consumer and provider exchange their data. Therefore, vulnerabilities might appear if an application developer lacks deep insight into security. One security issue in Arrowhead is if a service provider denies providing service after the verification, this might seriously hinder running the application. Therefore, the monitoring service for the orchestrated services is needed to avoid this problem. Furthermore, the availability aspects of the core services are critical since all the application system is dependent on the core services for running the application. Therefore, a mechanism must be present to monitor the resource utilized by the core services or monitor the core services correctly.

In RIAPS, an encrypted and cryptographically signed application package, including components and services required for running the application, is deployed to the RIAPS nodes. The RIAPS nodes decrypt the package and install the application. Encrypting the package before deployment protects the application from any alteration in application components and services if deployed from the deployment machine. The data integrity during the interaction between the components in a same application is done by elliptic curve cryptography. Similarly, the service registry is encrypted. The authorization of actors in communication is solved by public and private key pairs generated during the application deployment and stored. In addition, AppArmor, mandatory access control, restricts the communication between unauthorized nodes in an application process. Furthermore, logging service, resource monitoring services utilized in RIAPS to record the component interaction and monitor resource availability. However, RIAPS lacks the implementation of security mechanisms between different applications running in the nodes. In addition, RIAPS application deployment and interaction of actors happens in the same network. Thus, implementing RIAPS application in nodes residing in different networks and security solution is unaddressed.

Scalability. Scalability issues are handled in a distributed fashion in RIAPS. RIAPS utilizes OpenDHT (DHT = Distributed Hash Table) for registering services, and this is distributed to all the nodes participating in an application. OpenDHT is highly scalable and capable of connecting millions of nodes in an application in a local network. The storage of the service registry is no longer centralized, and the amount of storage is increased when the node is increased. The application component or actors in an application can be increased depending on the need. However, it is unclear how service discovery happens when the nodes are not a part of local systems.

On the other hand, Arrowhead can interchange services between different local clouds using a gatekeeper system. It means that Arrowhead supports the integration of different subsystems. The systems can be either local or higher-level systems. In addition, the scalability of the local cloud is fully dependent on the centralized core system. The core system can be implemented utilizing distributed virtualization in order to support large volumes of services within the local system.

Evolvability. In RIAPS, application components are tightly coupled to the actors running in a system. Therefore, the ability to change components is not possible in the node. Instead, it needs to be done from the deployment machine by making an application package including changed components and deployed to the nodes. Moreover, the deployment mechanism of the whole application to node hinders run-time addition or modification of components in nodes. However, adding new nodes and leaving the existing node is possible without affecting the application and performance. For example, adding new load node to the application using the RIAPS platform is possible in the blackstarting use case. On the other hand, Arrowhead provides a full autonomy for the application system in terms of programming languages and the resources where the application system is running. The application system consists of separate independent services, and the services have a high degree of autonomy to influence their execution environment. Thus evolving the services over time does not affect other services or framework core services. In addition, Arrowhead consortium has a long-term goal to add enhancement services considering the ease of application system design and reliability.

Real-time support. Both Arrowhead and RIAPS provide real-time features for the application but in different ways. For example, RIAPS depends on the real-time features of embedded Linux technology to enable prioritizing event-driven tasks. Secondly, device interface service, faster messaging pattern and serialization technology are used to minimize latency. Finally, a time synchronization service is used to enable scheduled control action in a timely manner. Arrowhead utilizes a Quality of Service (QoS) manager system to satisfy soft and hard real-time requirements. The QoS manager is responsible for orchestrating services that need real-time support. For real-time support, the service consumers need to request QoS requirement as a service level agreement to the orchestrator. Then, there is an interaction between the orchestrator and the QoS set up service of QoS manager to verify and configure the network as per QoS requirement parameter, such as end-to-end delay, priority requirement, etc. Another service from the QoS manager, QoS monitoring service, is utilized to monitor QoS parameters during the service provider

and consumer interaction. However, very strict hard real-time requirements between the service provider and consumer can only be met in the situation where the service discovery has been performed beforehand, and service should be implemented in a way to meet these requirements. Therefore, in dynamic environments hard real-time requirements can be challenging to reach with Arrowhead.

Fault tolerance. RIAPS provides fault management techniques to detect faults in the RIAPS framework and possible ways to solve them. For example, detection mechanisms and possible recovery of actor termination, network connection failure, and application deployment failure are present in the RIAPS platform. However, active fault handling without restarting the services is absent. Therefore, the application might stop for a few seconds due to the absence of an active fault management process providing redundant services. In addition, any application-level faults rely fully on developer skills to handle them in an efficient manner. Possible fault in the application and solution is also listed in [34, 54] to demonstrate fault tolerant capabilities of RIAPS platform. In Arrowhead, there is no fault management service to detect faults in the framework or application system, but the systems themselves can react by, e.g., requesting new services from the orchestrator that it requires.

7. Combining MAS and SOA approaches

The convergence of MAS and SOA can be beneficial for designing a microgrid automation system enabling the energy internet. This section contains a proposal to combine RIAPS and Arrowhead with an adoption example and its benefits for enabling energy internet.

7.1. Proposed architecture

The convergence of MAS and SOA is described in several previous research papers in [59] and in [26,60,61]. Those research mostly stated some techniques to combine MAS and SOA considering the best features from both paradigms. Moreover, most combinations are proposed to realize industrial automation. However, in [62] the authors proposed an approach combining agent and service-oriented frameworks to realize Industry 4.0. Currently, frameworks to develop an application using MAS or SOA are available in reasonable numbers. Therefore, selecting one framework from MAS and one from SOA, then combining them for microgrid automation considering energy internet, is proposed.

RIAPS meets the good level of requirement to design automation systems at the microgrid level. For instance, device-level interoperability to interact with the lower-level devices, mechanisms for enabling distributed control, and real-time support considering smart-grid applications. Moreover, the horizontal scalability supports the extension of nodes locally, which cannot hinder the growing number of nodes participating in one application locally. Finally, its cybersecurity measures and fault handling features ensure that the application is secure and resilient.

On the other hand, microgrid automation needs to be interoperable through the interfaces and network integration mechanisms to enable the energy internet to communicate with horizontally or vertically with other systems. The absence of these features makes RIAPS unscalable vertically or horizontally in different networks. Fortunately, providing a high degree of autonomy to design an application system interface, gateway mechanism, and cybersecurity mechanism provided by Arrowhead can extend the local network to the internet level in an interoperable, scalable, and secure way. Therefore, designing the microgrid application in RIAPS utilizing the local resources and providing an interface to interact with Arrowhead can achieve the goal of the energy internet.

Fig. 5 illustrates the combination of RIAPS and Arrowhead. In Microgrid cloud, applications are designed utilizing RIAPS and provide an interface compatible with the Arrowhead core system. The application made with RIAPS for the blackstarting use case is described in Section 4 and illustrated in Fig. 3. When the application system is compatible with Arrowhead, then the service provided by it can be registered and orchestrated among other service consumers within the local cloud. Moreover, a service needed by service consumers located in another cloud can be invokable. The gateway and gatekeeper systems of the local cloud can facilitate this service invocation from other clouds. The gatekeeper is used for inter-cloud service orchestration, where the gateway is used as a trusted tunneling agent supporting information exchange. For example, if DSO needs to utilize service from the microgrid cloud, the consumer application of DSO cloud looks for that service from the local orchestrator. The local orchestrator will initialize global service discovery through the gatekeeper system. Then, the orchestrator from the DSO cloud will choose the cloud providing that service wanted to invoke and request to the orchestrator of the provider cloud (microgrid). After that, the provider cloud's orchestrator checks the authorization status of the consumer from the authorization core service in response to the request. Finally, when the consumer service is authorized, information exchange is realized between two clouds facilitated by the gateway of each cloud.

7.2. Adoption example and benefits

The microgrid can offer various services for external stakeholders. The services can include, for instance, flexibility services, intentional islanding services, power balancing, and load balancing services for congestion management. In the cases mentioned, the microgrid needs to respond to external requests from actors such as DSO, TSO, aggregator, and market. Besides this, the microgrid may need to coordinate with other microgrids to fulfill external stakeholders' requirements. The proposed converging approach of RIAPS and Arrowhead can be one option to solve this issue. For instance, considering intentional islanding service and power supply restoration service from the microgrid utilized by DSO. In this case, Arrowhead compatible DSO finds the local microgrid cloud providing islanding service and power supply restoration service through the gatekeeper system. Later, the DSO cloud sends consumption of services requests to the microgrid cloud. Local microgrid checks the authorized consumer from the authorization service to utilize islanding service and power supply restoration service. Finally, DSO sends a command to the islanding service to initiate islanding, and then islanding initiates a blackstarting application implemented using RIAPS.

The combined approach of RIAPS and Arrowhead satisfies local and global requirements, enabling both control and internet-scale communication. The distributed control, real-time requirements, and some degree of resiliency at the microgrid level are solved by RIAPS as a local software framework. Arrowhead fulfills the internetwork interoperability and security requirements of inter-cloud service exchange between the microgrid and DSO cloud. This combination may outweigh the current practices of using either MAS or SOA for both control and communication.

8. Discussion

Neither MAS nor SOA can alone implement microgrid automation for control functionalities and the energy internet vision.

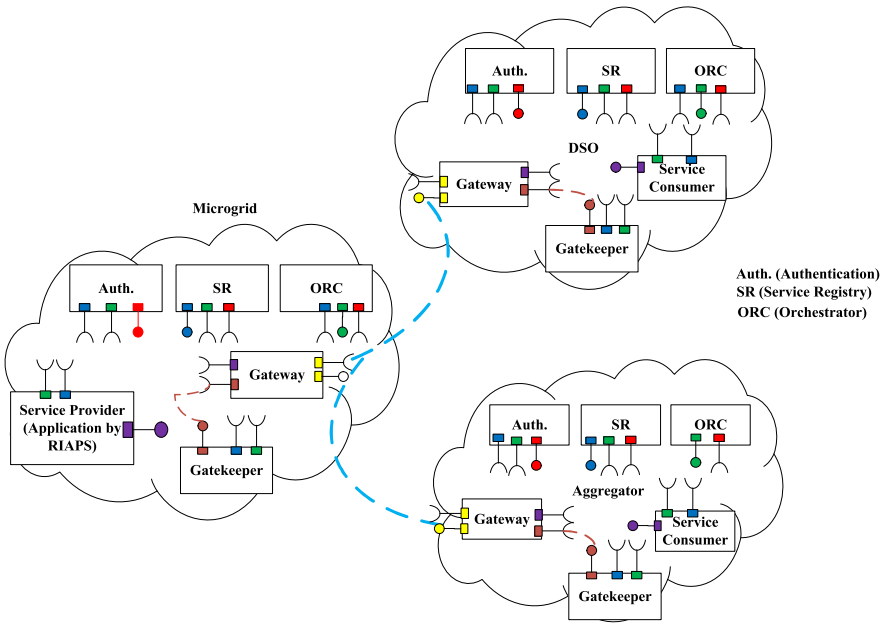


Fig. 5. Proposed convergence of RIAPS and Arrowhead.

This study proposes that the combination of RIAPS and Arrowhead, in other words, the integration of MAS-based platform and SOA-based framework, could be the solution for implementing microgrid automation under the energy internet umbrella.

The suggested architecture would provide benefits in a number of use cases. For example, first it is beneficial to lower the extent of the power peaks within the grid. In this effort, the SOA-based communication enables DSO to request for the microgrid and its participants to re-schedule their power consumption. Second, the architecture would as well enable the DSO to control DERs in a microgrid securely provided that the SOA framework provides security features (such as Arrowhead does). As a third example, any DERs in the microgrid could participate in the FCR-N (Frequency Containment Reserve for Normal Operation) market. In all of the examples, the local systems in the microgrid would directly communicate only via MAS technology, but an appropriate MAS-SOA gateway would connect these to the internet.

The concept of energy internet is to integrate several heterogeneous systems and operate them in an interoperable manner. Therefore, to guarantee the interoperability between systems of systems, the suggested MAS and SOA based architecture needs to adopt power system data standards, such as IEC 61850 and CIM. These can be easily utilized in the proposed architecture. This means that the system-wise adoption of the proposed architecture, containing Arrowhead technology, gives freedom to the developer to choose the data model in a unified way to integrate the systems. The lower-level devices that support IEC 61850, Modbus and others can be easily communicated by the RIAPS device interface services. Thus, the legacy systems, devices, and protocols can live together with the newly evolved services and systems. In addition, it might save investment cost by not replacing legacy systems, devices and protocols regardless of adopting new technologies.

Adapting the integration of MAS and SOA is difficult in the smart grid due to the system complexity and multi-stakeholder involvement. In the smart grid, domain professionals are required to design use cases and specify requirements based on business models and technical constraints. Therefore, to realize the adoption of the proposed architecture in the power system, there is a need to outsource the task to third parties. As a result, several third parties can be involved in the system development. Firstly, there can be a consultation company to derive business models based on the available technologies, both ICT and energy. Secondly, a software development company can design the system architecture, develop the system and performs upgrades. Thirdly, a communications company can establish the communication between different systems. Finally, a maintenance and operation company can maintain the automation system for it to perform well.

It is not only a technical question to change the current hierarchical network automation into the proposed architecture with service-oriented internet communication, but the business model must change as well. The proposed evolution necessitates a collaborative ecosystem that agrees about the common practices, technologies, and business models. The ecosystem would together decide, for instance, which information models and standards to use in the communication and how to develop these to match future needs. This enables ecosystem-wide governance and interoperability as suggested in European Interoperability Framework (EIF [63]). On the other hand, as new technology appears, it is a driver to change the business model [64, p. 61]. Still, concrete effort is necessary from the engineering community and possibly from public administrative bodies as the change cannot occur by itself.

As a limitation, this article lacks a detailed study about information and data models. These are crucially important in information exchange and could therefore be studied in the future. In particular, microgrid-related systems should aim at interoperability and thus apply existing standards, such as CIM (IEC 61968 and 61970) or IEC 61850.

This study evaluated RIAPS and Arrowhead in microgrid automation, to compare characteristics of MAS and SOA based concepts considering energy internet applications. It was conducted to find a suitable framework that enables energy internet for automating microgrid applications. Additionally, it is worthwhile to mention that the evaluation was performed from the viewpoint of software architecture and the framework services provided for application development. Thus, the study also lacks a performance analysis from the implementation context that is scheduled for future research. RIAPS has its strength on the local level, whereas Arrowhead shows its benefits on internet

level communication and integration. This research might lead to the practical implementation of the designed use case and its extension to interact with other systems utilizing RIAPS and Arrowhead separately, examining both frameworks based on the criteria set in this research. Later, the proposed combined approach could be used to solve the issues and conduct a feasibility study on microgrid automation and energy internet through implementation.

9. Conclusion

The purpose of this study was to determine the strengths and weaknesses of MAS-based and SOA-based software frameworks in microgrid automation from an energy internet perspective. The representatives of MAS and SOA-based framework, RIAPS and Arrowhead, were selected after a literature study. Furthermore, the microgrid automation scenario consists of functional, information, communication, and cybersecurity requirement developed for utilizing RIAPS and Arrowhead. However, the study is not limited to thinking only about the developed scenario. It also considers other use cases such as microgrid interaction with DSO and inter microgrid interaction. Evaluation criteria were set and applied to find RIAPS and Arrowhead's strengths and weaknesses.

Arrowhead and RIAPS, both, have strengths and weaknesses. Arrowhead provides good features in terms of interoperability of information exchange, communication, and network. It also has good cybersecurity and evolvability features for the dynamic environment. Those features are prerequisites for the energy internet vision. In addition, microgrid control and automation application implementation are also possible under Arrowhead framework. However, to implement an application, relatively lot of effort is necessary to maintain cybersecurity and compatibility with Arrowhead core system. In addition, a service-oriented system might fail to meet real-time requirements for the application where real-time requirements are crucial. On the other hand, the control and automation of microgrid applications in a distributed way are easy to implement utilizing RIAPS. Advanced control functionalities, distributed service discovery, hard-real time features, and fault management plan of the platform services make it easy to use for the developer. However, it lacks some interoperability and evolvability features that are better in the Arrowhead framework.

This conceptual investigation shows that only MAS-based or SOA-based software cannot enable energy internet automating advanced control functionalities in the microgrid. Service-oriented architecture is more suitable for the energy internet. On the other hand, MAS-based software has perfect features to implement distributed control functionalities in real-time applications locally. This insight gained from the study may be the assistance of choosing the proper software framework depending on the application scope. Therefore, this study proposes that MAS and SOA should co-exist to enable microgrid automation in the scale of energy internet.

Further experimental research could be conducted to verify the strengths and weaknesses of RIAPS and Arrowhead found in this research. Moreover, there should be practical case studies about the proposed combination of RIAPS and Arrowhead or MAS and SOA in general, as this seems like a promising solution for microgrid automation realizing energy internet.

CRediT authorship contribution statement

Md Tanjimuddin: Conceptualization, Writing – original draft, Writing – review and editing, Investigation, Software. **Petri Kannisto:** Writing – original draft, Writing – review and editing. **Peyman Jafary:** Writing – original draft, Writing – review and editing. **Mikael Filppula:** Writing – review and editing. **Sami Repo:** Conceptualization, Writing – review and editing, Supervision, Funding acquisition. **David Hästbacka:** Conceptualization, Writing – review and editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the Academy of Finland under the project Distributed Management of Electricity System (DisMa) [grant/application IDs 322673 and 322676]. The authors acknowledge support from the colleagues from Tampere University (TAU) and the project partner VTT Technical Research Centre of Finland, especially Kalle Ruuth (TAU), Antti Supponen (TAU) and Amir Safdarian (VTT).

References

[1] Government's climate policy: climate-neutral Finland by 2035, Ministry of the Environment Finland, 2021, https://ym.fi/en/climate-neutral-finland-2035 [Retrieved 27 Dec 2021].
[2] A clean planet for all: A European strategic long-term vision for a prosperous, modern, competitive and climate neutral economy, 2018, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52018DC0773&from=EN [Retrieved 27 Dec 2021].
[3] X. Song, Research on security protection architecture of energy internet information communication, in: MATEC Web of Conferences, Vol. 228, EDP Sciences, 2018, p. 02010.
[4] S. Hussain, F. Nadeem, M.A. Aftab, I. Ali, T.S. Ustun, The emerging energy internet: Architecture, benefits, challenges, and future prospects, Electronics 8 (9) (2019) 1037.
[5] H.M. Hussain, A. Narayanan, P.H. Nardelli, Y. Yang, What is energy internet? Concepts, technologies, and future directions, IEEE Access 8 (2020) 183127–183145.
[6] K. Mahmud, B. Khan, J. Ravishankar, A. Ahmadi, P. Siano, An internet of energy framework with distributed energy resources, prosumers and small-scale virtual power plants: An overview, Renew. Sustain. Energy Rev. 127 (2020) 109840.
[7] B. Yan, B. Wang, L. Zhu, H. Liu, Y. Liu, X. Ji, D. Liu, A novel, stable, and economic power sharing scheme for an autonomous microgrid in the energy internet, Energies 8 (11) (2015) 12741–12764.
[8] J. Ahmad, M. Tahir, S.K. Mazumder, Improved dynamic performance and hierarchical energy management of microgrids with energy routing, IEEE Trans. Ind. Inf. 15 (6) (2018) 3218–3229.
[9] L. Ribeiro, J. Barata, P. Mendes, MAS and SOA: Complementary automation paradigms, in: A. Azevedo (Ed.), Innovation in Manufacturing Networks, Springer US, Boston, MA, 2008, pp. 259–268.
[10] Resilient Information Architecture Platform for the Smart Grid, https://riaps.isis.vanderbilt.edu/ [Retrieved 13 Dec 2021].
[11] Arrowhead, https://www.arrowhead.eu/ [Retrieved 13 Dec 2021].
[12] K. Zhou, S. Yang, Z. Shao, Energy internet: the business perspective, Appl. Energy 178 (2016) 212–222.
[13] X. Zhang, K. Li, D. Li, M. Zhong, W. Huang, Digital twin in energy internet and its potential applications, in: 2020 IEEE 4th Conference on Energy Internet and Energy System Integration, EI2, IEEE, 2020, pp. 2948–2953.
[14] A. Joseph, P. Balachandra, Energy internet, the future electricity system: Overview, concept, model structure, and mechanism, Energies 13 (16) (2020) 4242.
[15] Z. Chen, Q. Liu, Y. Li, S. Liu, Discussion on energy internet and its key technology, J. Power Energy Eng. 5 (12) (2017) 1–9.
[16] H. Pourbabak, T. Chen, W. Su, Centralized, decentralized, and distributed control for energy internet, in: The Energy Internet, Elsevier, 2019, pp. 3–19, http://dx.doi.org/10.1016/C2016-0-04520-5.
[17] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, J. Wu, A survey on energy internet: Architecture, approach, and emerging technologies, IEEE Syst. J. 12 (3) (2017) 2403–2416.
[18] A.Q. Huang, M.L. Crow, G.T. Heydt, J.P. Zheng, S.J. Dale, The future renewable electric energy delivery and management (FREEDM) system: the energy internet, Proc. IEEE 99 (1) (2010) 133–148.
[19] T. Chen, H. Pourbabak, W. Su, Electricity market reform, in: The Energy Internet, Elsevier, 2019, pp. 97–121.

[20] A. Kantamneni, L.E. Brown, G. Parker, W.W. Weaver, Survey of multi-agent systems for microgrid control, Eng. Appl. Artif. Intell. 45 (2015) 192–203.
[21] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, Upper Saddle River, NJ, 2005.
[22] J. Xie, C.-C. Liu, Multi-agent systems and their applications, J. Int. Council Electr. Eng. 7 (1) (2017) 188–197.
[23] P.-M. Ricordel, Y. Demazeau, From analysis to deployment: A multi-agent platform survey, in: International Workshop on Engineering Societies in the Agents World, Springer, 2000, pp. 93–105.
[24] K. Kravari, N. Bassiliades, A survey of agent platforms, J. Artif. Soc. Soc. Simul. 18 (1) (2015) 11.
[25] D. Hästbacka, J. Halme, M. Larrañaga, R. More, H. Mesiä, M. Björkbom, L. Barna, H. Pettinen, M. Elo, A. Jaatinen, et al., Dynamic and flexible data acquisition and data analytics system software architecture, in: 2019 IEEE SENSORS, IEEE, 2019, pp. 1–4.
[26] P. Leitão, P. Vrba, T. Strasser, Multi-agent systems as automation platform for intelligent energy systems, in: IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2013, pp. 66–71.
[27] R. Belkacemi, A. Feliachi, M. Choudhry, J.E. Saymansky, Multi-agent systems hardware development and deployment for smart grid control applications, in: 2011 IEEE Power and Energy Society General Meeting, IEEE, 2011, pp. 1–8.
[28] S. Alwala, A. Feliachi, M.A. Choudhry, Multi agent system based fault location and isolation in a smart microgrid system, in: 2012 IEEE PES Innovative Smart Grid Technologies, ISGT, IEEE, 2012, pp. 1–4.
[29] A. Alseyat, J.-D. Park, Multi-agent system using JADE for distributed DC microgrid system control, in: 2019 North American Power Symposium, NAPS, IEEE, 2019, pp. 1–5.
[30] J.-C. Gu, M.-T. Yang, J.-D. Chen, H.-Y. Chung, C.-Y. Wang, Y.-R. Chang, Y.-D. Lee, C.-M. Chan, C.-H. Hsu, Application of multi-agent systems to microgrid fault protection coordination, in: 2016 International Symposium on Computer, Consumer and Control, IS3C, IEEE, 2016, pp. 188–191.
[31] R. Mehta, B. Menon, D. Srinivasan, S.K. Panda, A.K. Rathore, Market based multi-agent control of microgrid, in: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP, IEEE, 2014, pp. 1–6.
[32] M.H. Shawon, A. Ghosh, S. Muyeen, M.S. Baptista, S. Islam, Multi-agent based autonomous control of microgrid, in: 2020 2nd International Conference on Smart Power & Internet Energy Systems, SPIES, IEEE, 2020, pp. 333–338.
[33] K.A. Cort, J.N. Haack, S. Katipamula, A.K. Nicholls, VOLTTRON™: Tech-to-Market Best-Practices Guide for Small-and Medium-Sized Commercial Buildings, Tech. rep, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2016.
[34] P. Ghosh, S. Eisele, A. Dubey, M. Metelko, I. Madari, P. Volgyesi, G. Karsai, Designing a decentralized fault-tolerant software framework for smart grids and its applications, J. Syst. Archit. 109 (2020) 101759.
[35] S. Eisele, I. Mardari, A. Dubey, G. Karsai, Riaps: Resilient information architecture platform for decentralized smart systems, in: 2017 IEEE 20th International Symposium on Real-Time Distributed Computing, ISORC, IEEE, 2017, pp. 125–132.
[36] S. Lu, S. Repo, M. Salmenperä, J. Seppälä, H. Koivisto, Using IEC CIM standards and SOA technology for coordinated voltage control application, in: 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe, IEEE, 2019, pp. 1–5.
[37] M. Uslar, T. Schmedes, A. Lucks, T. Luhmann, L. Winkels, H.-J. Appelrath, Interaction of EMS related systems by using the CIM standard, ITEE (2005) 596–610.
[38] I. Lendak, E. Varga, A. Erdeljan, M. Gavrić, Restful web services and the common information model (CIM), in: 2010 IEEE International Energy Conference, IEEE, 2010, pp. 716–721.
[39] Q. Chen, H. Ghenniwa, W. Shen, Web-services infrastructure for information integration in power systems, in: 2006 IEEE Power Engineering Society General Meeting, IEEE, 2006, pp. 8–pp.
[40] Y. Shin, W. Park, I. Lee, Design of microgrid web services for microgrid applications, in: 2017 Ninth International Conference on Ubiquitous and Future Networks, ICUFN, IEEE, 2017, pp. 818–822.
[41] S. Lehnhoff, S. Rohjans, M. Uslar, W. Mahnke, OPC unified architecture: A service-oriented architecture for smart grids, in: 2012 First International Workshop on Software Engineering Challenges for the Smart Grid, SE-SmartGrids, IEEE, 2012, pp. 1–7.
[42] A. Burger, H. Koziolek, J. Rückert, M. Platenius-Mohr, G. Stomberg, Bottleneck identification and performance modeling of OPC UA communication models, in: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, in: ICPE '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 231–242, http://dx.doi.org/10.1145/3297663.3309670.
[43] OPC unified architecture specification part 14: PubSub, Release 1.04, OPC Foundation, 2018.
[44] P. Kannisto, D. Hästbacka, T. Gutiérrez, O. Suominen, M. Vilkko, P. Craamer, Plant-wide interoperability and decoupled, data-driven process control with message bus communication, J. Ind. Inf. Integr. (ISSN: 2452-414X) 26 (2022) 100253, http://dx.doi.org/10.1016/j.jii.2021.100253.
[45] FIWARE, https://www.fiware.org/ [Retrieved 27 Dec 2021].
[46] Eclipse BaSyx, https://www.eclipse.org/basyx/ [Retrieved 27 Dec 2021].
[47] AUTOSAR, https://www.autosar.org/ [Retrieved 27 Dec 2021].
[48] IoTivity, http://iotivity.org/ [Retrieved 27 Dec 2021].
[49] S. Kumar, S. Islam, A. Jolfaei, Microgrid communications–protocols and standards, in: Variability, Scalability and Stability of Microgrids, Vol. 139, 2019, p. 291.
[50] I. Serban, S. Céspedes, C. Marinescu, C.A. Azurdia-Meza, J.S. Gómez, D.S. Hueichapan, Communication requirements in microgrids: A practical survey, IEEE Access 8 (2020) 47694–47712.
[51] S. Repo, F. Ponci, D. Della Giustina, A. Alvarez, C.C. Garcia, Z. Al-Jassim, H. Amaris, A. Kulmala, The IDE4L project: Defining, designing, and demonstrating the ideal grid for all, IEEE Power Energy Mag. 15 (3) (2017) 41–51.
[52] P.P. Pereira, J. Eliasson, J. Delsing, An authentication and access control framework for coap-based internet of things, in: IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2014, pp. 5293–5299.
[53] A. Ashok, M. Govindarasu, J. Wang, Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid, Proc. IEEE 105 (7) (2017) 1389–1407.
[54] P. Ghosh, S. Eisele, A. Dubey, M. Metelko, I. Madari, P. Volgyesi, G. Karsai, On the design of fault-tolerance in a decentralized software platform for power systems, in: 2019 IEEE 22nd International Symposium on Real-Time Distributed Computing, ISORC, IEEE, 2019, pp. 52–60.
[55] H. Hoikka, A. Jaatinen, D. Hästbacka, Evaluating arrowhead framework for dynamic condition monitoring applications and edge computing in mining, in: 2020 IEEE Conference on Industrial Cyberphysical Systems, Vol. 1, ICPS, IEEE, 2020, pp. 577–582.
[56] C. Hegedűs, D. Kozma, G. Soós, P. Varga, Enhancements of the arrowhead framework to refine inter-cloud service interactions, in: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2016, pp. 5259–5264.
[57] P. Varga, F. Blomstedt, L.L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, I.M. de Soria, Making system of systems interoperable–the core components of the arrowhead framework, J. Netw. Comput. Appl. 81 (2017) 85–95.
[58] Eclipse Arrowhead, https://projects.eclipse.org/projects/iot.arrowhead/releases/4.3.0/plan [Retrieved 8 Jul 2022].
[59] J.M. Mendes, P. Leitão, F. Restivo, A.W. Colombo, Service-oriented agents for collaborative industrial automation and production systems, in: International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, 2009, pp. 13–24.
[60] R.L. Hartung, Service oriented architecture and agents: Parallels and opportunities, in: Agent and Multi-Agent Technology for Internet and Enterprise Systems, Springer, 2010, pp. 25–48.
[61] D.I. Tapia, J. Bajo, J.M. Corchado, Distributing functionalities in a SOA-based multi-agent architecture, in: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems, PAAMS 2009, Springer, 2009, pp. 20–29.
[62] H. Baumgärtel, R. Verbeet, Service and agent based system architectures for industrie 4.0 systems, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, pp. 1–6.
[63] New European Interoperability Framework, Tech. rep., European Commission, 2017, http://dx.doi.org/10.2799/78681.
[64] Innovation landscape for a renewable-powered future: Solutions to integrate variable renewables, Tech. rep., IRENA, 2019, https://www.irena.org/publications/2019/Feb/Innovation-landscape-for-a-renewable-powered-future [Retrieved 27 Jun 2022].